

**UNIVERZITA PARDUBICE**  
**FAKULTA ELEKTROTECHNIKY A**  
**INFORMATIKY**

**DIPLOMOVÁ PRÁCE**

**AUTOR PRÁCE: Bc. Mejsnar Lubor**  
**VEDOUČÍ PRÁCE: Ing. Fidler Tomáš**  
**2010**

**UNIVERZITA PARDUBICE**  
**FAKULTA ELEKTROTECHNIKY A**  
**INFORMATIKY**

**Optimalizace rozvoje firmy pomocí lokačních  
algoritmů.**

**AUTOR PRÁCE: Bc. Mejsnar Lubor**  
**VEDOUCÍ PRÁCE: Ing. Fidler Tomáš**  
**2010**





Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně Univerzity Pardubice.

V Pardubicích dne 21.5 2010

Bc. Mejsnar Lubor

## **ABSTRAKT**

Hlavním tématem diplomové práce je popis a implementace lokačních algoritmů pro optimalizace dep na dopravní síti.

V praktické části práce je vytvořena aplikace pro optimalizaci dep na dopravní síti, která bude následně využívána v praxi, dále jsou popsány důležité části aplikace, a způsob jejich řešení. Závěr je věnován praktickému příkladu. Jsou popsány podmínky optimalizace, a představeno řešení pro danou firmu.

## **KLÍČOVÁ SLOVA**

aplikace, lokace, lokační algoritmy, weisfeld, brute force, iterativní algoritmus, dopravní síť

## **TITLE**

Optimizing the development of business with locator algorithms.

## **ABSTRACT**

The main theme of this thesis is the description and implementation of algorithms for locational optimization dep on the transport network. location algorithm.

In the practical section is designed to optimize application dep on the transport network, which will subsequently be used in practice, are described as important parts of the application, and how to tackle them. The conclusion is devoted to practical examples. Condition optimization are described, and presented solutions for the firm.

## **KEYWORDS**

application, location, location algorithms, weisfeld, brute force, iterative algorithm, the transport network.

## Poděkování

Na tomto místě bych rád poděkoval Ing. Tomáši Fidlerovi za odborné vedení diplomové práce, za nápadité připomínky ale i kritické hodnocení práce. Také bych chtěl poděkovat majiteli firmy Mepo za předání odborných zkušeností z daného oboru.

*seznam obrázků*

|   |           |
|---|-----------|
| <b>Obrázek 1. Ukázka reprezentace grafu .....</b>                           | <b>24</b> |
| <b>Obrázek 2. Princip QuadTree .....</b>                                    | <b>37</b> |
| <b>Obrázek 3. Max. přiblížení využívající systém QuadTree.....</b>          | <b>38</b> |
| <b>Obrázek 4. UML diagram tříd uchovávající data. ....</b>                  | <b>43</b> |
| <b>Obrázek 5. popisující ovládací prvky systému. ....</b>                   | <b>44</b> |
| <b>Obrázek 6. Třída algoritmu Brute force .....</b>                         | <b>45</b> |
| <b>Obrázek 7. Třída algoritmu Weisfeld.....</b>                             | <b>46</b> |
| <b>Obrázek 8. Panel ovládacího prvku situační mapy. ....</b>                | <b>48</b> |
| <b>Obrázek 9. Situační mapa dle maximalizace zisku .....</b>                | <b>49</b> |
| <b>Obrázek 10. Ovládací panel pro práci s variantami .....</b>              | <b>50</b> |
| <b>Obrázek 11. Výsledek optimalizace variant.....</b>                       | <b>51</b> |
| <b>Obrázek 12. „Reálná“ data v systému .....</b>                            | <b>52</b> |
| <b>Obrázek 13. Jedno fixní depo, a 5 nových potenciálních dep.....</b>      | <b>53</b> |
| <b>Obrázek 14. Aktuální výchozí situace bez zobrazení popisků.....</b>      | <b>53</b> |
| <b>Obrázek 15. Grafické znázornění výsledku bez zobrazení zakázek. ....</b> | <b>54</b> |



# Obsah

|   |           |
|---|-----------|
| seznam obrázků .....  | 8         |
| <b>1. Cíl práce a motivace firmy pro optimální rozmístění dep .....</b> | <b>11</b> |
| 1.1 Cíl práce.....  | 11        |
| 1.2 Motivace firmy pro optimální rozmístění dep.....                    | 11        |
| 1.3 Představení konkrétní firmy .....                                   | 14        |
| <b>2. Podnikání subjektu na trhu v konkurenčním prostředí .....</b>     | <b>16</b> |
| 2.1 Stávající konkurence na trhu .....                                  | 17        |
| 2.2 Potenciální konkurence firmy.....                                   | 17        |
| 2.3 Konkurenční výhoda optimální lokace dep v dopravní síti.....        | 18        |
| <b>3. Analýza dat pro optimalizaci .....</b>                            | <b>20</b> |
| 3.1 Maximalizace zisku .....  | 20        |
| 3.2 Minimalizace vzdáleností mezi jednotlivými depy .....               | 22        |
| <b>4. Definice základních pojmů z lokalizačních úloh.....</b>           | <b>23</b> |
| 4.1 Grafy v teorii grafů .....  | 24        |
| 4.2 Matematická definice grafu .....                                    | 25        |
| 4.3 Depo .....  | 25        |
| 4.4 Dopravní práce .....  | 25        |
| 4.5 Atrakční obvod depa.....  | 26        |
| 4.6 Prvotní atrakční obvod depa.....                                    | 26        |
| 4.7 Přidělený atrakční obvod depa .....                                 | 26        |
| 4.8 Excentricita vrcholu.....   | 27        |
| 4.9 Vážená excentricita vrcholu .....                                   | 27        |
| 4.10 Vzdálenostně optimální umístění depa na síti .....                 | 27        |
| 4.11 Vážená excentricita bodu $b$ hrany $(v_i, v_j)$ .....              | 27        |
| 4.12 Absolutní depo.....  | 27        |
| 4.13 Vážená excentricita množiny dep $D_k$ .....                        | 28        |
| 4.14 Vzdálenostně optimální umístění $k$ dep na síti .....              | 28        |
| 4.15 Vzdálenost vrcholu $u \in V$ od depa $d \in D_k$ .....             | 28        |
| <b>5. Lokační algoritmy.....</b>  | <b>29</b> |
| 5.1 Určení kvality algoritmu.....                                       | 29        |
| 5.1.1 Časová složitost algoritmů.....                                   | 30        |
| 5.1.2 Paměťová složitost algoritmů.....                                 | 31        |
| 5.2 Možnosti výpočtu vzdáleností v dvojrozměrném prostoru .....         | 31        |
| 5.2.1 Euklidovská vzdálenost .....                                      | 32        |
| 5.2.2 Rektilineární vzdálenost .....                                    | 32        |
| 5.2.3 Umístění jediného objektu.....                                    | 32        |
| 5.2.3.1 Weiszfeldův algoritmus .....                                    | 33        |
| 5.3 Diskrétní lokalizační algoritmy .....                               | 34        |
| 5.3.1 Iterativní algoritmus.....  | 35        |
| <b>6. Praktická část.....</b>   | <b>36</b> |
| 6.1 Představení vytvořeného SW .....                                    | 36        |
| 6.1.1 Popsání problému s využitím map a možnosti řešení.....            | 36        |
| 6.1.1.1 Zjištění měřítka z rastrové mapy .....                          | 38        |
| 6.1.1.2 Výpočet vzdálenosti dvou bodů na rastrové mapě .....            | 39        |
| 6.1.2 UML .....   | 40        |
| 6.1.2.1 Požadavky na SW .....   | 40        |
| 6.1.2.2 Případy užití.....  | 41        |
| 6.1.2.3 UML diagramy .....  | 41        |

|            |  |           |
|------------|--|-----------|
| 6.1.2.4    | UML diagram tříd.....                            | 42        |
| 6.1.2.4.1. | Uchovávání dat .....                             | 42        |
| 6.1.2.4.2. | Správci dat .....                                | 43        |
| 6.1.2.4.3. | Třídy reprezentující data.....                   | 44        |
| 6.1.2.5    | Lokalizační algoritmy.....                       | 44        |
| 6.1.2.5.1. | Algoritmus Brute force .....                     | 45        |
| 6.1.2.5.2. | Weiszfeldův algoritmus .....                     | 46        |
| 6.1.2.5.3. | Srovnání algoritmu Brute force a Weiszfeld ..... | 46        |
| 6.1.2.5.4. | Iterativní algoritmus.....                       | 47        |
| 6.1.3      | Situační mapa.....                               | 48        |
| 6.1.4      | Uživatelské přidávání variant .....              | 50        |
| 6.1.5      | Praktický příklad.....                           | 51        |
| 6.1.6      | Srovnání s realitou .....                        | 54        |
| 6.1.7      | Popis výsledného řešení.....                     | 54        |
| <b>7.</b>  | <b>Závěr .....</b>                               | <b>56</b> |
|            | <i>seznam tabulek:</i> .....                     | 58        |
|            | <i>seznam vzorců</i> .....                       | 59        |

# 1. Cíl práce a motivace firmy pro optimální rozmístění dep

## 1.1 Cíl práce

Cílem práce je vytvořit softwarovou aplikaci implementující vybrané lokační algoritmy pro společnost, která podniká v oblasti půjčování stavebních strojů. Hlavní funkce aplikace je určena pro zjištění optimálního rozmístění dep v dopravní síti. Do systému lze zadávat požadavky po službách a tyto údaje jsou využity k optimalizaci umístění nových dep. Naimplementovaný software umožňuje uživateli zadávat do systému objekty typu stroj, zákazník, depo, zakázka.

## 1.2 Motivace firmy pro optimální rozmístění dep

Zisk je definován jako kladný rozdíl mezi celkovým hrubým příjmem a náklady. Pro efektivní růst a zajištění stability podniku je nezbytné, aby firma dosahovala zisku. Snahou každého subjektu na trhu by měl být maximální zisk, toho může dosáhnout buď snížením nákladů nebo snahou co nejvíce zvýšit své příjmy. Poté je možné nejen efektivně, ale hlavně rychle a s většími možnostmi rozvíjet podnik.

Při rozvoji podnikání je důležité, aby investice do toho vynaložené byly co nejefektivněji využité a vrátily se v dalších příjmech a v nových příležitostech podniku. Každý zásah do strategie podnikání prosperující firmy je rizikem a je už jen na vedení, jak je schopné toto riziko analyzovat a připravit se na potenciální hrozby, které tím vzniknou. Ideální situace nastává, když se riziko blíží nule. Tato situace se téměř nevyskytuje, protože vždy s každou změnou strategie vzniká určité riziko.

Podnikání se v hrubém základu rozšiřuje na dvě možnosti, poskytování služeb nebo výrobků. Tato práce se bude zabývat

výhradně tou první možností a to je poskytování služeb. Konkrétně půjčování stavebních strojů. Obecněji by se to dalo shrnout jako podnikání v silniční dopravě. Cena pro objednavatele služby z tohoto typu podnikání se skládá ze dvou částí

- a) cena za vykonanou práci (měřeno v hodinách),
- b) cena za ujeté kilometry (měřeno v kilometrech),

Cena za ujeté kilometry bývá podstatně nižší a pohybuje se řádově v desítkách korun. Naproti tomu cena za vykonanou práci se pohybuje v rozmezí stovek až desetitisíců korun. Tato cena se odvíjí od typu objednaného stroje. Obě tyto ceny určují konečnou cenu služby pro zákazníka. Firmy se dělí do několika typů, například:

- a) státní
- b) soukromé

Cíle státních firem může být mnoho. Například je to maximální počet uspokojených zákazníků, ale ne za účelem velkého zisku, zisk pro subjekty tohoto typu není prioritou, především mají zájem poskytnout co největší spektrum služeb nebo výrobků a uspokojit tím velké množství zákazníků. Tyto subjekty kladou velký důraz analýzy a na pozitivní zpětnou vazbu ze strany zákazníků neboli feedback. Jejich strategie je ovlivňována především politickou situací a přidělenými penězi ze státního rozpočtu.

U soukromých firem je cílem především zisk. U tohoto typu firem, nezáleží tolik na aktuální politické situaci v zemi, pokud cílový typ zákazníka není stát. Rozdíl mezi oběma typy je především v marketingové strategii, která je odlišná vzhledem k cílům daného podniku.

Tato práce se bude zabývat výhradně soukromými subjekty a jejich snahou o maximalizaci zisku a optimalizaci jejího rozvoje. Již bylo řečeno, snahou firmy v soukromém vlastnictví je především maximalizovat zisk, ale snahou zákazníka je minimalizovat náklady. Proto zákazník vždy sleduje konečnou cenu a vybere si tu nabídku, která je pro něj nejvýhodnější jak z hlediska ceny tak z hlediska

kvality poskytované služby. Zákazník je tedy ovlivňován těmito faktory:

- cena za provedenou práci
- cena za ujetou vzdálenost
- časová vzdálenost
- kvalita poskytované služby
- kvalita komunikace a přístup zaměstnanců k zákazníkům
- šířka spektra poskytovaných služeb

Zákazník se řídí cenou služby a kvalitou poskytované služby. U některých zákazníků může být přednější kvalita služby než cena. Jak již bylo zmíněno, cena se odvíjí od 2 faktorů, a to cena za vykonanou práci a cena za ujeté kilometry. Následující příklad ukáže podíl ceny za kilometr v celkové ceně zakázky. Z příkladu 1.0 vidíme, že podíl ceny za kilometry na celkové ceně zakázky se může pohybovat kolem 47,88% a výše.

Příklad 1.0.: Zákazník má zájem o pronájem stavebního stroje s úctovací cenou 800 Kč/hod a cenou 42 Kč/km.

| Typ stroje | cena Kč/hod | cena Kč/km | počet hodin práce | počet km k místu zakázky | Celková cena zakázky | podíl ceny za km na celkové ceně |
|------------|-------------|------------|-------------------|--------------------------|----------------------|----------------------------------|
| AD 20      | 800         | 42         | 8                 | 30                       | 7660                 | 16,45%                           |
| AD 20      | 800         | 42         | 8                 | 70                       | 9340                 | 31,48%                           |
| AD 20      | 800         | 42         | 4                 | 30                       | 4460                 | 28,25%                           |
| AD 20      | 800         | 42         | 4                 | 70                       | 6140                 | 47,88%                           |

Tabulka 1. Přehled strojů a přibližné ceny.

Příklad 1.0 ukazuje, že na koncové ceně pro zákazníka má nemalý podíl cena za ujeté kilometry. Proto zákazník raději využije nabídku, která mu je geograficky blíže, než nabídka vzdálenější a tím samozřejmě i dražší. Je patrné, že v případě zájmu podniku o maximalizaci zisku je nutné, aby kladl důraz na minimalizaci vzdáleností k potenciálním zákazníkům. Z toho vyplývá, že v tomto typu podnikání se klade velký důraz na vzdálenost mezi depem

poskytovatele a místem zakázky. Neboli podnik má při rozvoji za cíl optimalizovat vzdálenosti mezi svými depy.

Představený problém by se dal definovat jako lokační problém neboli rozmisťovací problém. Cílem tedy bude rozmístit depa na dopravní síti takovým způsobem, aby obsluha zákazníků probíhala co nejefektivněji, neboli s co nejmenšími náklady pro firmu. Tyto problémy řeší lokační úlohy. A lokační úlohy jsou řešeny lokačními algoritmy. Podrobněji to bude rozvedeno v další části práce, ale prozatím bude stačit vědět, že pod pojmem lokační algoritmy se rozumí ucelený matematický postup využívající k výpočtu konkrétní informace zadané uživatelem s výsledkem umístění dep tak, aby minimalizovalo náklady na obsluhu zákazníků.

### ***1.3 Představení konkrétní firmy***

Tato práce bude využívat data z konkrétního subjektu na trhu. Pro jakoukoli práci je to velmi důležité, neboť bude zachycovat reálnou situaci vyskytující se na trhu. Jak již bylo zmíněno, práce se bude zabývat problémem optimálního umístění dep v dopravní síti na základě požadavků od zákazníků. Firma Mepo se zabývá půjčováním stavebních strojů a autodopravou, ve svém oboru působí již přes 10 let a se svým hlavním a zatím jediným sídlem v Královéhradeckém kraji. Tento podnik má zájem o expanzi na trhu, podnik disponuje 9 stroji, z toho 6 strojů je stavebních (autojeřábů) a 6 zaměstnanci, s ročním obratem do 10 mil. Kč. Následující tabulka 1.1 ukáže, o které stroje jde a uvede přibližnou účtovací cenu.

| <b>Typ autojeřábu</b>       | <b>Nosnost</b> | <b>cena manipulace<br/>(v Kč)</b> | <b>cena za km<br/>(v Kč)</b> |
|-----------------------------|----------------|-----------------------------------|------------------------------|
| AD 080 PV3S                 | 8 t            | 700,-                             | 36,-                         |
| AD20 Tatra 148              | 20 t           | 800,-                             | 42,-                         |
| AD 28 Tatra 815             | 28 t           | 1000,-                            | 44,-                         |
| GROVE GMK<br>2035           | 35 t           | 1300,-                            | 60,-                         |
| <b>Montážní<br/>plošina</b> |                |                                   |                              |
| MP 22 Liaz                  | dosah<br>22m   | 700,-                             | 34,-                         |
| <b>Přepravní stroje</b>     |                |                                   |                              |
| Liaz Valník                 | 6 t            | 400,-                             | 32,-                         |
| Avia kontejner              | 2,5 t          | 380,-                             | 22,-                         |

Tabulka 2. Přehled strojů a přibližné ceny.

V tabulce 1.1 jsou uvedeny pouze přibližné ceny, jelikož majitel si nepřál zveřejnit opravdovou cenovou nabídku. Tato firma patří podle následujícího rozdělení vyhotoveného dle EU na základě možnosti čerpání dotací do kategorie „mikrofirmy“.

|                      | <b>Počet zaměstnanců</b> | <b>Roční obrat nebo celková bilance</b> |
|----------------------|--------------------------|---|
| <b>Mikrofirmy</b>    | do 10 zam.               | do 2 mil. EUR,                          |
| <b>Malé firmy</b>    | do 50 zam.               | do 10 mil. EUR,                         |
| <b>Střední firmy</b> | do 250 zam.              | do 50 mil. EUR,                         |

Tabulka 3. Rozdělení firem do skupin dle Evropské unie.

## **2. Podnikání subjektu na trhu v konkurenčním prostředí**

Při podnikání v určitém odvětví má vždy daný podnik ve svém okolí konkurenty podnikající ve stejném odvětví a nabízející stejné nebo podobné služby. Snahou majitelů podniků je přesvědčit zákazníka, aby využil jejich služeb nikoli konkurence.

Snahou každého subjektu na trhu je minimalizace hrozby ze strany konkurence. A to jak je tato snaha úspěšná je vidět zda-li podnik prosperuje či nikoli. Jeden z velkých problémů je hrozba ze strany konkurence, ale také s tím související problém, jak tyto hrozby minimalizovat.

U mikrofirem, ale také u malých firem, je důležité maximálně uspokojit získaného zákazníka tak, aby jakmile se odhodlá jednou využít služeb určitého podniku, ideálně, nikdy nepřešel ke konkurenci. Jelikož v oblasti silniční dopravy neexistuje takové množství potenciálních zákazníků jako například u služeb poskytování internetového připojení. Protože teoreticky každá domácnost s osobním počítačem je pro takového poskytovatele internetu potenciální zákazník.

Každé odvětví podnikání má své výhody a nevýhody a je už jen na vedení daného podniku jak výhody využije a ubrání se nevýhodám. Jedna z výhod v podnikání v oblasti poskytování stavebních strojů je taková, že nepodniká v přesyceném konkurenčním prostředí, které daný subjekt mohou výrazně ohrozit z důvodu omezenosti trhu. V České republice existuje jen několik středních a velkých podniků zabývajících se půjčováním stavebních strojů, například je to firma Hanyš, Metrostav, Švestka. Druhá zmiňovaná firma patří do kategorie nadnárodní a její spektrum poskytovaných služeb nezahrnuje jen půjčování autojeřábů, představený podnik je už spíše podnik stavební. Tyto zmiňované velké společnosti se zabývají projekty mnohem vyššího stupně než firmy patřící do kategorie mikrofirem a malých



firm. Proto je s určitou nadsázkou nemusíme brát jako hlavní konkurenci. Pod pojmem vyšší stupeň je myšleno pronajímání autojeřábů o nosnosti přes 100 tun, tyto jeřáby jsou jedinečné jednak svými možnostmi, ale hlavně jejich pořizovací cenou za nový autojeřáb, která se pohybuje v desítkách až stovkách milionů korun. Hlavní konkurence malých podniků a mikrofirem se vyskytuje v úrovni kraji kde firma působí. Na tento typ konkurence se podnik musí zaměřit, chce-li se udržet na trhu. V základě existují dva typy konkurence:

- a) stávající konkurence
- b) potenciální konkurence

### ***2.1 Stávající konkurence na trhu***

Stávající konkurence jsou firmy, které již působí na trhu v daném odvětví. Tyto firmy představují největší hrozbu, jelikož pokud stále existují, znamená to, že mají své zákazníky a že tito zákazníci využívají nabídek této společnosti. Bere-li se v úvahu snaha o vstup do územního sektoru, kde stávající konkurence existuje, což je téměř každý sektor, musí se daný subjekt nejdříve seznámit s hrozbami vstupu na zmíněný trh. A po vstupu nastavit takovou politiku podniku, aby přilákala co největší množství zákazníků. Tato práce se nebude zabývat analýzami trhu před vstupem do nové oblasti, jelikož bude vycházet z požadavků zákazníků po službách dané firmy. Tyto požadavky poskytne na základě dlouhodobého sledování společnost Mepo. Tento sběr dat bude popsán v další části této práce zabývající se konkrétním řešením problému lokace.

### ***2.2 Potenciální konkurence firmy***

Druhý typ je potenciální konkurence. Tato potenciální konkurence může být společnost podnikající ve stejném odvětví, společnost, která může daný podnik ohrozit vstupem do jejího územního sektoru. Vstup nové konkurence na trh vyžaduje mít

dostatečný finanční kapitál, jelikož cena stavebních strojů se odráží od specifických vlastností stroje jako je nosnost jeřábu, značka, typ podvozku atd. Řádově se nákup nového autojeřábu pohybuje od 3mil. Kč a výše. Ale i takto vysoká cena stroje samozřejmě nevylučuje vstup nové firmy na daný trh. Konkurenční firma může využít již opotřebované stroje, které jsou levnější dle typu stroje a dle velikostí a způsobu opotřebení.

Snahou firmy v soukromém vlastnictví za účelem maximalizace zisku je tedy minimalizace hrozeb ze strany konkurentů. Pro minimalizaci hrozeb ze strany konkurence je možné využít cenové strategie, komunikační strategie, reklamní strategie atd. V základu to znamená, že se podnik snaží minimalizovat hrozby svými výhodami u poskytovaných služeb oproti své konkurenci. Tyto výhody se také v některých literaturách nazývají konkurenční výhody. A jedna z největších výhod z odvětví poskytování stavebních strojů je výhoda optimální lokace dep na dopravní síti, bude jí věnována celá následující kapitola.

### ***2.3 Konkurenční výhoda optimální lokace dep v dopravní síti***

Obecně konkurenční výhoda je obvykle rozhodující faktor proto, zda-li podnikání bude úspěšné či nikoli. Mezi konkurenční výhodou lze zařadit faktory, které ovlivní rozhodnutí zákazníka, jestli využije služeb u dané společnosti nebo využije služeb konkurence. Faktory, které ovlivňují zákazníka, byly již zmíněny v kapitole 2.

Optimální lokace dep na dopravní síti patří mezi nejsilnější faktory ovlivňující rozhodnutí zákazníka. Protože z tohoto faktoru jsou odvozovány další neméně důležité konkurenční výhody jako například je:

- a) celková cena služby pro zákazníka
- b) výhoda blízkého depa – celkový čas od vyjetí stroje z depa do příjezdu na místo zakázky
- c) reklama na principu zákazník – zákazník

- d) velký počet potenciálních zákazníků vzdálených od sebe stejně daleko
- e) výhradní zastoupení firmy v dané lokalitě a z toho plynoucí velký tržní podíl společnosti v daném tržním prostředí
- f) vysoká konkurenceschopnost
- g) zavření trhu pro nově vstupující ale i stávající podniky z daného odvětví
- h) možné přemísťování strojů mezi jednotlivými depy s minimálními náklady

Firma nemůže zabránit existenci konkurence. Podnik se ale může pokusit získat takovou pozici na trhu v daném území, že stávající konkurenci se stane hrozbou a nové konkurenci nastaví takové vstupní podmínky, které si bez podstoupení velkého rizika nemůže dovolit. Jak již bylo výše zmíněno, optimální lokace dep ovlivňuje důležité vlastnosti služby, podle kterých se zákazník rozhoduje, zda-li využije danou firmu nebo využije služeb konkurence. Pokud tedy společnost vytvoří takovou síť dep na dopravní síti, která bude optimálně pokrývat určité vybrané území a zároveň bude disponovat takovým počtem strojů, který pokryje většinu objednávek zákazníků. Pak se společnost s takto zvolenou firemní strategií stane vysoce konkurenceschopnou. Protože cenu služby pro koncové zákazníky si bude moci zvolit tak, aby byla menší než u konkurence, ale zároveň aby to danému podniku zachovalo dobrý zisk.

### 3. Analýza dat pro optimalizaci

Při jakékoli optimalizaci vycházející z určité situace je nejdůležitější sběr dat a kvalita těchto dat, se kterými se bude následně pracovat. Z toho je zřejmé, že ani sebelepší optimalizace nepomůže, pokud se bude vycházet z dat, která budou nepravdivá, zavádějící, neúplná, neboli neužitečná. Při sběru dat je nutné dbát na kvalitu dat a na význam těchto informací. Předtím, než se začne s jakýmkoli získáváním informací, se musí nejdříve analyzovat situace a zjistit jaká data jsou pro manažery významná při dané optimalizaci.

Tato data se ale mohou lišit ve významnosti podle typu optimalizace, dokonce pro každou optimalizaci je možné mít úplně odlišný soubor dat. Optimalizací může být více, zde budou uvedeny jen ty, které budou využity v praktické části práce:

- a) maximalizace zisku
  1. dle vzdálenosti depa od zakázky
  2. dle podílu ceny za ujeté kilometry vůči ceně zakázky
- b) minimalizace vzdálenosti mezi depy

#### 3.1 Maximalizace zisku

Takovýto typ optimalizace je spíše pro rozvoj podniku, který již na trhu existuje. Protože je zde nutnost znát objednávky po službách daného subjektu, ať už splněné nebo nesplněné a z těchto požadavků vycházet. Samozřejmě, že lze vycházet i z fiktivních dat, která budou generována na základě průzkumu trhu. Společnost má za cíl rozmístit depa takovým způsobem, aby pokryla všechny objednávky od zákazníků a zároveň aby vzdálenost dep od zakázek byla co nejpříjemnější.

Při získávání dat je možné objednavatele rozdělit do dvou skupin: fyzický zákazník nebo firma. Pro fyzického zákazníka je charakteristická nepravidelnost objednávaných zakázek a krátká doba objednávaná práce. Tito zákazníci jsou na cenu zakázky nejvíce

náchylní. Oproti tomu společnosti jsou z hlediska potenciálu mnohem výhodnější. Mají větší finanční možnosti a větší pravděpodobnost opakované objednávky a delší dobu objednané práce. Dalším kritériem pro dobrou optimalizaci bude, zda-li je to fyzická osoba nebo firma. V atributech bude také její název nebo popis a kontakt na zákazníka z důvodu pozdější zpětné vazby.

Tabulka 4 ukáže rozdíl v cenách práce u jednotlivých typů strojů:

| Typ stroje | cena Kč/hod | cena Kč/km | počet hodin práce | počet km k místu zakázky | Celková cena zakázky |
|------------|-------------|------------|-------------------|--------------------------|----------------------|
| AD 080     | 700         | 36         | 8                 | 30                       | 6680                 |
| AD 20      | 800         | 42         | 8                 | 30                       | 7660                 |
| AD 28      | 1000        | 44         | 8                 | 30                       | 9320                 |
| AD 35      | 1300        | 45         | 8                 | 30                       | 11750                |

Tabulka 4. Přehled cen celkové práce pro různé typy strojů

U tohoto typu optimalizace je vidět jak je důležité zaznamenat jaký stroj je objednáván z důvodů rozdílných cen u jednotlivého stroje za hodinu práce a i za jednotku kilometru. Tato optimalizace se tedy bude řídit následujícími daty:

- a) typ objednávaného stroje
- b) místo zakázky (vzdálenost od depa)
- c) počet objednávaných hodin

Můžeme také využít informativní data, která mohou být:

- a) jméno zákazníka nebo firmy
- b) jeli to fyzická osoba nebo firma
- c) kontakt na příslušnou osobu

Výsledná tabulka dat bude vypadat následovně:

| Typ stroje        | AD 080     | AD 20       | AD 28       | AD 35        |
|-------------------|------------|-------------|-------------|--------------|
| počet hodin práce | 8          | 8           | 8           | 8            |
| jméno zákazníka   | Bak        | p. Vydražil | p. Nedorost | Krk. Vápenky |
| Adresa zákazníka  | adresa x   | adresa y    | adresa z    | adresa f     |
| je to firma       | ano        | ne          | ne          | ano          |
| kontakt           | bak@bak.cz | 606XXXXXX   | 606XXXXXX   | kvk@kvk.cz   |

Tabulka 5. Přehled užitečných dat

Z adresy zákazníka se vypočte vzdálenost k zakázce a z typu objednaného stroje se získá cena za hodinu práce a za jednotku kilometru.

### ***3.2 Minimalizace vzdáleností mezi jednotlivými depy***

Jak je z názvu patrné, při této optimalizaci se klade důraz na minimalizaci vzdálenosti mezi depy. Proto při této optimalizaci stačí znát pouze adresu depa.

## 4. Definice základních pojmů z lokalizačních úloh

Lokalizační úlohy řeší problém optimálního umístění objektu. Umístění těchto objektů je optimalizováno z hlediska optimalizačního kritéria, které závisí na charakteru úlohy a funkci objektu. Kritérium může zahrnovat náklady na umístění zařízení (cena pozemku, cena vybavení, vzdálenost od jiného objektu atd.) nebo také na maximalizaci poskytovaných služeb, což může znamenat například maximalizaci zisku. Na tato kritéria se lze dívat ze dvou hledisek, a to jak z hlediska zákazníka a tak z hlediska firmy. Pro zákazníka může být důležité něco jiného než pro podnik. Ale obecně by se zájmy společnosti měly podřizovat, či utvářet podle zájmů zákazníků, neboť zákazníci vytváří zisk.

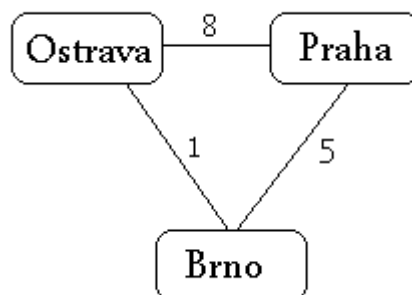
Problém optimálního umístění objektu v síti tedy spočívá v nejvhodnějším umístění jednoho či více objektů v síti vzhledem k jiným objektům. Objekty mohou být různého typu, například to mohou být depa, požadavky zákazníků, obchodní střediska, letiště atd. Vztah mezi jednotlivými objekty charakterizují vazby mezi těmito objekty. Mezi hlavní kritéria rozdělení lokalizačních úloh patří počet umisťovaných bodů, zda-li je to jeden nebo více bodů. Lokace také je možné rozdělovat podle umístění nového objektu, zda-li je závislé na umístění dalších nových objektů či nikoli, při tomto typu kritéria je možné aby se optimalizace řídila vzhledem ke statickým (neboli také již vytvořeným) objektům. Prostor, ve kterém tyto problémy řešíme, může být jedno až trojrozměrný. Tato práce se zabývá pouze dvojrozměrným prostorem tzn.  $(x, y)$ . Lokalizační problémy se také mohou rozlišovat podle množiny přípustných umístění nových objektů, ta může být diskrétní nebo spojitá a to s omezením při umisťování vkládaných objektu nebo bez nich. A v neposlední řadě podle měření vzdálenosti mezi zkoumanými objekty, tyto vzdálenosti se mohou měřit jako euklidovské nebo rektilineární, tato práce

využívá euklidovské měření vzdáleností. Tyto zmiňované odlišnosti v typech lokalizačních úloh zdaleka nejsou všechna, jsou zde zmíněna pouze ta, se kterými se tato práce zabývá v praktické části.

Při řešení lokalizačních problémů se využívá klasických aparátů matematické analýzy, matematického programování, optimalizace na grafech, ale také heuristické postupy. Algoritmy využívají vlastnosti z oblasti teorie grafů. V této kapitole budou nejdříve vysvětleny základní pojmy z teorie grafů a poté základní pojmy z lokalizačních úloh a na závěr budou představeny použité lokalizační algoritmy a jejich srovnání s jinými typy lokalizačních algoritmů.

#### 4.1 Grafy v teorii grafů

Pojem graf v teorii grafů neznamena žádný statistický graf a ani koláčový nebo sloupcový graf. Pod pojmem graf bude chápán zjednodušený reálný svět, který je složen ze dvou disjunktních prvků a to jsou body a přímky, které body spojují a tím popisují jejich vlastnosti. Body reprezentují vrcholy grafu, tyto vrcholy představují například města, křižovatky neboli řídicí prvky. Hrany reprezentují vztah mezi body. Každá hrana může mít své specifické vlastnosti jako je například ohodnocení hrany. Následující obrázek ukáže, jak může být takový ohodnocený graf reprezentován. Představuje 3 body a 3 hrany ohodnocené váhou reprezentující vzdálenost mezi jednotlivými body.



Obrázek 1. Ukázka reprezentace grafu



## 4.2 Matematická definice grafu

Jak již bylo řečeno, grafy jsou vhodným prostředkem pro popis situací, které lze znázornit konečným počtem bodů a hranami popisující vztah mezi nimi. Prostý graf je definován jako dvojice dvou množin:

- vrcholu ( $V$ )
- hran ( $H$ )

Graf je matematicky popsán:  $G = (V, H)$ , kde  $V$  je množina hran a  $H \subseteq V^2 \cup P_2(V) \cup V$  je množina hran grafu. Kde:

- $V^2 = V \times V = \{(v_1, v_2) \mid v_1 \in V, v_2 \in V\}$
- $P_2(V) = \{\{v_1, v_2\} \mid v_1 \neq v_2, v_1 \in V, v_2 \in V\}$  - neorientované hrany

Graf, jehož hrany nebo vrcholy jsou opatřeny hodnotami, je nazýván ohodnoceným grafem nebo sítí. Síť je libovolný orientovaný graf  $G = (V, H)$ , uvažovaný zpravidla s ohodnocením  $c: H \rightarrow \mathbf{R}_+$ , které nazýváme *kapacitou*. Existují i jiné typy grafů, ale těmi se tato práce nebude zabývat, popis lze najít ve veškerých literaturách zabývajících se teoriemi grafů.

## 4.3 Depo

Depo je místo na síti, z kterého je prováděna obsluha vrcholů a hran sítě. Depem jsou tedy nazývána střediska obsluhy např. sklady materiálu, garáže firem, atd. Obecně lze depo umístit do libovolného místa na síti, tedy na hranu nebo do vrcholu a v síti je možné umístit libovolný počet dep. Množina dep se značí  $D_k$ , kde počet dep je  $k = |D_k|$ . Pro  $k$  platí:  $1 \leq k \leq n$ , kde  $n = |V|$  neboli počet vrcholů.

## 4.4 Dopravní práce

Udává velikost přepravy, kterou je nutné provést při obsluze vrcholu  $u \in V$  nebo hrany  $h \in H$  obsluhované z depa  $d \in D_k$ . Při výpočtu této velikosti je nutné brát zřetel, že obsluhovací zařízení

musí dojet až na obsluhované místo a poté zpět po obsluhované cestě se vrátit do výchozího depa nejkratší cestou nebo stejnou cestou jen v opačném pořadí. Projetou vzdálenost násobíme ohodnocením vrcholu  $u \in V$  resp. hrany  $h \in H$ .

#### 4.5 Atrakční obvod depa

Atrakční obvod depa, značí se  $A(v)$  je taková množina vrcholů a hran, které jsou obsluhovány z jednoho depa  $p \in D_k$  pro které platí:

- vrchol  $v \in A(v)$ , pokud  $\nexists$  depo  $w \in D_k$ , pro které  $d(w, v) < d(p, v)$
- hrana  $h \in A(h)$ , pokud  $\nexists$  depo  $w \in D_k$ , pro které  $d(w, h) < d(p, h)$

#### 4.6 Prvotní atrakční obvod depa

$A'$  je množina vrcholů a hran sítě obsluhovaných z depa  $p \in D_k$ , pro které platí:

- vrchol  $v \in A'(v)$ , pokud  $\nexists$  depo  $w \in D_k$ , pro které  $d(w, v) \leq d(p, v)$
- hrana  $h \in A'(h)$ , pokud  $\nexists$  depo  $w \in D_k$ , pro které  $d(w, h) \leq d(p, h)$

Prvotní atrakční obvod depa  $A'(v)$  je podmnožinou atrakčního obvodu depa  $A(v)$ .

#### 4.7 Přidělený atrakční obvod depa

Přidělený atrakční obvod depa s označením  $A^*(p)$  je množina vrcholů a hran sítě obsluhovaných z depa  $p \in D_k$ , pro které platí následující vztahy:

- $A'(v) \subseteq A^*(v) \subseteq A(v)$
- $\bigcup_{p \in D_k} A^*(v) = X \cup V$
- $A^*(v) \cap A^*(m) = \emptyset$  pro  $v \neq m; v, m \in D_k$

#### 4.8 Excentricita vrcholu

Excentricita neboli výstřednost vrcholu  $v \in V$ , udává nejdelší vzdálenost vrcholu  $v \in V$  k vrcholu  $u \in V$  v síti  $G = (V, H)$  to znamená:

- $e(v) = \max \{d(v, u)\}$

#### 4.9 Vážená excentricita vrcholu

Vážená excentricita vrcholu  $v \in V$  s označením  $ec(v)$  udává excentricitu  $e(v)$  násobenou váhou vrcholu  $v \in V$ , to znamená:

- $ec(v) = \max \{d(v, u) \times w(v)\}$

Kde  $w(v)$  je označením pro váhu vrcholu  $v$ .

#### 4.10 Vzdálenostně optimální umístění depa na síti

Vzdálenostně optimální umístění depa na síti  $G = (V, H)$  se nazývá vrchol  $v^* \in V$  ve kterém, je hodnota vážené excentricity vrcholu  $v \in V$  minimální:

- $ec(v^*) = \min_{v \in V} \{e(v)\}$

#### 4.11 Vážená excentricita bodu $b$ hrany $(v_i, v_j)$

Vážená excentricita bodu  $b$  hrany  $(v_i, v_j)$  je definována jako maximum ze vzdáleností bodu  $b$  k vrcholu  $u \in V$  v síti  $G = (V, H)$  váženou ohodnocením vrcholu  $u$  což znamená:

- $ec(b) = \max \{d(b, u) \times w(u)\}$ , kde
- $d(b, u) = \{\min [e(v_i, b) + e(v_i, u), e(v_j, b) + e(v_j, u)]\}$
- $c_{ij} = e(v_i, b) + e(b, v_j)$

#### 4.12 Absolutní depo

Absolutní depo neboli úplné depo je takový bod  $b^* \in G$ , který je úplně vzdálenostně optimálním umístěním depa na síti a pro takovýto bod platí, že vážená excentricita bodu  $b$  hrany  $(v_i, v_j)$  je minimální:

- $ec(b^*) = \min \{ec_{b \in G}(b)\}$

#### 4.13 Vážená excentricita množiny dep $D_k$

Jestliže v síti  $G = (V, H)$  určíme množinu  $D_k \subset V$ , vážená excentricita této množiny je

- $ec(D_k) = \max_{v \in V} \{w(v) \times d(D_k, v)\},$

kde vzdálenost  $d(D_k, v)$  se určí jako

- $d(D_k, v) = \min_{v_i \in D_k} \{d(v_i, v)\}$

#### 4.14 Vzdálenostně optimální umístění $k$ dep na síti

Pro množinu vrcholů  $D_k^* \subset V$  platí:

- $ec(D_k^*, v) = \min_{D_k \subset V} \{ec(D_k)\}$

kde  $D_k$  jsou všechny  $k$  prvkové podmnožiny množiny  $V$ .

#### 4.15 Vzdálenost vrcholu $u \in V$ od depa $d \in D_k$

Vzdálenost vrcholu  $u \in V$  od depa  $d \in D_k$  je definována, jako délka minimální cesty:

- $D(u, v) = \min_{m(u,v) \in M} \{\sum_{m(u,v) \in M} o(h)\}$

kde  $M$  je množina všech cest mezi  $(u, v)$ .

## 5. Lokační algoritmy

Základní možnosti rozdělení lokačních úloh byly nastíněny v předešlé kapitole. Tato práce se zaměřuje na typ úloh rozdělené dle umístění nových dep. Umístění může být dvojího typu: diskrétní nebo spojité. Pokud se umísťuje nové depo pomocí spojitého algoritmu, tak výsledné umístění nového depa může být kdekoliv na mapě. U diskrétního algoritmu je výsledné depo vybíráno z konečné množiny potenciálních umístění. Rozdíl je tedy následující, spojitý algoritmus vybere optimální depo ze všech možností, kdežto diskrétní algoritmus vybere optimální depo pouze z takové množiny, které definuje uživatel. Druhý typ lokalizačních algoritmů je ten, který vybírá optimální umístění nikoli z celého prostoru, nýbrž pouze z dep, která si nadefinuje sám uživatel. Ale i takto definovaná množina může obsahovat velké množství prvků a tím i možností umístění nového depa. Čím je větší množina potenciálních umístění, tím je větší počet možných řešení. Stejně tak jako u příkladu existuje více cest ke správnému řešení, tak i algoritmy se mohou lišit svojí stavbou.

### 5.1 *Určení kvality algoritmu*

Algoritmus, jak již bylo řečeno, je možné posuzovat z několika hledisek. Základní hodnocení vyplývá z vlastností výsledného programu či aplikace. Protože pokud program je nefunkční, je nepravděpodobné, že použité algoritmy jsou správné. U algoritmů obvykle sledují dvě kritéria:

- časová složitost
- paměťová složitost

Obě tato kritéria jsou velmi odlišná, v ideálním případě je snahou nalézt takový algoritmus, který obě kritéria kombinuje s minimálními složitostmi. Důraz na minimalizaci jednotlivých složitostí se klade podle typu výsledné aplikace. Jiné nároky mohou

být pro aplikací běžící na PC (Personal computer), a jiné nároky mohou být pro aplikaci určenou výhradně pro mobilní telefony.

### 5.1.1 Časová složitost algoritmů

Časová složitost algoritmu vyjadřuje počet operací stroje, které je nutné vykonat po dobu provádění algoritmu. Není nutné znát přesný počet operací, ale spíš jde o obecné vyjádření trendu růstu počtů operací v závislosti na velikosti vstupních dat. Snaha je zjistit kolikrát bude delší doba zpracování, když vstupní data budou větší. Složitost závisí nejvíce na velikosti vstupních dat, proto jí je možné označit  $T = O(n)$ , kde  $n$  je velikost vstupních dat. Pro odhad složitosti jsou důležité jen funkční závislosti, než hodnota konstanty. Typy funkčních závislostí mohou být lineární, exponenciální, logaritmický atd. Snahou programátora je vyhnout se exponenciální složitosti nebo také složitosti  $n!$ , protože jakékoli i malé navýšení vstupních dat může znamenat obrovské navýšení dobu výpočtu neboli počet zpracovávaných operací.

|      | $\log_2(n)$ | $n \cdot \log_2(n)$ | $n^2$   | $n^3$    | $2^n$     | $n!$     | $n^n$    |
|------|-------------|---------------------|---------|----------|-----------|----------|----------|
| 5    | 2,321928    | 11,60964            | 25      | 125      | 32        | 120      | 3125     |
| 10   | 3,321928    | 33,21928            | 100     | 1000     | 1024      | 3628800  | 1E+10    |
| 20   | 4,321928    | 86,43856            | 400     | 8000     | 1048576   | 2,43E+18 | 1,05E+26 |
| 50   | 5,643856    | 282,1928            | 2500    | 125000   | 1,126E+15 | 3,04E+64 | 8,88E+84 |
| 100  | 6,643856    | 664,3856            | 10000   | 1000000  | 1,268E+30 | 9,3E+157 | 1E+200   |
| 200  | 7,643856    | 1528,771            | 40000   | 8000000  | 1,607E+60 | #NUM!    | #NUM!    |
| 400  | 8,643856    | 3457,542            | 160000  | 64000000 | 2,58E+120 | #NUM!    | #NUM!    |
| 800  | 9,643856    | 7715,085            | 640000  | 5,12E+08 | 6,67E+240 | #NUM!    | #NUM!    |
| 1000 | 9,965784    | 9965,784            | 1000000 | 1E+09    | 1,07E+301 | #NUM!    | #NUM!    |

Tabulka 6. Zpracovávaný čas dle složitostí v závislosti na počtu dat.

Pro obrovská vstupní data, nemusí mít optimální složitost algoritmu ani s lineární funkční závislostí. Existují 3 typy složitostí:

- složitost v průměrném případě
- dolní odhad složitosti
- horní odhad složitosti

Složitost v průměrném případě nás zajímá, pokud horní odhad složitosti dosahuje své předpovědi jen ve vzácných případech. Složitost v průměrném případě se počítá jako střední hodnota náhodné složitosti  $O(n)$  při určitém rozložení vstupních dat.

Dolní odhad složitosti nám udává ideální složitost daného algoritmu při určitém rozložení vstupních dat. Při dokazování dolního odhadu složitosti se musí dokázat, že neexistuje lepší algoritmus s lepší dolní složitostí.

Horní odhad složitosti zajímá programátora nejvíce, je to maximální možná doba provádění daného algoritmu. Podle této složitosti se rozhoduje, zda-li algoritmus je optimální nebo nikoli. Například  $T(n) = O(n^2)$  vyjadřuje kvadratickou horní složitost.

### **5.1.2 Paměťová složitost algoritmů**

Paměťová složitost algoritmů vyjadřuje míru využití paměti počítače při provádění algoritmu. Paměťová velikost kódu bývá zanedbatelná oproti paměti pro vstupní data. Paměťová složitost algoritmů je především závislá na použitých datových strukturách. Pokud algoritmus potřebuje paměť pouze pro vstupní data a nevyužívá další paměť počítače, tak algoritmus pracuje in-situ – na svém vlastním prostoru.

## ***5.2 Možnosti výpočtu vzdáleností v dvojrozměrném prostoru***

V následujících kapitolách bude nutné stanovit vzdálenost dvou bodů v dvojrozměrném prostoru. Protože algoritmy řešící úlohy optimalizace umístění středisek resp. objektů se mohou odlišovat podle použitého výpočtu vzdáleností. A tím i výsledky jednotlivých algoritmů mohou být různé. Tato práce se bude zabývat pouze výpočtem v euklidovském prostoru. Druhý typ výpočtu vzdálenosti je rektilineární. V následujících odstavcích budou obě možnosti představeny.

### 5.2.1 Euklidovská vzdálenost

Euklidovská vzdálenost bodů A a B je délka přímky mezi těmito body. V euklidovském prostoru existuje právě jedna přímka, která tyto dva body spojuje. Tato přímka představuje tzv. „vzdušnou čáru“.

$$d = d(A, B) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2} \quad [5.2.1.1]$$

[5.2.2.1] Výpočet Euklidovské vzdálenosti.

### 5.2.2 Rektilineární vzdálenost

Rektilineární vzdálenost je užívána tam, kde není možné spojení dvou bodů přímkou, tedy je nutné se pohybovat pouze ve směru os. V dvourozměrném prostoru to je osa x a y. Například to může být kanalizace, kabeláž ve výrobní hale atd. Vzdálenost  $d(A, B)$  je tedy určena jako:

$$d = d(A, B) = |A_x - B_x| + |A_y - B_y| \quad [5.2.2.2]$$

[5.2.2.2] Spojité lokalizační algoritmy

Při použití spojitěho lokalizačního algoritmu na vyhledání optimálního umístění nového depa v určité oblasti vždy nejvíce záleží na typu prozkoumávané oblasti. Pokud prozkoumávaná oblast je rastrový obrázek o velikosti  $800 \times 600$ , počet přípustných řešení je  $800 \times 600 = 480\,000$ . To znamená, že existuje 480 000 potenciálních nových míst. To znamená, že pro výběr 1 depa je nutné provést 480 tisíc operací. Výhoda této varianty je taková, že algoritmus najde skutečně to neoptimálnější řešení dané úlohy podle daných kritérií. Nevýhodou je velká časová náročnost. Z tohoto důvodu je nutné dát velký pozor na výběr algoritmu použitých při tomto typu lokalizace dep.

### 5.2.3 Umístění jediného objektu

Umístění jediného objektu, v euklidovském prostoru, v základě spočívá v nalezení minima kriteriální funkce:

$$f(x, y) = \sum_{i=1}^n \sqrt{(x - a_i)^2 + (y - b_i)^2} = \text{MIN} \quad [5.2.3.1]$$



## [5.2.3.1] Brute force algoritmus

Brute force neboli algoritmus hrubé síly, jak již název napovídá, tento algoritmus prochází celou zkoumanou oblast a hledá takový bod, který odpovídá zadanému kritériu. Tak tedy algoritmus hledá takový bod  $b \in G$ , kde  $G$  je matice bodů, reprezentující rastrový obrázek o rozměrech  $m \times n$ . A bod  $b = F_{krit}(G[m, n])$ . Kde  $F_{krit}$  je funkce, která vyjadřuje zadané kritérium podle kterého je daný bod vybrán.

**5.2.3.1 Weiszfeldův algoritmus**

Weiszfeldův algoritmus je určitou optimalizací algoritmu Brute force. Neprochází všemi body, nýbrž pomocí transformace předchozího vypočteného bodu získá souřadnice nového bodu. Tato posloupnost konverguje k optimálnímu řešení problému. Algoritmus končí pokud změna dvou po sobě vypočtených bodů je zanedbatelná. Algoritmus konverguje k řešení vždy, kromě případu, že všechny zkoumané body neleží v jedné přímce. Pro rychlejší konvergenci k řešení je jako počátečný bod vhodný nastavit těžiště vrcholů ze zkoumané množiny vrcholů.

$$x = \frac{\sum_{i=1}^n a_i \times w_i(x, y)}{\sum_{i=1}^n w_i(x, y)} \quad [5.2.3.1.1]$$

[5.2.3.1.1] Výpočet souřadnice x.

$$y = \frac{\sum_{i=1}^n b_i \times w_i(x, y)}{\sum_{i=1}^n w_i(x, y)} \quad [5.2.3.1.2]$$

[5.2.3.1.2] Výpočet souřadnice y.

$$w_i(x, y) = \frac{vaha_i}{\sqrt{(x-a_i)^2 + (z-b_i)^2 + e}} \quad [5.2.3.1.3]$$

[5.2.3.1.3] Výpočet  $w_i(x, y)$ .

Souřadnice x konečného bodu se získá dosazením do rovnice [5.2.3.1.4].

$$x^{(k)} = \frac{\sum_{i=1}^n a_i \times w_i(x^{(k-1)}, y^{(k-1)})}{\sum_{i=1}^n w_i(x^{(k-1)}, y^{(k-1)})} \quad [5.2.3.1.4]$$

[5.2.3.1.4] Výpočet souřadnice x.

Souřadnice y konečného bodu se získá dosazením do rovnice [5.2.3.1.5].

$$y^{(k)} = \frac{\sum_{i=1}^n b_i \times w_i(x^{(k-1)}, y^{(k-1)})}{\sum_{i=1}^n w_i(x^{(k-1)}, y^{(k-1)})} \quad [5.2.3.1.5]$$

[5.2.3.1.5] Výpočet souřadnice y.

Kde index (k) značí pořadové číslo iterace,  $k=1,2,3,\dots$

Jak již bylo zmíněno, pro rychlejší konvergenci k řešení je výhodné dosadit jako počáteční bod těžiště, souřadnice x se vypočítá dle vzorce [5.2.3.1.6] a souřadnice y se vypočítá dle vzorce [5.2.3.1.7].

$$x^{(0)} = \frac{\sum_{i=1}^n a_i \times vaha_i}{\sum_{i=1}^n vaha_i} \quad [5.2.3.1.6]$$

[5.2.3.1.6] Výpočet souřadnice x pro těžiště.

$$y^{(0)} = \frac{\sum_{i=1}^n a_i \times vaha_i}{\sum_{i=1}^n vaha_i} \quad [5.2.3.1.7]$$

[5.2.3.1.7] Výpočet souřadnice y pro těžiště.

### 5.3 Diskrétní lokalizační algoritmy

Jak již bylo zmíněno, výsledkem diskrétních lokalizačních algoritmů jsou taková depa, která nejlépe vyhovují zadaným kritériím. Tato depa jsou umístěna do míst, která jsou v množině možných umístění. Tuto množinu je vždy nutné před spuštěním algoritmu nadefinovat. Proto diskrétní algoritmy nemusí poskytovat optimální výsledek k celku, ale pouze k zadaným datům. V případě výběru 2 dep z množiny dep obsahující 5 potenciálních dep, složitost algoritmu je zanedbatelná, počet možných řešení je 20. Pokud ale množina potenciálních dep se zvýší na 500 objektů, počet řešení vzroste na

249°500. Z toho je patrné, že i při diskretních lokacích musíme použít takové algoritmy, které zadaný problém vyřeší v přijatelném čase.

### **5.3.1 Iterativní algoritmus**

Jak již bylo zmíněno, iterativní algoritmus patří do skupiny diskretních algoritmů. Výsledek iterativního algoritmu nemusí být vždy optimální vzhledem ke všem možnostem. Výsledek iterativního algoritmu je tedy nejvíce závislý na vstupních datech. Tento algoritmus nemusí vždy prozkoumat všechny možnosti. Z čehož vyplývá velká úspora časové i paměťové potřeby. Konkrétněji bude tento algoritmus bude popsán v praktické části.

## 6. Praktická část

V praktické části bude představen program optimalizující lokaci nových dep na rastrové mapě. Výsledná depa budou přehledně vyobrazena. Praktická část bude také popisovat nejdůležitější algoritmy a bude zde také představena existující firma z reálného tržního prostředí, na této společnosti bude ukázán sběr a analýza dat. Následně bude zhodnocen výsledek získaný pomocí představeného SW a tím spojené doporučení pro manažery společnosti.

### 6.1 Představení vytvořeného SW

Vytvořený program využívá přes 20 pomocných souborů a používá kolem 40 pomocných tříd. Zde nebudou popsány všechny třídy a soubory. V následujících odstavcích a kapitolách budou představeny pouze ty nejdůležitější. Vytvořený SW umožňuje uživateli zadávání dat do systému nebo jejich náhodné generování. Program umožňuje ukládat a načítat tato data ve formátu XML. A samozřejmě tento program umožňuje výpočet optimální lokace jednoho depa nebo více dep pomocí odlišných algoritmů. Výsledné řešení je přehledně ukázáno na rastrové mapě.

Než budou představeny vlastní lokační algoritmy, je nutné zmínit problém, který patří mezi nejzákladnější při tvorbě programu využívající mapy.

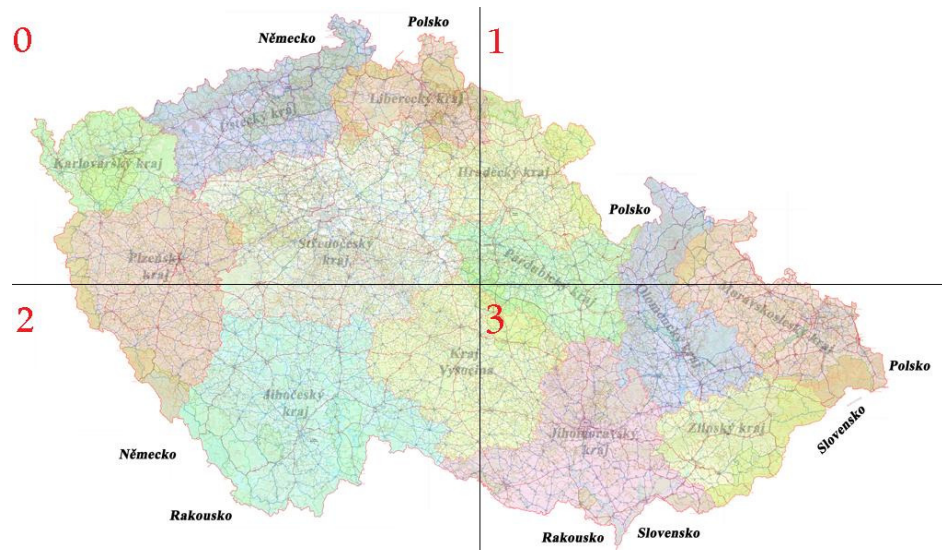
#### 6.1.1 Popsání problému s využitím map a možnosti řešení

U každého programu je nejdůležitější přehledné znázornění výsledků, protože pokud by dané výsledky byly správné, ale nesprávně nebo nepřehledně zobrazené, dané optimální řešení ztrácí na své kvalitě.

Nejdůležitější je zjistit účel těchto map. Pokud se od SW vyžaduje znázornění nejkratší cesty z místa A do místa B, je nutné mít takové mapové podklady, které toto umožňují. Tedy využívaná mapa musí být vektorová. Tzn. silnice musí být určitým způsobem

matematicky popsány aby z nich bylo možno zjistit vzdálenost, tvar trasy atd. Pokud tedy chceme využít mapy s vektorovým podkladem, možností je několik. Takové mapy si můžeme koupit, cena takových map se pohybuje od 100 000,- výše. Což pro studenty nebo i malé podnikatele nepřipadá v úvahu. Další možnost je využít například Google API. Tzn. využití již existujících map a funkcí na nich a pouze zprostředkovat napojení na takové mapy. Pokud se pomine právní politika, tak problémy nastávají i při dalších činnostech. Konkrétně Google API umožňuje pouze 1000 požadavků denně a při jakékoli změně ze strany Google API je nutné měnit i SW, který tento systém využívá. Proto je nejvýhodnější nalezení vlastního řešení.

Pro tento SW byl využit systém vycházejícího z principu QuadTree. Tento systém je založen rozdělování oblasti do pravidelných kvadrantů. Následující obrázek představuje, jak tento systém pracuje. Každá tato oblast je označena určitým kódem dle jejího kvadrantu. Tudíž některý kód oblasti může vypadat následovně 00132.



Obrázek 2. Princip QuadTree

Každá oblast ohraničená černou přímkou umožňuje znovu rozdělit danou oblast na 4 části. Konkrétně při využití tohoto principu byla původní mapa o velikosti  $17000 \times 10000$  pixelů. A jen pro

představu, po rozřezání vzniklo 341 souborů. Tento systém umožňuje jednoduše přibližovat nebo oddalovat danou oblast. Jak je vidět na obrázku 3.



Obrázek 3. Max. přiblížení využívající systém QuadTree.

#### 6.1.1.1 Zjištění měřítka z rastrové mapy

V případě, že není známo měřítko mapy a přesto je požadavek pracovat s reálnými vzdálenostmi, musí se zjistit měřítko mapy. Měřítko mapy může být například 1000:1. To znamená, že jeden centimetr na mapě je 1000 centimetrů ve skutečnosti. Co když ale nejsou známy žádné údaje o mapě a přesto je nutné například počítat vzdálenost dvou bodů na mapě?

Existuje několik možností, jak to zjistit, zde bude představena jedna z nich.

- a) je nutné změřit vzdálenost dvou bodů na zkoumané mapě (Mapová vzdálenost = MV)
- b) zjistit skutečnou vzdálenost těch samých bodů jako v bodě a), tomu se říká (reálná vzdálenost = RV)
- c) pozor RV a MV musí být ve stejných délkách
- d) měřítko se vypočítá dle vzorce [6.1.1.1.1]

$$\text{měřítko} = 1:X, \text{ kde } X = RV/MV \quad [6.1.1.1.1]$$

#### [6.1.1.1.1] Výpočet měřítka

Příklad:

$$MV = 4.32 \text{ cm}$$

$$RV = 2.16 \text{ km (216000 cm)}$$

$$\text{Měřítko } 1:X, \text{ kde } X = 216000/4.32 = 50000$$

Hledané měřítko je tedy 1:50000.

#### 6.1.1.2 Výpočet vzdálenosti dvou bodů na rastrové mapě

V případě, že se místo vektorových map používají rastrové, je programátor vystaven problému výpočtu vzdálenosti dvou bodů. Na rastrové mapě jak již bylo řečeno, je vzdálenost dvou bodů přímka. U vektorové mapy by to byla sada cest, které by tvořily křivku. Každá část neboli cesta křivky by měla své ohodnocení z hlediska délky. Což by pro výpočet vzdálenosti dvou bodů bylo ideální. Rozdíl je tedy patrný na první pohled, přímka bude vždy kratší než křivka. U rastrové mapy nelze změřit vzdálenost tak, aby odpovídala skutečné vzdálenosti. Ale můžeme se té vzdálenosti alespoň přiblížit. Následující tabulka nám ukáže rozdíly mezi vzdálenostmi dvou bodů.

| MV     | RV   | Podíl    |
|--------|------|----------|
| 21,873 | 31,6 | 1,444704 |
| 10,215 | 11,5 | 1,125795 |
| 12,44  | 18,6 | 1,495177 |
| 20,08  | 26,7 | 1,329681 |
| 33,26  | 43,8 | 1,316897 |
| 28,175 | 36,1 | 1,281278 |
| 41,102 | 51,5 | 1,25298  |
| 31,653 | 39   | 1,232111 |
| 53,873 | 67,4 | 1,251091 |

Tabulka 7. Přehled vzdáleností dvou bodů na mapách.(v km)

MV je vzdálenost mapová (neboli vzdálenost na rastrové mapě) a RP je vzdálenost reálná (neboli skutečná cesta mezi dvěma body po silnici). A podíl je RV/MV.

Tento podíl hodnot udává kolikrát je RV větší než MV. Průměr z těchto hodnot je roven 1,303. Proto vzdálenost dvou bodů se bude počítat dle následujícího vzorce:

$$\text{Celk. vzdálenost} = \text{VzdálenostDvouBodu} * 1.303. \quad [6.1.1.2.1]$$

[6.1.1.2.1] Výpočet vzdálenosti dvou bodů.

## 6.1.2 UML

UML (Unified Modeling Language) slouží k zjednodušení vývoje aplikace. Tento nástroj používá sjednocený jazyk. Protože právě jedním z cílů vývojářů UML bylo sjednocení výrazových prostředků. Při vývoji SW aplikace se jazyk UML může využít již od začátku. Od sběru požadavků po koncové nasazení aplikace. Tato práce nemá za cíl využít a popsat všechny jeho části. Budou zde uvedeny jen ty důležité. Také je nutno říci, že jazyk UML je sice unifikovaný, ale vysvětlení například pojmu „objekt“ může být v každé literatuře odlišný. Proto vždy nejvíce záleží na návrháři a jeho chápání světa v jazyce UML.

### 6.1.2.1 Požadavky na SW

Při vývoji aplikace je nejdůležitější si uvědomit jaké jsou na ni kladeny požadavky. V základu se dají požadavky rozdělit na dvě části: funkční a nefunkční. Mezi funkční požadavky patří například to, co má zadaná aplikace umět. A nefunkční požadavky jsou požadavky na systém. Například aplikace má běžet 24 hodin denně 365 dní v roce. Pro správnost a kvalitu požadavků je nejdůležitější sběr těchto požadavků. Tato práce vychází z požadavků uvedených v zadání. Funkční požadavky byly od zadavatele práce následující:

- grafické znázornění situační mapy
- možnost zadávání dat do systému

Nefunkční požadavky byly od zadavatele práce následující:

- implementace lokačního algoritmu optimalizující rozvoj firmy
- grafické znázornění situační mapy



- export/import dat ve formátech XML

### 6.1.2.2 Případy užití

Případy užití neboli Use Case v základu představují oddělení systému od jeho okolí. Případy užití mohou mít textovou a grafickou podobu. Slouží k rychlé představě o funkcích systému. Případy užití vycházejí z požadavků na SW aplikaci. V UML diagramech se zachycují případy užití, aktéry a jejich vztahy. Pod pojmem aktér je myšleno uživatel provádějící daný případ. Případ užití v UML diagramech je ovál s názvem případu užití.

Aktér “manažer“ využívá systém k přidávání dat do systému, exportu a importu dat ve formátech XML a provedení optimalizace rozvoje firmy. Jednotlivé případy užití se mohou podrobněji rozvést, určit scénáře a alternativní scénáře jednotlivých případů užití. To ale není cílem této práce.

### 6.1.2.3 UML diagramy

V jazyce UML rozlišujeme 8 typů diagramů, neboli 8 různých pohledů na systém. Mezi těchto 8 patří:

- diagramy tříd a objektů
- modely jednání
- scénáře činností
- diagramy spolupráce
- stavové diagramy
- diagramy aktivit
- diagramy komponent
- diagramy nasazení

Každý z těchto pohledů na systém má svoji váhu pro úspěšné zvládnutí návrhu, analýzy, implementace a nasazení vyvíjeného systému. Čím větší je robustnost systému, tím je důležitější propracovávat jednotlivé části. Tato práce představí pouze diagram tříd a objektů, protože cílem této práce je především úspěšné implementování vybraného lokačního algoritmu, nikoli podrobné

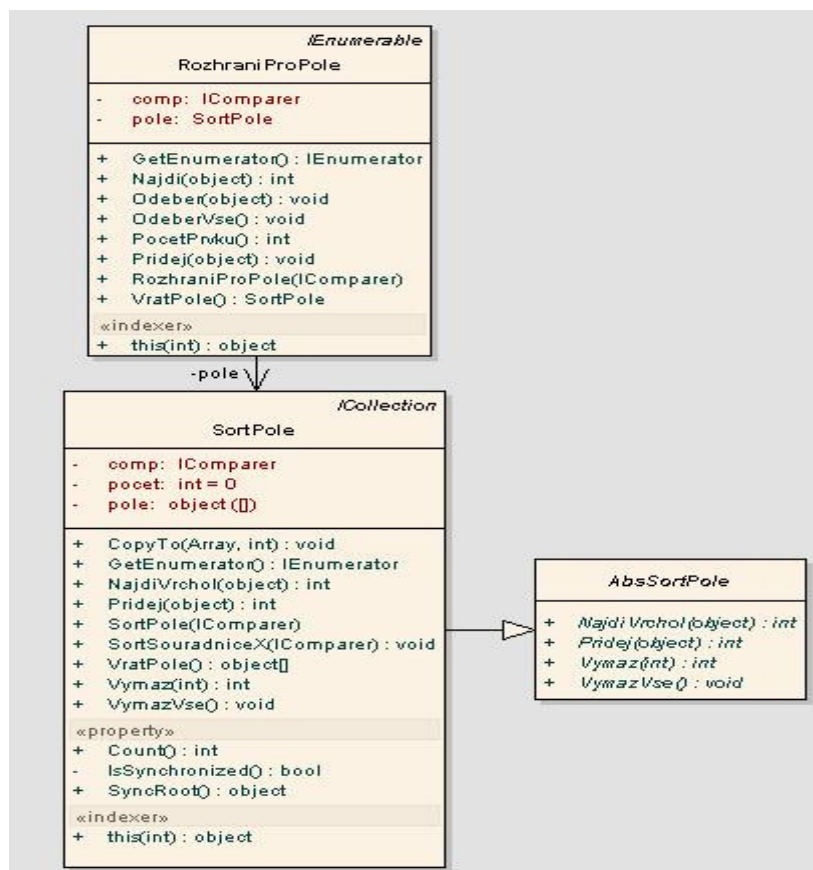
propracování jednotlivých částí vývoje systému za použití UML jazyka.

#### 6.1.2.4 UML diagram tříd

Diagramy tříd patří do částí zabývající se analýzou. Tzn. definování systému, sběru požadavků a prvotního návrhu modelu. V implementační části se podle diagramu tříd buduje, neboli implementuje daný systém. Jak již bylo řečeno, vytvořený SW využívá přes 20 vytvořených tříd a obsáhnout všechny třídy do jediného UML diagramu tříd je z hlediska velikosti stránky A4 a čitelnosti nemožné. Proto daný UML diagram bude rozdělen na několik částí a ty nejdůležitější třídy budou představeny podrobněji.

##### 6.1.2.4.1. Uchovávání dat

Základní struktura pro uchovávání dat je třída SortPole. Tato třída dědí od abstraktní třídy AbsSortPole její metody. Jak už z názvu vypovídá, jedná se o utříděnou strukturu. Následující obrázek ukazuje vztahy mezi jednotlivými třídami uchovávající a spravující data.



Obrázek 4. UML diagram tříd uchovávající data.

Pole má několik charakterizujících vlastností:

- všechny prvky pole existují od vytvoření po zrušení pole,
- prvky jsou zpřístupňováni na základě čísla udávající umístění od začátku pole,
- nedisponuje klasickou operací odeber.

Pole bylo vybráno na základě jeho charakteristik, ale i požadovaných vlastností programu. Časté vkládání, hledání a občasné mazání.

Operace vlož má asymptotickou složitost  $O(\log n)$ , stejně tak vyhledávání. Operace odeber má v nejhorším složitost  $O(n)$ , ale vzhledem k malému využívání operace odeber a vzhledem k jeho ostatním výhodám je tato složitost akceptovatelná.

Vyhledávání v poli je prováděno binárním algoritmem neboli metodou půlení. Tato metoda probíhá tak, že se vždy hledaný prvek porovná s prvkem uprostřed intervalu. Pokud se rovnají, algoritmus končí, pokud ne, dále je prohledávána horní oblast od prvku nebo dolní, to záleží na hledaném prvku. Toto se provádí dokud prvek není nalezen nebo zjištěn, že zadaný prvek neexistuje. Poté algoritmus končí. Tento algoritmus je také využíván k vložení nového prvku. A je již patrné, že tento algoritmus vyžaduje utříděná data.

#### 6.1.2.4.2. Správci dat

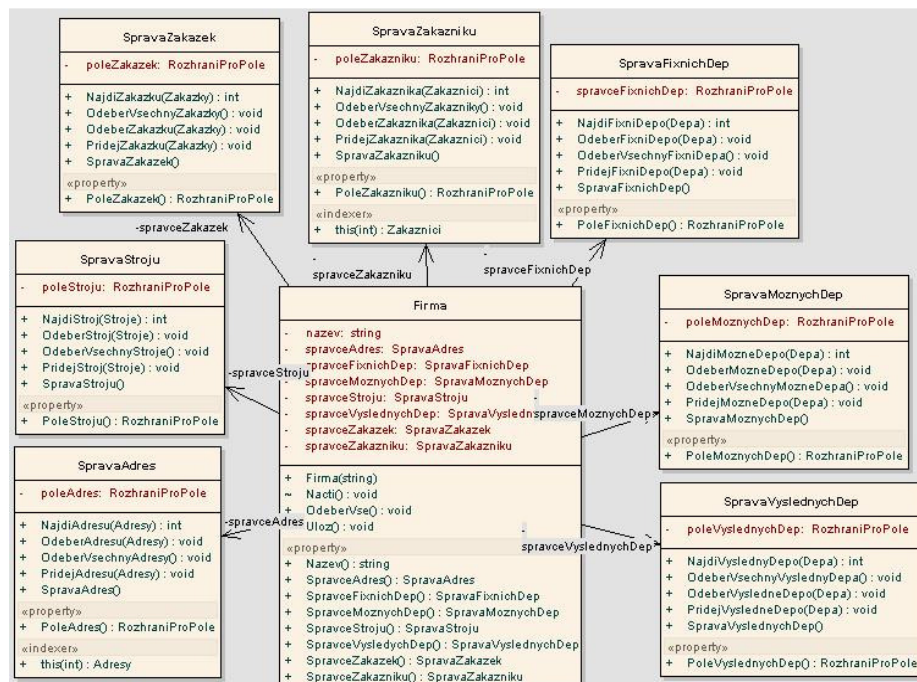
Pro správu dat jsou vytvořeny správci, kteří pracují s polem RozhraniProPole. Základní třída, která zpřístupňuje všechny správy, je třída Firma. Třída Firma je navržena tak, aby bylo možné přidat nového správce a funkčnost zůstala zachována. Správci spravují následující prvky:

- adresy
- zákazníci
- zakázky
- fixní depa

- možná depa
- výsledná depa
- stroje

Každý správce disponuje následujícími metodami:

- Najdi – najde hledaný prvek a vrátí index v utříděném poli daného prvku
- Odeber – odebere daný prvek z pole, pokud existuje
- Odeber vše – odebere všechny prvky v utříděném poli
- Přidej – přidá prvek na určité místo v utříděném poli



Obrázek 5. popisující ovládací prvky systému.

#### 6.1.2.4.3. Třídy reprezentující data

Mezi třídy reprezentující data patří třída Adresy, Zakaznici, Zakazky, Stroje a Depa. Tyto třídy jsou navrženy dle požadavků zadávajícího podniku. Tyto třídy budou představeny spolu s analýzou a sběrem užitečných dat v další části práce.

#### 6.1.2.5 Lokalizační algoritmy

Tato část podrobněji popíše a představí třídy zabývající se optimalizací lokace dep. Ukáže tabulky rychlosti výpočtu jednotlivých

algoritmů a na závěr bude představen praktický příklad a na něm otestovány jednotlivé algoritmy.

#### 6.1.2.5.1. Algoritmus Brute force

Brute force, neboli algoritmus hrubé síly. Teoreticky byl tento algoritmus popsán v kapitole 6.3.1.1., v této kapitole bude popsán způsob implementace tohoto algoritmu. Jak tato třída vypadá znázorňuje obrázek 6.

| BruteForce |   |
|------------|---|
| -          | poleVrcholu: RozhraniProPole              |
| -          | velikost: Size                            |
| <hr/>      |   |
| +          | BruteForce(RozhraniProPole, Size)         |
| +          | SpocitiNejlepsi(bool) : Point             |
| -          | VzdalenostDvouBodu(Point, Point) : double |

Obrázek 6. Třída algoritmu Brute force

Pro algoritmus Brute force je nejdůležitější výpočet vzdálenosti mezi zkoumaným bodem a místem zakázky. Tento požadavek reprezentuje *VzdalenostDvouBodu*, funkce počítá vzdálenost dvou bodů v euklidovském prostoru. To znamená, využití vzorce 1.0 z kapitoly 6.2.1. Výhoda tohoto algoritmu je v prověření všech variant, tudíž algoritmus vyhledá celkového optimálního řešení. Nevýhoda tohoto algoritmu spočívá například ve vložení dvou bodů. Je nutné ošetřit, aby nalezené minimum se nenalézalo v umístění zakázky. Tento typ problému je závislý na počtu desetinných míst využívaných v porovnávání nákladových funkcí. Nákladová funkce tedy vyjadřuje vzdálenost zkoumaného bodu s umístěním všech zakázek. Velikost nákladové funkce se počítá jako:

$$2 * \text{VzdalenostDvouBodu} \qquad [6.1.2.5.1]$$

[6.1.2.5.1]Výpočet velikosti nákladové funkce.

### 6.1.2.5.2. Weiszfeldův algoritmus

Podle teoretických znalostí uvedených v kapitole 5.3.1.2. nyní bude představena část praktická. Tedy implementace daného algoritmu a jeho třída. Jak tato třída vypadá znázorňuje obrázek 7

| Weiszfeld |   |
|-----------|---|
| -         | dleCeny: bool                           |
| -         | chyba: double = 0.0005                  |
| -         | pole: RozhraniProPole                   |
| -         | G(double, double, Zakazky) : double     |
| -         | Teziste() : Bod                         |
| -         | VzdalenostDvouBodu(Bod, Point) : double |
| +         | Weiszfeld(RozhraniProPole)              |
| +         | WeiszfeldAlgorithm(bool) : Bod          |

Obrázek 7. Třída algoritmu Weiszfeld.

Popis algoritmu je jednoduchý, nejdříve se vypočte souřadnice těžiště zakázek, která se vypočte dle vzorců v kapitole 5.3.1.2. A následně se souřadnice těžiště dosadí do funkce G viz. obrázek 7. Funkce G vypočte koeficient posunu směr nové, lépe konvergující, souřadnice. Tento koeficient se vynásobí původními souřadnicemi a výsledkem jsou nové souřadnice depa. Pokud rozdíl mezi souřadnicemi původního depa a nově vypočteného depa je menší než  $\varepsilon$ , což je námi definovaná velikost, algoritmus končí, pokud tomu tak není, algoritmus pokračuje. Nadále se již nepočítá nové těžiště, ale dosazuje se nově vypočtený bod.

### 6.1.2.5.3. Srovnání algoritmu Brute force a Weiszfeld

Rozdíl v těchto dvou algoritmech je patrný, náročnost algoritmu Weiszfeld je oproti Brute force minimální, jak je vidět z následující tabulky. Výsledky vždy konvergují k téměř stejnému výsledku, s rozdílem několika pixelů, což je pro náš případ přijatelná odchylka. Proto jednoznačné doporučení je využívat algoritmus Weiszfeld, pokud je náš cíl zjištění jediného depa.

|      | vykonávaný čas [ms] |          | počet operaci |          |
|------|---------------------|----------|---------------|----------|
|      | Brute Force         | Weisfeld | Brute Force   | Weisfeld |
| 20   | 8145                | 2        | 612000        | 3        |
| 50   | 21977               | 4        | 612000        | 10       |
| 100  | 42560               | 7        | 612000        | 15       |
| 500  | 213227              | 13       | 612000        | 24       |
| 1000 | 439955              | 29       | 612000        | 35       |

Tabulka 8. Srovnání algoritmů Brute Force a Weisfeld

#### 6.1.2.5.4. Iterativní algoritmus

Iterativní algoritmus, jak již bylo řečeno, patří do diskrétních lokačních algoritmů. Tzn., vybírá optimum z konečného počtu možností. Iterativní algoritmus neprochází všechny možnosti. Z toho plyne jeho největší výhoda. A výsledkem je optimální řešení vzhledem ke kriteriím. Iterativní algoritmus využívá opakování cyklů. V případě, že bychom nepoužili iterativní algoritmus, počet kombinací řešení by se počítal dle následujícího vzorce [6.1.2.5.4.1]

$$C_k(n) = \binom{n}{k} = \frac{n!}{k!(n-k)!} \quad [6.1.2.5.4.1]$$

[6.1.2.5.4.1] Výpočet počtu možností.

V této kapitole nebude popsána přesná implementace algoritmu, protože vše je patrné ze zdrojového kódu v programu. Budou zde popsány jen důležité části tohoto algoritmu. Mezi nejdůležitější funkce daného algoritmu patří výpočet nákladové funkce. Ta je ovlivněna především kriteriem, podle kterého se hodnota vypočítává. Například zda-li je to maximalizace zisku nebo minimalizace vzdáleností mezi depy. U maximalizačních kriterií je podstatné kolik zakázek je vzhledem k podmínce schopno obsloužit. U minimalizace vzdáleností je důležitá pozice na mapě a způsob výpočtu vzdálenosti mezi dvěma body. Ale to již bylo popsáno v předchozích kapitolách.

Iterativní algoritmus vždy pro vybraná depa spočítá zmiňovanou nákladovou funkci, pokud je tato funkce z hlediska kritéria lepší než doposud optimální, uloží se do paměti spolu s polem dep pro danou hodnotu nákladové funkce. Zjednodušeně řečeno, iterační algoritmus

se opakuje tak dlouho, dokud je v daném kroku nalezeno lepší optimum než doposud naleznuté.

### 6.1.3 Situační mapa

Jelikož iterativní algoritmus nedokáže sám naleznout místo pro optimální umístění depa, ale pouze posuzuje, jaké rozmístění dep je optimální z konečné množiny dep. Je důležité, aby vkládaná depa byla již umístěna tak, aby odpovídala určitým kritériím. Jak již bylo zmíněno, aplikace umožňuje optimalizaci dep na základě několika kritérií. A jsou to:

- a) Dle vzdálenosti zakázky od depa
- b) Dle podílu ceny za ujeté kilometry vůči celkové ceně (z pohledu zákazníka).
- c) Dle minimalizace vzdáleností mezi depy.

Jak ale poznat, že námi zadané umístění depa je výhodnější než náhodné rozmístění? K tomuto účelu je v aplikaci možnost zobrazení situační mapy. Situační mapa může být 3 typů. Již výše zmíněné typy a), b), a také kombinace těchto dvou typů. Ovládací panel je vidět na obrázku 8. Jak je vidět z obrázku, lze vybrat kritérium, dle kterého bude situační mapa vytvořena.

Situační mapa

Rozhodovací kritérium :

vzdálenost od depa  
30 km

cenový podíl  
5 %

Vytvořit situační mapu

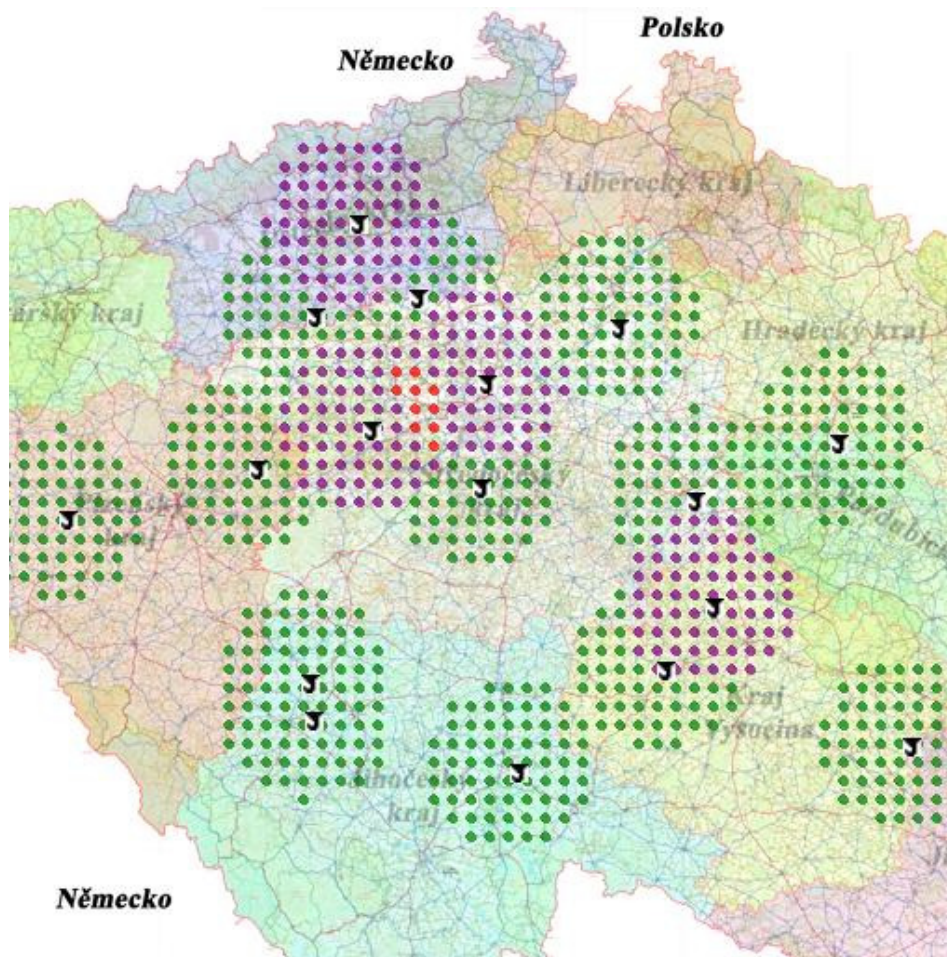
Smaž sit. mapu

Obrázek 8. Panel ovládacího prvku situační mapy.

Situační mapa je vytvořena jako síť bodů rozmístěných 5km (v reálné vzdálenosti) od sebe. Situační mapa funguje při jakémkoli



zoomu mapy. A barevné spektrum bodů znázorňuje míru výhodnosti lokace. Na mapě jsou zobrazeny pouze ty body, které mají význam, body s nulovou hodnotou dle kritéria nejsou zobrazeny a ani nejsou v poli bodů obsaženy. Zobrazení situační mapy je patrné na následujícím obrázku.



Obrázek 9. Situační mapa dle maximalizace zisku

Na mapě mohou být 4 typy oblastí:

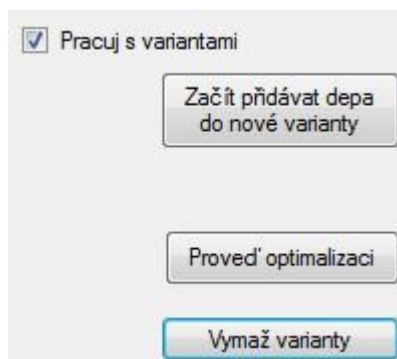
- a) Nepokrytá
- b) Zelená
- c) Fialová
- d) Červená
- e) Černá

Nepokryté území vyznačuje nulový potenciální zisk pro společnost. Zelené území vyznačuje prostor kde v případě umístění

depa firma má šanci na zisk do 25 000Kč. Fialová naznačuje potenciální zisk do 75 000Kč. A červená barva reprezentuje místa s potenciálním ziskem do 150 000Kč. Černá barva ukazuje prostor, kde by depo mohlo mít zisk nad 150 000Kč. Území označené černou barvou je tedy z hlediska maximalizace zisku optimální.

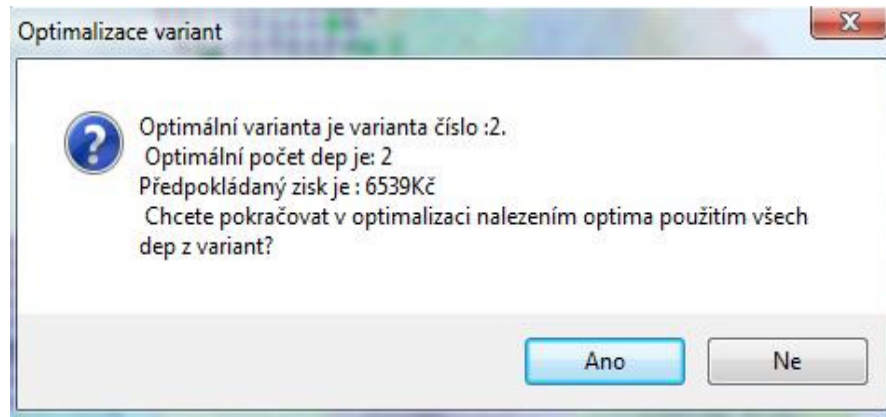
#### 6.1.4 Uživatelské přidávání variant

Pro rozšíření možnosti uživatele, neboli vedoucího pracovníka pověřeného rozvojem firmy, výsledná aplikace nabízí možnost zadat do systému více variant pro posouzení a následnou optimalizaci. Pro přidávání variant slouží následující panel. Z obrázku je vidět, že po zaškrtnutí checkboxu „Pracuj s variantami“ se zobrazí nejdůležitější tlačítka „Začít přidávat depa do nové varianty“. Poté můžeme začít přidávat depa, pro ukončení klikneme na tlačítka „Ukončit zadávání“ zobrazené pod tlačítkem se začátkem přidávání. Pokud je nějaká varianta v systému a systém také obsahuje nějaké zakázky, je možné začít s optimalizací.



Obrázek 10. Ovládací panel pro práci s variantami

Po kliknutí na tlačítka provést optimalizace se na dané varianty použije iterativní algoritmus. To znamená, že se z každé varianty vybere optimální počet dep a jejich optimální rozmístění na základě kritérií. Tyto výsledky se porovnají s ostatními a vybere se taková varianta, která má nejlepší výsledky dle kritéria. Po této optimalizaci je uživateli nabídnuta možnost optimalizovat i toto řešení, viz obr. 11.



Obrázek 11. Výsledek optimalizace variant

Pod touto optimalizací se rozumí, že jako výchozí rozmístění dep bude nastaveno předešlé optimální řešení. A dále se budou vybírat depa ze všech variant a bude hledaná taková kombinace dep, která zajišťuje nejlepší výsledky dle kritéria.

### 6.1.5 Praktický příklad

Jak již bylo řečeno firma Mepo na trhu se službami v Královéhradeckém kraji působí přes 10 let. Za tuto dobu si vybudovala pevnou pozici na trhu. Podnik si drží vysoký kredit svým působením na daném trhu. Protože vždy včas a odborně splní požadované práce. Ale také svojí „rodinnou atmosférou“. Ovšem po určitém čase je nutné společnost oživit, jinak její růst začne stagnovat. Proto vedení podniku začalo uvažovat o rozšíření působnosti. Zmiňovaná společnost vlastní 9 pracovních strojů, proto si může dovolit tyto stroje rozprostřít do více dep, aniž by to zákazník jakkoli poznal. Zadala tedy studii o tom, kde by bylo nejvýhodnější nové depo umístit. Společnost poskytla soukromé informace o zákaznících a zakázkách, které si nepřeje zveřejnit z důvodu obavy o zneužití tohoto souboru dat konkurencí. I přes tuto obavu firma dovolila použít určité data pod podmínkou, že budou změněna způsobem tak, aby nebylo možno jakkoli zneužít.

Po dohodnuté změně bylo společností Mepo do programu zadáno 20 zakázek cca od 10 firem. Počet těchto údajů je tak malý,

protože vedení podniku vybralo zakázky nejlépe odpovídající dlouhodobému pracovnímu nasazení. Firma Mepo nedovolila použít všechny zakázky z několika měsíčního sběru dat. Na následujícím obrázku je jejich grafická reprezentace a zároveň zobrazení situační mapy k těmto zakázkám. Firma si přeje provést následující studii.

- Studie pro jedno nové depo ke stávajícím depu.



Obrázek 12. „Reálná“ data v systému

Po zadání dat do systému je nyní důležité určit potenciální místa nových dep. Optimalizace vychází ze situace, že jedno fixní depo již existuje. Depo je umístěno dle reálných souřadnic a to ve vesnici Kunčice nad Labem. Společnost Mepo poskytla 3 souřadnice, kde by podle nich bylo potenciálně výhodné nové depo umístit.



Obrázek 13. Jedno fixní depo, a 5 nových potenciálních dep

Fixní depo je zobrazeno modře a možná depa jsou zobrazena oranžově. Firma Mepo by nejraději nové depo umístila do dvou lokací

- a) Umístění Nová Paka
- b) Umístění Trutnov

Obě tato místa byla zadána do systému. Na základě situační mapy byla také přidána depa do měst

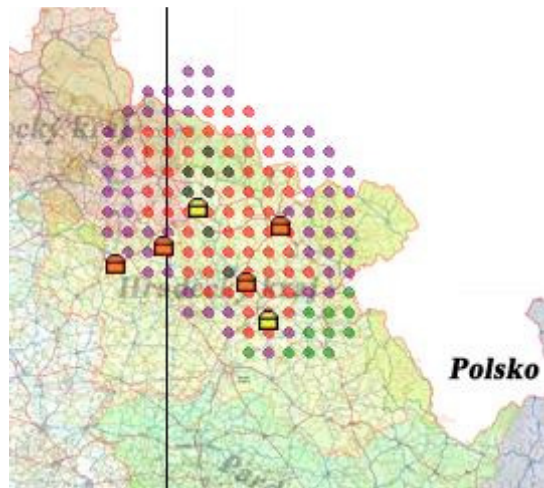
- c) Dvůr Králové nad Labem
- d) Nová Paka
- e) Jaroměř

Optimalizace byla nastavená pro výběr dvou dep na základě kritéria podílu z celkové ceny za ujetou vzdálenost do 10%. Těchto 10% firma Mepo dodala na základě svého výzkumu, který si nepřeje zveřejnit. Po zobrazení situační mapy pro dané kritérium situace vypadá následovně.



Obrázek 14. Aktuální výchozí situace bez zobrazení popisků.

Nyní je načase spustit optimalizaci, nejprve je vybrána varianta, aby optimalizace vypočítala 2 depa. Na základě kritéria pro podíl. Viz výše. Výsledek optimalizace je následující: Ani depo a), a ani depo b) nebylo vyhodnoceno jako optimální. Nýbrž depo e) Jaroměř s potenciálním ziskem 262 164Kč. Grafický výsledek optimalizace je tedy následující:



Obrázek 15. Grafické znázornění výsledku bez zobrazení zakázek.

#### 6.1.6 Srovnání s realitou

Společnost Mepo na základě vlastních zkušeností provedla samostatně studii bez použití jakýchkoli algoritmů. Konkrétní postup a data si firma nepřála zveřejnit. Po porovnání obou výsledků je závěr následující: bylo vybráno umístění optimálního depa do města Dvůr Králové nad Labem. Vzdálenost mezi oběma výslednými depy je více než 20km. Proto umístění do méně výhodné lokace může mít zásadní vliv na velikost poptávky zákazníků.

#### 6.1.7 Popis výsledného řešení

Jak již bylo řečeno, optimální umístění je v městě Jaroměř. Firmě Mepo bylo tedy doporučeno přehodnotit své studie, a zároveň jim byl navrhnout výsledek z iterativního algoritmu. Pořízení nového depa není levná záležitost. Z hlediska zákazníka může být matoucí, kdyby se umístění dep některé firmy měnilo. Proto je důležité

důkladně zvážit umístění nového depa. Výsledek zobrazuje také potenciální zisk. Tento údaj je informativní, nikoli však neúčinný. Pod touto hodnou jsi lze představit, jaká míra zisku se dá očekávat.

## 7. Závěr

Hlavní cíl diplomové práce bylo vytvoření softwarové aplikace implementující vybraný lokační algoritmus. Tato aplikace měla být využita jako pomocník při rozvoji firmy zabývající se půjčováním stavebních strojů.

Hlavní funkcionalita měla být optimalizace rozmístění dep na dopravní síti pomocí vybraného lokačního algoritmu.

Na základě zadaných požadavků stavební firmou byla vytvořena softwarová aplikace, která umožňuje uživateli zadat zakázky a potenciální nová depa do systému. A na tento soubor dat poté použít naimplementovaný algoritmus pro optimalizaci rozmístění nových dep.

Podstatou naimplementované aplikace je najít takové optimální rozmístění dep, aby dle potřeb maximalizovala zisk, nebo minimalizovala vzdálenost mezi depy.

Uživatel má možnost zadávat do systému zakázky, stroje, zakázky a umístění potenciálních, ale také již stávajících dep. Na základě zakázek lze graficky zobrazit situační mapu. Body situační mapy jsou barevně rozlišeny dle výhodnosti umístění depa do dané lokace dle daného kritéria. Jako kritérium lze zadat minimální vzdálenost zakázky od depa, nebo podíl ceny za ujetou vzdálenost strojem z celkové ceny zakázky.

Implementovaný iterativní algoritmus umožňuje optimálně rozmístit depa na mapě z diskrétní množiny dep dle zadaných kritérií.

Systém by v budoucnu mohl být rozšířen o funkce generující faktury dle zadaných zakázek. Nebo také o funkce zajišťující lepší správu objednávek a faktur ve firmě.

Vytvořená softwarová aplikace je již nasazena do provozu u zadavatele a v nejbližší době je firma Mepo rozhodnuta provést rozvoj firmy o nové depo s přihlédnutím na výsledky z této práce.



## Zdroje:

- [1.]ARLOW, Jim; NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací Objektově orientovaná analýza a návrh prakticky*. První vydání. Brno: Computer Press, a.s., 2007. 555 s. ISBN 978-80-251-1503-9.
- [2.]Návrh aplikací v jazyce UML - začínáme s případy užití RSS komentářů [online]. [cit. 2010-02-5]. Dostupný z WWW: <http://interval.cz/clanky/navrh-aplikaci-v-jazyce-uml-zaciname-s-pripady-uziti/>
- [3.]*Prostorová analýza dat* [online]. [cit. 2010-03-5]. Dostupný z WWW: [http://gis.vsb.cz/pad/Kap\\_5/kap\\_\\_5\\_5.htm](http://gis.vsb.cz/pad/Kap_5/kap__5_5.htm)
- [4.]Genesis upce[online]. [cit. 2010-03-15]. Dostupný z WWW: [http://genesis.upce.cz/priloha/dfjp\\_kid\\_4-LOKACE](http://genesis.upce.cz/priloha/dfjp_kid_4-LOKACE)
- [5.]Sdružení knihoven ČR [online]. [cit. 2010-03-25]. Dostupný z WWW: <http://www.sdruk.cz/sec/2006/sbornik/2006-3-270.pdf>
- [6.]Práce s mapou:Měření vzdáleností [online]. [cit. 2010-04-15]. Dostupný z WWW: <http://survival.specialista.info/view.php?cisloclanku=2006041701>
- [7.]*Metodiky a Notace-UML*[online]. Objektová analýza, návrh a programování, [cit. 2010-04-18]. Dostupný z WWW: <http://objekty.vse.cz/Objekty/MetodikyANotace-UML>
- [8.]Dileep r. Sule.Logistics of Facility Location and Allocation. První vydání. New York: Marcel Dekker, Inc. , 2001. 555 s. ISBN 0-8247-0493-2

**seznam tabulek:**

|   |           |
|---|-----------|
| <b>Tabulka 1. Přehled strojů a přibližné ceny. ....</b>                       | <b>13</b> |
| <b>Tabulka 2. Přehled strojů a přibližné ceny. ....</b>                       | <b>15</b> |
| <b>Tabulka 3. Rozdělení firem do skupin dle Evropské unie. ....</b>           | <b>15</b> |
| <b>Tabulka 4. Přehled cen celkové práce pro různé typy strojů ....</b>        | <b>21</b> |
| <b>Tabulka 5. Přehled užitečných dat.....</b>                                 | <b>22</b> |
| <b>Tabulka 6. Zpracov. čas dle složitostí v závislosti na počtu dat .....</b> | <b>30</b> |
| <b>Tabulka 7. Přehled vzdáleností dvou bodů na mapách.(v km) .....</b>        | <b>39</b> |
| <b>Tabulka 8. Srovnání algoritmů Brute Force a Weisfeld .....</b>             | <b>47</b> |

*seznam vzorců*

|  |    |
|--|----|
| [5.2.2.1] Výpočet Euklidovské vzdálenosti. ....      | 32 |
| [5.2.2.2] Spojité lokalizační algoritmy.....         | 32 |
| [5.2.3.1] Brute force algoritmus.....                | 33 |
| [5.2.3.1.1] Výpočet souřadnice x.....                | 33 |
| [5.2.3.1.2] Výpočet souřadnice y.....                | 33 |
| [5.2.3.1.3] Výpočet $w_i(x, y)$ .....                | 33 |
| [5.2.3.1.4] Výpočet souřadnice x.....                | 34 |
| [5.2.3.1.5] Výpočet souřadnice y.....                | 34 |
| [5.2.3.1.6] Výpočet souřadnice x pro těžiště. ....   | 34 |
| [5.2.3.1.7] Výpočet souřadnice y pro těžiště. ....   | 34 |
| [6.1.1.1.1] Výpočet měřítka. ....                    | 34 |
| [6.1.1.2.1] Výpočet vzdálenosti dvou bodů. ....      | 40 |
| [6.1.2.5.1] Výpočet velikosti nákladové funkce. .... | 45 |