



Certifikovaná metodika

**Nmet1 - Popis procesního modelu
automatizované technologie tvorby
map**

Výzkumný ústav geodetický, topografický a kartografický, v.v.i.
T-MAPY spol. s r.o.

Z D I B Y 2 0 1 6

Oponenti:

Doc. Ing. Václav Talhofer, CSc - Univerzita obrany, Brno

Ing. Radek Wildmann - Vojenský geografický a hydrometeorologický úřad, Dobruška

„Projekt č. TB04CUZK001 „Výzkum a vývoj metod pro kartografickou generalizaci státního mapového díla středních měřítek“ je řešen s finanční podporou TA ČR“.

Tato metodika je jedním z výstupů projektu TB04CUZK001. Certifikační protokol č.j. ČÚZK-17688/2016-22 byl vydán dne 8.12.2016.

T A

Č R

Program **Beta**

Obsah

1. Cíl metodiky	1
1 Teoretické základy automatizované generalizace	3
1.1 Kartografická generalizace	3
1.1.1 Geometrické vs. kartografické řešení	3
1.1.2 Strategie řešení konfliktů při generalizaci	3
1.1.3 Optimalizovaná kritéria	6
1.1.4 Omezující podmínky	11
1.2 Strategie řízení generalizace	13
1.2.1 Optimalizace	13
1.2.2 Omezující podmínky v optimalizaci	20
1.2.3 Zhodnocení optimalizační strategie	22
1.2.4 Batch Processing	23
1.2.5 Rule Based Generalization	24
1.2.6 Multi Agent Systems	25
1.2.7 Heuristika	27
1.3 Závěr	28
2. Stávajícího stav implementace automatizované generalizace	30
2.1 Rané fáze, 80-léta	30
2.2 Přechod k digitální kartolitografii, 90-tá léta	30
2.2.1 Generalizace založená na pravidlech	30
2.2.2 Generalizace založená na modelování	32
2.2.3 Generalizace založená na pravidlech nebo modelování?	33
2.2.4 Technologie	34
2.3 Po roce 2010	35
2.3.1 Kartografické reprezentace ArcGIS a Model Builder	35
2.3.2 Agilní vývoj a scrum	35
2.3.3 Schopnost postavit široký tým	36
3. Procesní model řešení generalizace	38
3.1 Generalizační modul M1	39
3.2 Generalizační modul M2	41
3.3 Generalizační modul M3	44
Literatura	48
Přílohy - 1 -	
Optimalizace rychlosti procesů a algoritmů	- 3 -
Související literatura	- 4 -
Ověření výkonu prostorového indexu	- 5 -
Výstupy testů	- 5 -
Závěry	- 5 -
Kód testovacího skriptu	- 7 -
Měření kvality kresby	- 9 -
Jednoduché měření	- 9 -
Složitější měření	- 11 -

1. Cíl metodiky

Tato metodika shrnuje výsledky výzkumu v projektu TB04CUZK001 „Výzkum a vývoj metod pro kartografickou generalizaci státního mapového díla středních měřítek“ v oblasti procesního modelu automatizované technologie tvorby map. Na základě průzkumu řešené problematiky v jiných zemích a národních mapovacích agenturách navrhuje nejvhodnější procesní model automatizované tvorby topografických map s aplikací technologií automatizované technologie kartografické generalizace pro obor topografických map středních měřítek. Navrhuje dekompozici systému na dílčí podsystémy, popisuje jejich funkci a procesy, vzájemné vazby a časové posloupnosti a vstupní a výstupní rozhraní mezi jednotlivými podsystémy.

Rozsah řešení je primárně stanoven na řešení automatizované generalizace při tvorbě Základních map České republiky středních měřítek 1 : 10 000 a 1 : 25 000 (ZM 10 a ZM 25), je ovšem dostatečně obecný, aby byla možné jeho uplatnění i pro tvorbu map dalších měřítek.

Tato metodika navazuje na výsledky řešení dalších úkolů, které jsou také součástí projektu TB04CUZK001 „Výzkum a vývoj metod pro kartografickou generalizaci státního mapového díla středních měřítek.“

Metodika je rozdělena do čtyř logických celků:

1. Kapitola Kartografická generalizace poskytuje souhrnný teoreticko-analytický základ pro pochopení oblasti automatizované generalizace. Cílem kapitoly je umožnit formou krátkého textu seznámit uživatele metodiky ve zjednodušené formě s tématem automatizované generalizace jako takové. To mohou využít jak přímo uživatelé ke snažšímu porozumění funkcionalitě generalizačního software, tak zejména noví členové budoucích vývojových týmů k orientaci.
2. Kapitola Shrnutí stávajícího stavu v oblasti automatizované generalizace navazuje na předchozí kapitolu, předpokládá teoretické znalosti minimálně na její úrovni a poskytuje souhrnnou informaci o stavu využití metod automatizované generalizace u národních mapovacích agentur relevantních v kontextu Základních map ČR.
3. Kapitola Procesní model řízení generalizace popisuje model řízení generalizace vyvinutý v rámci tohoto projektu. Model je stavěn obecně a modulárně tak, aby ho bylo možno v budoucnosti využít jak pro rozšiřování o další měřítka, zvýšení kvality provedené generalizace i vylepšení zdrojových dat.
4. V přílohách jsou připojeny výsledky některých analýz dokreslující složitost problému automatizované generalizace a usnadňující porozumění textu jednotlivých kapitol.

1 Teoretické základy automatizované generalizace

1.1 Kartografická generalizace

Kartografická generalizace představuje komplexní proces, při kterém dochází k redukci, resp. prostorové, tvarové či atributové transformaci obsahových prvků mapy. Odvozené mapové dílo musí respektovat kartografické zásady, zejména požadavky obsahové přehlednosti a grafické názornosti při zachování úplnosti či geometrické přesnosti. Tyto požadavky definované kritérii jsou často ve vzájemném rozporu, výsledný kartografický produkt je obvykle jejich kompromisem. Jednotlivá kritéria tak mohou ale nemusí být splněna zcela, avšak pouze částečně; míra a forma závisí na řadě faktorů.

Klasická kartografie tento problém řeší zpravidla empiricky, a to s využitím formalizovaných postupů reprezentujících generalizační operátory. Tento přístup, založený na kognitivní analýze generalizované mapy, je však poměrně subjektivní, generalizační operátory jsou často kodifikovány volně či pouze v obecné rovině, poskytují značnou míru volnosti při řešení generalizačních situací a jejich následném hodnocení. Takto popsaný přístup k řešení problému, při kterém jsou postupně, systematicky a opakovaně aplikovány jednotlivé generalizační operátory s cílem vytvoření nového kartografického díla, představuje určitou formu optimalizace. Splnění/nesplnění optimalizačního kritéria je ověřováno zpravidla na základě vizuálního hodnocení vztahu generalizovaného prvku k ostatním prvkům.

Cílem tohoto materiálu bude navrhnout algoritmy pro automatizovanou či semi-automatizovanou kartografickou generalizaci topografických map velkých měřítek simulující chování kartografa, porovnat jednotlivé strategie, zmínit jejich klady a zápory a navrhnout funkční řešení tohoto problému.

Vzhledem k faktu, že při kartografické generalizaci topografických map středních měřítek je převládající operací odsun prvku, bude této operaci věnována v textu největší pozornost. Detailně bude popsána problematika detekce grafických konfliktů mezi jednotlivými prvky v závislosti na jejich tvaru a metody řešení těchto konfliktů. Se zmenšujícím se měřítkem mapy se začíná výrazněji uplatňovat vliv dalších generalizačních operátorů, zejména tvarových korekcí prvků, agregaci či typifikaci.

1.1.1 Geometrické vs. kartografické řešení

Chceme-li automatizovat proces kartografické generalizace, výše uvedené kroky mohou být odrazovým můstkem při hledání vhodných postupů a přístupů k řešení tohoto problému. Narážíme však na řadu omezení daných charakterem řešené úlohy. Kartografie je chápána jako věda i umění, postupy založené na vizuálním hodnocení prvků jsou subjektivního rázu, obtížně kvantifikovatelné či exaktně verifikovatelné. Geometrické řešení problému není ve většině případů postačující, představuje spíše první iteraci. Vzhledem k absenci hodnotících kritérií exaktního charakteru či přesné definici omezujících podmínek má úloha kartografické generalizace více řešení (obecně nekonečně mnoho). Neexistuje tedy jediné správné řešení problému, ale řada optimálních řešení, která jsou vzájemně obtížně porovnatelná.

1.1.2 Strategie řešení konfliktů při generalizaci

Kartografická generalizace reprezentuje proces zjednodušení a zevšeobecnění obsahu mapy a vzájemných vztahů mezi prvky. Vzhledem k měřítku, obsahu a typu mapového díla představovalo řešení grafických konfliktů nejčastější generalizační operaci; v této kapitole jí bude věnována největší pozornost.

Cílem kartografické generalizace je uchování názornosti, čitelnosti a přehlednosti kartografického díla. *Grafický konflikt* prvků mapy je zpravidla důsledkem absence generalizace či nerespektování kartografických zásad při tvorbě mapy. Nejčastěji se projevuje nedodržením minimální vzdálenosti mezi prvky, při které dojde k optickému slití obou prvků, nevhodným umístěním jednoho prvku dovnitř jiného, překrytím části

prvku jiným prvkem či překročením míry grafické zaplněnosti mapy. Dochází tak ke snížení přehlednosti a čitelnosti mapového díla.

S ohledem na velikost, rozsah a typ zpracovávaných dat existuje několik základních strategií řešících konflikty jednotlivých prvků.

1. Hrubá síla

Tato strategie je založena na vyhledávání a řešení vzájemných grafických konfliktů mezi všemi dvojicemi prvků v mapě. Obsah mapy je zpracováván jako celek, hledání potenciálních konfliktů je prováděno stylem každý s každým (tj. všechny existující kombinace). Tento přístup nepoužívá k odhalení problematických míst ani k řešení konfliktů žádnou vnitřní inteligenci. Jednotlivé operátory jsou na prvky aplikovány opakovaně tak dlouho, dokud řešení nesplňuje stanovenou výchozí podmínku. Nevýhodou přístupu je fakt, že množina není dělena na zpracované/nezpracované prvky, což prodlužuje dobu běhu algoritmu. Prvky korektně generalizované v předchozích fázích mohou být v budoucnu dále modifikovány. V průběhu iteračního procesu se poloha, tvar i vztahy mezi prvky mohou významně měnit. Tato strategie je náročná vzhledem k časové složitosti, pro rozsáhlé množiny není v praxi použitelná. Významnou roli hraje zejména grafická zaplněnost mapy, která přímo ovlivňuje dobu běhu algoritmu.

2. Inkrementální přístup

Inkrementální strategie je založena na principu, že přidání jednoho prvku do mapy neovlivní mapu jako celek, avšak pouze lokální okolí kolem přidávaného prvku. Do generalizované mapy jsou prvky přidávány po jednom, pro každý nový prvek a jeho lokální okolí jsou aplikována dostupná generalizační pravidla. V obecném případě vede přidání malého konfliktního prvku k menšímu počtu následných generalizačních operací než přidání velkého prvku (zpravidla vyvolá více následných korekčních kroků spojených s umístěním takového prvku do mapy). Přístup vychází z předpokladu, že malý prvek s nevhodnou polohou či tvarem při zaplněné mapě negativně ovlivní méně již zpracovaných prvků než nově přidávaný nevhodný prvek většího rozsahu. Pozitiva tohoto přístupu rostou přímo úměrně s grafickou zaplněností mapy, kdy se následné generalizační operace stávají stále rozsáhlejšími a složitějšími. Existují dva základní principy, které mohou být aplikovány na přidávaný prvek:

a. *Inkrementální konstrukce*

Vychází z předpokladu, že dříve vytvořené řešení je bezchybné, tj. již dříve umístěné prvky nejsou v žádném vzájemném konfliktu. Pro přidávaný prvek jsou nejprve nalezeny konflikty s již dříve umístěnými prvky. Vlastnosti přidávaného prvku jsou modifikovány tak dlouho, dokud nejsou všechny grafické konflikty odstraněny. Nevýhodou přístupu je fakt, že při značné zaplněnosti mapy nemusí být možné nový prvek korektně generalizovat. Může dojít k zpřetrhání geometrických, obsahových či atributových vazeb mezi přidávaným prvkem a jeho okolím. Výhodou přístupu je relativně nízký počet následných generalizačních kroků, které přidáním prvku vzniknou.

b. *Inkrementální vkládání*

Řešení z předchozích iterací není chápáno jako pevné, přidáním nového prvku může být aktualizováno. Nutným předpokladem funkcionality je fakt, že následné generalizační operace vyvolané přidáním prvku postihnou pouze malé procento mapy. Tato metoda nerozděluje mapu na již zpracovanou a nezpracovanou, každá již zpracovaná část generalizované mapy může být v následných krocích revidována a upravena. Z kartografického hlediska bude poskytovat věrnější výsledky; nedochází k izolované úpravě pouze jednoho prvku, ale i lokálního okolí přidávaného prvku, na které existují kartografické, logické, obsahové či

geometrické vazby. Tento přístup lze chápat jako analogii ruční generalizaci, při které kartograf postupně zpracovává jednotlivé prvky a jejich okolí modifikuje tak dlouho, dokud nejsou veškeré grafické konflikty vyřešeny.

Metoda vede k rekurzivnímu řešení problému. Nejprve jsou nalezeny prvky v grafickém konfliktu s přidávaným prvkem a jsou na ně aplikovány generalizační operátory. V důsledku těchto generalizačních kroků může dojít u modifikovaných prvků k novým grafickým konfliktům. Pro nově modifikované prvky jsou následně opět vyhledány potenciálně konfliktní prvky, na které jsou aplikovány generalizační operátory. Konfliktní prvky mohou být ukládány do zásobníku, proces generalizace je ukončen v případě, kdy žádný upravitelný prvek neexistuje. Podobný princip bývá používán u tzv. propagačních metod.

Výhodou inkrementální metody je absence požadavku předzpracování vstupních dat.

3. Zametací přímka

Tento přístup je založen na rozdělení vstupní množiny na dvě části. První část představuje množinu, ze které již byla vytvořena generalizovaná mapa; je chápána jako neměnná. Druhá část vstupní množiny čeká na své zpracování. Obě části lze chápat jako poloroviny oddělené zametací přímkou (Sweep Line). Metoda vyžaduje předzpracování vstupních dat; všechny lomové a vztažné body prvků jsou seříděny podle x nebo y souřadnice. V pomocné datové struktuře je navíc udržován seznam otevřených (tj. zpracovávaných prvků). Zametací přímka se pohybuje ve směru souřadnice x/y a to v závislosti na způsobu předzpracování dat. Jakmile narazí na nějaký bod, je generována nová událost. Při ní je testováno, zda je bod součástí již nějakého zpracovávaného prvku. Pokud není, je nově přidán do seznamu zpracovávaných prvků a na přidávaný prvek jsou aplikována generalizační kritéria. Pokud prvek v seznamu již je, není provedena žádná operace. Jedná-li se o poslední vrchol prvku (všechny jeho předešlé vrcholy již byly navštíveny), je prvek ze seznamu zpracovávaných prvků odstraněn.

Metoda má podobné vlastnosti jako inkrementální vkládání, generalizační operace řeší pouze mezi zpracovávanými prvky (tj. prvky, které zametací přímka protíná), což je blízké kartografickému chápání reality. Vzhledem k faktu, že zametací přímka prochází celou mapou (horizontální/vertikální), může být v jednom kroku zpracovávána poměrně velká podmnožina mapy. V takovém případě je nalezení vhodného řešení poměrně časově náročné. V některých situacích nemusí být řešení generalizačních situací v množinách tvarově blízkých vodorovnému či svislému tvaru přirozené vzhledem ke kartografickému vnímání reality, okolí je nesymetrické vzhledem ke zpracovávanému prvkem.

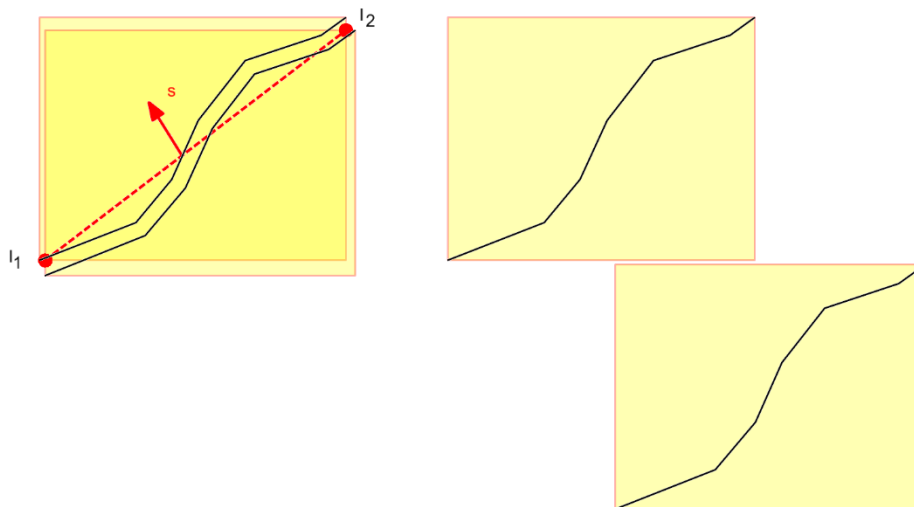
Zametací přímka se plynule pohybuje ve směru horizontálním/vertikálním nad generalizovanou mapou.

4. Heuristika

Heuristika je založena na snaze minimalizovat v každém kroku počet grafických konfliktů. Jedna z možných heuristik je založena na jednoduchém přístupu, při kterém je regulováno pořadí prvků, v kterém jsou přidávány do mapy (prvky nelze do mapy přidávat randomizovaně). Nejprve jsou do mapy přidávány takové prvky, které mohou díky své velikosti či tvaru způsobit větší množství grafických konfliktů. V prvních krocích jsou vkládány rozsáhlé a tvarově komplexní prvky, jako poslední malé či tvarově jednoduché prvky. Myšlenka je založena na faktu, že konflikty složitých prvků je jednodušší řešit nad málo zaplněnou mapou než nad mapou zaplněnou. Tato technika je často kombinována s inkrementálním vkládáním, vyžaduje předzpracování prvků.

1.1.3 Optimalizovaná kritéria

Na generalizovanou mapu je kladeno mnoho různých požadavků, uvedeme pouze nejpodstatnější. Z vizuálního hlediska by si mapa měla uchovat zejména přehlednost, čitelnost, názornost či míru grafické zaplněnosti. Z kartografického pohledu by měla být redukce obsahu provedena tak, aby byly uchovány logické, tematické i prostorové vztahy a vazby mezi jednotlivými prvky. Tyto požadavky, které jsou mnohdy protichůdného charakteru, je proto obtížné harmonizovat do formy jednotného univerzálního kritéria, které by reflektovalo takto rozmanité nároky kladené na mapu. Některá kritéria nejsou exaktně měřitelná, významnou roli při jejich používání hraje zejména vnímání kartografického díla jako celku. Z výše uvedených důvodů je proto velmi obtížné provést verifikaci výsledků. Kartografické úlohy z tohoto pohledu patří mezi algoritmicky obtížné úlohy, u kterých zpravidla není postačující geometrické řešení problému; prostorové vztahy jsou pouze podmnožinou vztahů mezi prvky, které musí být uchovány.



Obrázek č.1.1 Aproximace prvků min-max boxem (vlevo), vpravo posun prvků tak aby nedošlo ke grafickému konfliktu, směr posunu označen s .

Z praktického důvodu není možné provést návrh komplexního kritéria tak, aby zohledňovalo všechny požadované aspekty kladené na generalizované dílo; je nutné zvolit rozumný kompromis mezi kartografickou věrností a výpočetní náročností. Kritérium založené na geometrických a prostorových vztazích mezi jednotlivými prvky zůstává základním nástrojem hodnocení, řešení založené na uchování prostorových vztahů mezi prvky může být chápáno jako první iterace. Kritéria negeometrického charakteru jsou obtížněji vyjádřitelná v analytické formě, jejich použití vede k návrhu složitých modelů z oblasti umělé inteligence. Tyto postupy jsou zpravidla výpočetně náročnější, mají nedeterministický charakter; nejsou tedy vhodné pro mapová díla velkého rozsahu.

1.1.3.1 Grafický konflikt

Základní myšlenkou, která kombinuje grafický a geometrický přístup, představuje minimalizace množství grafických konfliktů na generalizované mapě. Vzhledem k typové a tvarové variabilitě prvků chápeme jako grafický konflikt takovou situaci, kdy se dva prvky dostanou k sobě blíže, než je grafické rozlišení mapy (zpravidla uváděna jako 0.1 mm), protínají se, popř. prvek leží uvnitř jiného prvku.

1.1.3.1.1 Zjednodušená reprezentace prvku

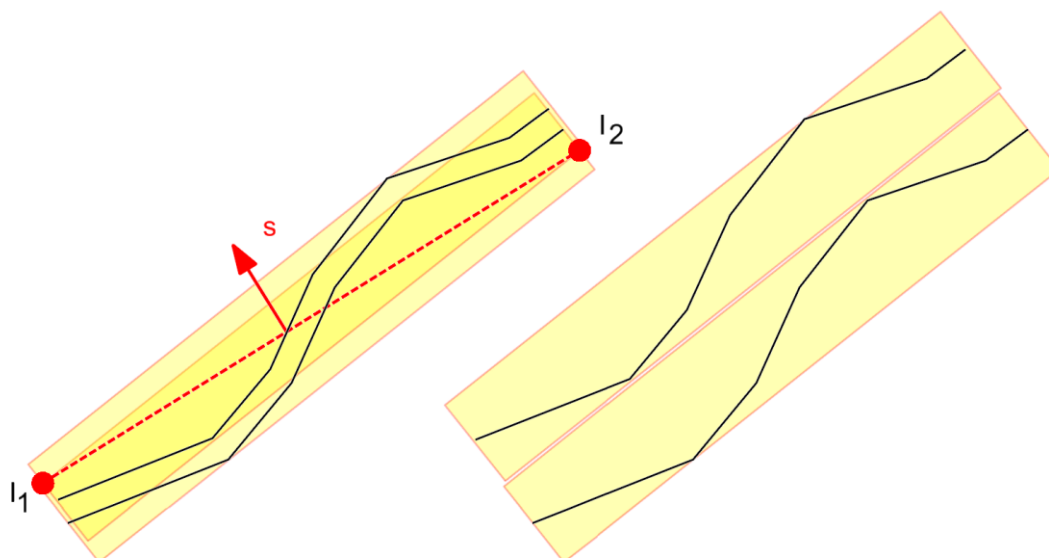
Z výpočetního hlediska není vhodné posuzovat grafické konflikty přímo mezi jednotlivými prvky, ale jejich zjednodušenou reprezentací. Pro tento účel lze použít následující geometrické struktury:

- a. Min-max box

Výhodou této reprezentace je snadná konstrukce ($O(N)$), je však vhodná pouze pro prvky s podobnou šířkou/ délkou (poměr šířky a délky min-maxboxu se blíží jedné) s převládajícím směrem daným osami x,y (prvek není natočen). V opačném případě vede ke vzniku falešných kolizí, kdy se protínají pouze zjednodušené reprezentace těchto prvků a nikoliv prvky samotné. Zasahuje příliš „daleko“ od prvku, protíná prvky, které nemusí být v kontaktu s analyzovaným. Jako nevhodné se jeví zejména diagonální prvky liniového charakteru (vodstvo, silniční síť) jdoucí napříč mapou, u kterých je min-max box nevhodnou tvarovou aproximací. Reprezentace min-max boxem umožňuje snadnou detekci kolizí, vede k analýze průsečíku dvou obdélníků.

b. Minimum Bounding Rectangle

Minimum Bounding rectangle je lepší aproximací prvku, zohledňuje jeho natočení. Jeho celková plocha není větší než plocha min-max boxu, k prvku se přimyká těsněji. Není vhodný pro výrazně nekonvexní prvky (nedostatečná aproximace tvaru), lepších výsledků dosahuje pro prvky konvexní. Konstrukci lze realizovat v čase $O(M \log(N))$, nevýhodou je složitější detekce kolizí. Pro obě výše uvedené reprezentace je detekce kolize úlohou s konstantní časovou složitostí.



Obrázek č.1.2 Aproximace prvků Minimum Bounding Rectanglem (vlevo), vpravo posun prvků tak aby nedošlo ke grafickému konfliktu, směr posunu označen s .

c. Convex Hull

Konvexní obálka poskytuje lepší tvarovou aproximaci prvku, u prvků konvexních nedojde ke změně reprezentace. Pro výrazně nekonvexní prvky je však její tvar příliš generalizovaný. Detekce kolizí konvexních obálek v takových případech vede k falešným kolizím; jejich procento je však výrazně nižší než v předcházejících případech. Konstrukci lze realizovat v čase $O(M \log(N))$, detekce kolizí konvexních mnohoúhelníků má kvadratickou složitost $O(NM)$, je funkcí počtu vrcholů N , M obou útvarů. Tato reprezentace přináší nižší počet falešných kolizí, vlastní detekce kolize je však výpočetně náročnější.

d. Rasterizace

Rasterizace prvků mapy umožňuje rychlou detekci vzájemných grafických konfliktů, je však výpočetně dosti náročná, přesto poměrně často používaná.

1.1.3.1.2 Zjednodušená reprezentace bufferu

Globální aproximace prvku jeho zjednodušenou reprezentací pro účely detekce kolizí naráží na své limity v zaplněných oblastech mapy, a to hlavně u rozsáhlých nekonvexních útvarů, které se v praxi vyskytují často. V důsledku falešných kolizí dochází k následné generalizaci prvků, u kterých tato operace není nezbytná. Jako perspektivnější varianta pro přesnější detekci kolizí se jeví aproximace ekvidistanty prvku, tj. jeho bufferu.

e. Aproximace min-max boxem

Nejjednodušší varianta aproximace je založena na konstrukci min-max boxu nad každým segmentem, nevýhodou je malá těsnost vzhledem k aproximovanému prvku (vede ke vzniku falešných kolizí).

f. Natočený obdélník

Buffer je aproximován obdélníkem, jehož jedna strana je rovnoběžná s prvkem. Délka druhé strany odpovídá dvojnásobku šířky bufferu.

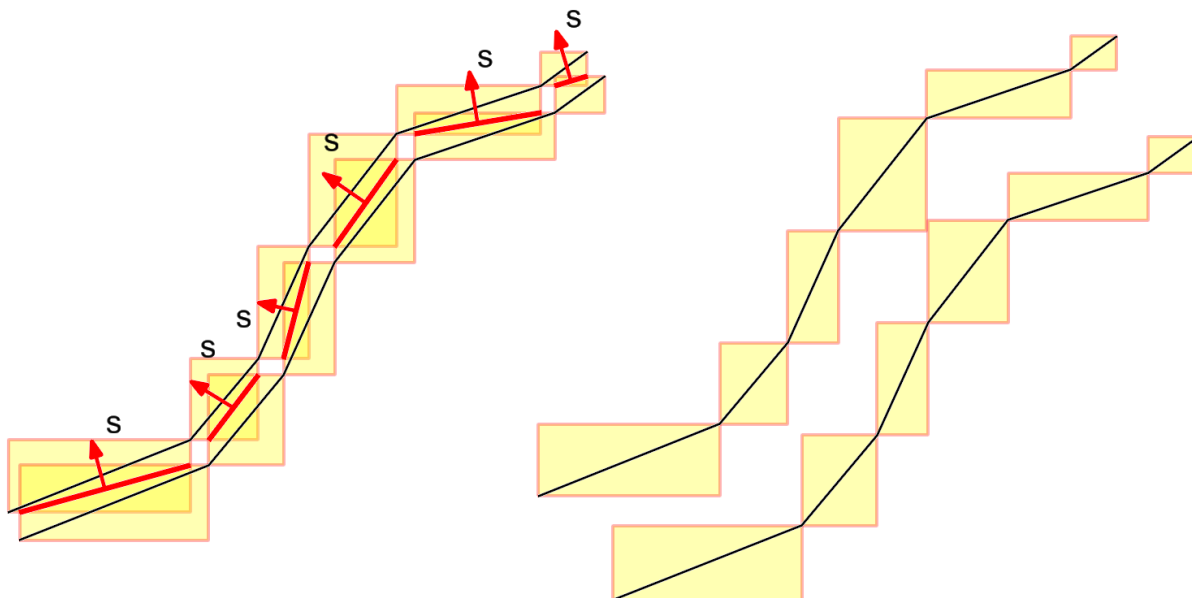
g. Kosodélník

Další varianta aproximace je založena kosodélníku. Jedna jeho strana je rovnoběžná se směrem prvku. Druhá strana, s délkou odpovídající dvojnásobku šířky bufferu, je rovnoběžná s osou x či y .

Výhodou přístupů 2), 3) je velmi rychlá konstrukce (lze realizovat on-the-fly). Tato aproximace vede ke vzniku relativně malého počtu falešných kolizí. Tvarově se tato aproximace více či méně blíží bufferu, doba konstrukce i řešení kolizí jsou však výrazně kratší.

1.1.3.2 Řešení grafického konfliktu

Základním myšlenkou představuje posun dvojice prvků, které jsou v grafickém konfliktu, tak, aby jejich průnik byl představován prázdnou množinou. Směr posunu může být náhodný, lze ho zadat vektorem, velikost posunu však není zpravidla známa.



Obrázek Č.1.3 Aproximace prvků obdélníkovým bufferem (vlevo), vpravo posun prvků tak aby nedošlo ke grafickému konfliktu, směry posunu označeny s .

Úloha je analyticky řešitelná pouze pro konvexní polygony, a to s využitím Axis Separation Theorem (SAT). S jeho využitím lze určit posuny s_x , s_y pro prvky tak, aby oba prvky nebyly v grafickém konfliktu.

Pro nekonvexní prvky či nekonvexní reprezentace nemá úloha analytické řešení, je nutné použít nějakou formu heuristiky, např. řízené procházení stavovým prostorem. Při něm je metoda pokus-omyl hledáno akceptovatelné řešení. Nevýhodou tohoto přístupu je nemožnost odhadnout dobu, za kterou bude nějaké řešení nalezeno. U graficky zaplněné mapy se může jednat o mnoho zbytečných pokusů. Umožníme-li hledání pouze v určitém směru, úloha nemusí mít řešení, popř. bude posunutý prvek ležet příliš daleko od původních prvků.

Pokud není možné posunout celý prvek, provede se zpravidla posun pouze některé jeho části.

1.1.3.3 Minimalizace grafických konfliktů

První minimalizované kritérium vychází z požadavků přehlednosti, čitelnosti a názornosti, které jsou kladeny na generalizované dílo. Cílem je minimalizace grafických konfliktů jednotlivých prvků na mapě. Z geometrického hlediska představuje nalezení posunů s_x , s_y pro každý prvek tak, aby celková plocha průniků všech vzájemných kombinací prvků byla nulová.

Jak již bylo uvedeno výše, mnohé prvky mají příliš složitý tvar. Pro detekci konfliktů lze použít zjednodušenou reprezentaci prvků, nejlepších výsledků dosahuje aproximace bufferem tvořeným natočenými obdélníky. Hodnota odsunu je nejmenší ze všech přibližných aproximací prvků, výpočty konfliktů jsou však výrazně složitější než u obdélníku v základní poloze.

Další úprava metody může spočívat v tom, že při vzniku konfliktu použijeme analytické řešení pro ověření, zda ke kolizi skutečně došlo (tj. nejedná -li se o falešný konflikt); neprovádíme tedy řešení konfliktu grafických prvků pouze na základě kolize jejich zjednodušených reprezentací. Nejprve jsou hledány potenciální kolizní prvky, následně je provedeno ověření, zda ke grafickému konfliktu skutečně dochází.

Tato úloha má vždy řešení, avšak nevhodné použití zjednodušených reprezentací může vést k velkým posunům tvarově členitých či rozsáhlých prvků, které má zpravidla izolovaný charakter. Vznikají tak místa, u kterých sekundárními posunutými dochází ke ztrátě prostorových, topologických, atributových či významových vztahů.

Problém lze chápat také jako klasickou optimalizační úlohu, jejíž obtížnost závisí na *strategii* zvolené pro řešení konfliktů.

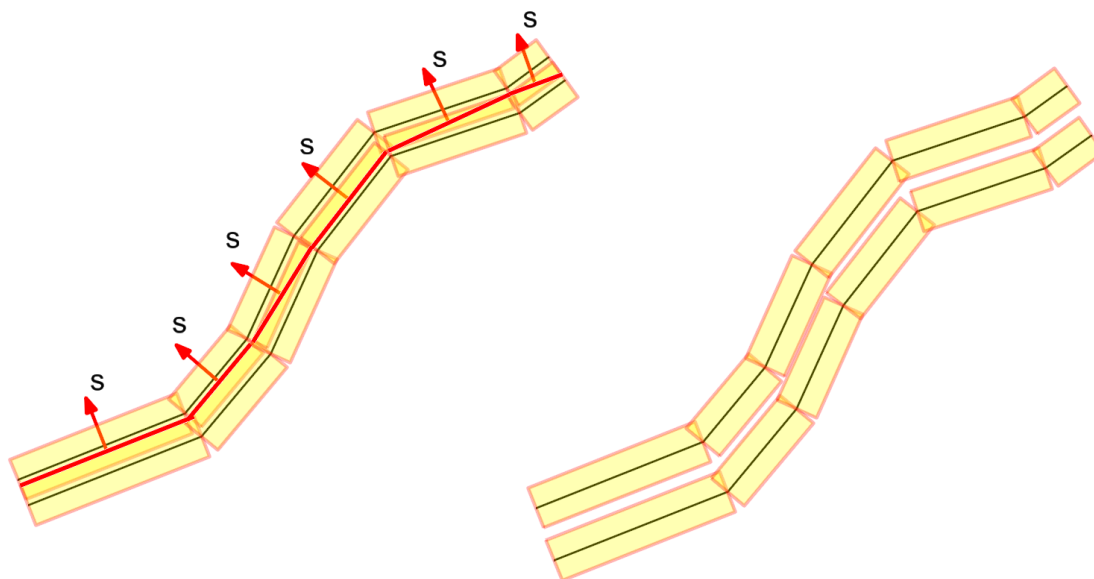
Metoda hrubé síly.

U metody hrubé síly, kdy je zpracovávána mapa jako celek, tento postup vede k optimalizaci s $2n$ neznámými, kde n představuje počet prvků mapy. Úloha má nekonečně mnoho řešení. Pro běžný mapový list s počtem prvků v řádek tisíců se jedná o optimalizační úlohu středního rozsahu s výpočetními dobami v řádech minut či hodin dle zvolené optimalizační strategie. Vzhledem k celkovému počtu objektů ZABAGED (cca 17 mil.) je tato technika nepoužitelná.

Úlohu lze pro některé reprezentace (obdélníky) řešit prostřednictvím MNČ. V ostatních případech nelze průsečnici reprezentací vyjádřit v analytickém tvaru vhodném pro optimalizaci. S využitím Separation Axis Theorem (SAT) lze analyticky pro dvojici konvexních polygonů určit velikost posunů s_x , s_y tak, aby průnikem obou prvků byla prázdná množina.

U stochastických metod, kdy je řešení hledáno náhodným, resp. pseudonáhodným, procházením stavového prostoru, dochází ke generování řešení, která z kartografického pohledu nemusí být korektní. Prvky mohou být posouvány o značně velké vzdálenosti (dokonce i mimo analyzované území), podmínka minimalizace grafických konfliktů je tedy splněna. Tyto posuny však vedou ke ztrátě vztahů mezi prvky pro některé

optimalizační strategie k úplné destrukci mapového díla. Z tohoto důvodu je nutné použít doplňkové kritérium, které minimalizuje změnu vzdálenosti přesouvaných prvků.



Obrázek č.1.4 Aproximace prvků obdélníkovým bufferem s natočením (vlevo), vpravo posun prvků tak aby nedošlo ke grafickému konfliktu, směry posunu označeny s .

Inkrementální konstrukce.

Při použití inkrementální konstrukce je tato úloha jednoduchá, pro přidávaný prvek jsou určovány dva posuny s_x , s_y optimalizační úloha je snadno řešitelná.

Inkrementální vkládání.

Při inkrementálním vkládání není a-priori známo, jaká část řešení bude po přidání dalšího prvku updatována. Základní myšlenka inkrementálního vkládání vychází z předpokladu, že dotčených prvků bude málo, např. 1% obsahu mapového listu. V iniciální fázi jsou nalezeny prvky, které jsou přidáním nového prvku dotčeny. Nad těmito prvky proběhne optimalizace grafických konfliktů metodou hrubé síly s využitím MNČ či stochastických metod. Tato operace může vést u posunutých prvků ke vzniku sekundárních grafických konfliktů, úloha proto vede na rekurzivní implementaci.

Zametací přímka.

V dynamické datové struktuře je udržován seznam prvků, které jsou zametací přímkou protnuty. Mezi těmito prvky následně probíhá vzájemná optimalizace grafických konfliktů.

1.1.3.4 Minimalizace změny polohy prvku

Jak již bylo výše zmíněno, úloha minimalizace grafických konfliktů má nekonečně mnoho řešení. Díky příliš dlouhým posunům optimalizovaných prvků, mají -li navíc stochastický charakter, taková řešení nemusejí být kartograficky korektní (ztráta souvislostí mezi prvky). Jednou z možností se jeví omezení velikosti či stanovení směru odsunů, pak by se úloha transformovala na problém řešitelný podmíněnou optimalizací. Určení maximální vzdálenosti odsunu či nalezení vhodných směrů odsunu jsou pro obecnou mapu obtížně stanovitelné, důležitou roli hraje její měřítko, typ, či lokální grafická zaplněnost v okolí generalizovaného prvku. Tentokrok by si vyžádal komplexnější výzkum zahrnující rozbor kartografických pravidel a generalizačních operátorů.

V praxi se jako vhodnější jeví kombinace kritéria minimálního grafického konfliktu s dalším geometrickým kritériem. Zohlednění velikosti posunů jednotlivých generalizovaných prvků představuje vhodnější variantu řešení. V takovém případě bude

preferováno řešení eliminující grafické konflikty s minimální sumou posunů všech generalizovaných prvků. Minimalizovaná funkce tedy zohledňuje obě kritéria, tj. kritérium minimálního překrytu a minimální délky posunutého prvku; obě kritéria jsou mapována na interval (0,1). Optimalizovaná funkce byla navržena jako součet obou mapovaných kritérií. V praxi však lze použít i jiné varianty, např. $f(x, y) = x/(1 - y)$, kde x, y představují hodnoty obou kritérií. Význam obou kritérií je také možné vážit zavedením vhodného váhového koeficientu a vyvážit tak míru částečného splnění/nesplnění obou kritérií. Stanovení hodnot vah lze provést pouze empiricky.

U navrženého postupu kombinující obě kritéria stále přetrvávají výše zmíněné nevýhody. Generalizovaný prvek může být posunut tak, že i přes malou velikost posunu dojde ke zpřetrhání vazeb se svými sousedy. Tato metoda nezohledňuje polohu ani vztah generalizovaného prvku k prvkům ostatním, problém řeší izolovaně. Dosahované kartografické výsledky mohou působit nepřirozeným či umělým dojmem.

1.1.3.5 Minimalizace změny polohy prvku vzhledem k sousedům

Zahrnutí vztahu generalizovaného prvku ke svým sousedům představuje důležitou vlastnost navrhovaného řešení. První krok zahrnuje nalezení vhodných sousedů generalizovaného prvku. V praxi budou za sousedy považovány:

- prvky, které jsou generalizovanému prvku geometricky blízké. Jako vhodné se jeví výběr všech přirozených sousedů. Pro konstrukci přirozených sousedů lze použít Voronoi teselaci, jednotlivé generátory budou tvořeny 0D-2D objekty. Za přirozené sousedy budou označeny takové generátory (tj. prvky), jejichž Voronoi buňky sdílejí alespoň jednu hranu s Voronoi buňkou příslušející generalizovanému prvku.
- prvky, které v kartografickém modelu sdílejí nějaký typ kartografické, logické či jiné vazby. Tyto prvky nemusejí sousedit s generalizovaným prvkem či se nacházet v jeho blízkosti, spojují je s ním vztahy, které by v rámci generalizačního procesu neměly zaniknout (např. řeka-jez, silnice-most).

Navrhovaná funkce bude minimalizovat sumu vzdáleností generalizovaného prvku ke všem prvkům, které jsou označeny jako sousední. Vzdálenost dvou prvků může být parametrizována vhodnou metrikou, např. Hausdorffovou. Komplexní kritérium je kombinací minimalizace grafických konfliktů a minimalizace změny polohy prvků vůči svým sousedům. Minimalizovaná funkce je volena stejně jako v předchozím případě, její chování vzhledem k jednotlivým kritériím lze modelovat zavedením vah.

Tato varianta přináší výsledky, které jsou bližší ruční kartografické generalizaci, postihuje vzájemné vztahy mezi generalizovanými prvky. Konflikty nejsou řešeny izolovaně, ale hromadně, tj. pro generalizovaný prvek a jeho nejbližší okolí. Avšak ani geometrický, atributový či kartografický vztah nemusejí být dostatečné pro definici okolí generalizovaného prvku. U rozsáhlých liniových nebo plošných prvků či u míst s lokálně velkou zaplněností může výše uvedená definice vést k příliš rozsáhlé definici okolí s velkým množstvím sousedů, algoritmus se stává méně efektivním vzhledem k výsledkům i implementaci.

1.1.4 Omezující podmínky

Výše uvedené návrhy vedou k řešení problému zohledňujícím geometrický aspekt problému. Důvodem je především fakt, že při řešení grafických konfliktů je preferován geometrický přístup. Geometrické řešení problému kartografické generalizace však nemusí být (a zpravidla ani není) postačující, protože není schopné podchytit některé typy kartografických či jiných vazeb mezi prvky. Z tohoto důvodu musí být přidány omezující podmínky, tzv. *constraints*, pomocí kterých lze specializované vztahy popsat, či vymezit okolí, směr, interval, ve kterých mohou být kartografické operátory použity.

Omezující podmínky jsou poměrně snadno formalizovatelné, lze je přepsat do jednoduchých matematických vztahů. Jako výrazně obtížnější se jeví nalezení prvků

mapy, na které jsou aplikovatelné, a způsob jejich použití, pořadí aplikace, či maximální hodnoty těchto kritérií. Je patrné, že výsledek generalizační operace závisí na pořadí, v jakém jsou tyto podmínky aplikovány (neplatí zde pravidlo komutativity). Tento proces je součástí strategie řízení generalizace, která bude popsána v kapitole 2.

1.1.4.1 Vzdálenostní omezení (Distance Constraint)

Tato omezení popisují splnění kartografických pravidel souvisejících s regulací limitní vzdálenosti mezi prvky. Týkají se zejména odsunů prvků, kdy je nutné dodržet minimální vzdálenost od jiného prvku, aby nedošlo k jejich grafickému slití (zpravidla se používá hodnota 0.1mm). Aby nedošlo ke ztrátě vztahu mezi dvěma či více prvky, je vhodné stanovit maximální hranici odsunu.

1.1.4.2 Polohová omezení (Positional Constraint)

Další významnou kartografickou vazbu představuje polohové omezení dvou či více prvků. Do této skupiny můžeme zahrnout např. vzájemnou orientaci prvků vzhledem k jinému prvku (vlevo/vpravo), orientovanou vzdálenost, či vzdálenost v určitém směru. Vzdálenostní omezení definované v předchozím bodu nemusí být v takovém případě postačující, pro L2 metriku je množina bodů o stejné vzdálenosti představována kružnicí. Prvky si po aplikaci generalizačního operátoru tedy mohou uchovat vzájemnou vzdálenost, nikoliv však polohu, orientaci, či orientovanou vzdálenost.

1.1.4.3 Úhlová omezení (Angular Constraint)

Úhlová omezení patří do skupiny často používaných kritérií. Mohou být aplikována na prvek či skupiny prvků samostatně nebo v kombinaci s předchozími dvěma skupinami. Jejich použití souvisí se zachováním tvarových charakteristik či vzájemné orientace prvku.

- *Zachování tvaru*

U některých typů prvků by v průběhu generalizačního procesu mělo docházet pouze k malým změnám tvaru. Tato podmínka vede k požadavku zachování úhlu, zpravidla mezi sousedícími segmenty. Typickým příkladem představuje zástavba, u většiny budov (s výjimkou moderní architektury), vzniká potřeba generalizace tvaru se zachováním pravoúhlosti mezi sousedními segmenty.

- *Zachování křivosti*

Tento požadavek se uplatňuje zpravidla u prvků liniového charakteru, zejména u komunikací či železnice. V takovém případě je preferováno uchování křivosti prvku před generalizací tvaru (Bend Simplify Algorithms). Při změně směru některého ze segmentů je nutné zachovat oblé přechody mezi původním směrem a směrem novým na úkor tvaru prvku. Taková preference není absolutní, u některých prvků je preferováno zachování tvaru (státní hranice).

- *Zachování orientace*

Vzájemná orientace dvou nebo více prvků je důležitou vlastností, kterou je nutné u prvků zachovat. Významnou roli hraje zejména kolmost či rovnoběžnost prvků, uplatňují se zejména u segmentů liniového charakteru. Často je využívána v souvislosti s generalizačními operátory slícování, ztotožnění, či paralelizace. Požadavek zachování orientace lze kombinovat s polohovými omezeními, zejména s maximální vzdáleností.

- *Zachování směru*

Směrem prvku chápeme jeho azimut, nikoliv jako orientaci vzhledem k sousednímu prvku. Směr je významnou charakteristikou indikující, v jakém azimutu je určitý předmět nebo jev znázorněn na mapě dominantní, směrové vektory jsou často používány v tematických mapách.

1.2 Strategie řízení generalizace

Kartografická generalizace je komplexní proces zjednodušení a zevšeobecnění mapového díla probíhající v různých fázích, které na sebe vzájemně navazují. Elementární krok je představován aplikací konkrétního generalizačního pravidla na všechny prvky mapového díla, na specifické skupiny prvků, či pouze izolovaně (tj. na jednotlivé prvky), v jeho rámci jsou řešeny grafické, obsahové, kartografické interakce či konflikty prvků. Pořadí, v jakém jsou jednotlivá pravidla aplikována, způsob jejich použití a množina vstupních prvků jsou předmětem strategie řízení generalizace. Jak již bylo řečeno výše, při kartografické generalizaci zpravidla neplatí kritérium komutativity operátorů, výsledek operace je závislý na pořadí jednotlivých generalizačních operátorů.

Při ruční generalizaci jsou parametry generalizačních operací ovlivněny různými faktory, významnou roli hraje zejména způsob vnímání mapového díla kartografem, účel, typ, či měřítko mapy. Chápání obsahu kartografického díla v širším kontextu, zohlednění grafických logických interakcí a vazeb mezi jednotlivými obsahovými prvky mapy výrazně ovlivňují efektivitu generalizačního procesu. Tento komplexní přístup, založený jak na empirickém tak na analyticko syntetickém hodnocení kartografického díla, je doposud spojen pouze s procesem ruční generalizace, jeho automatizace je obtížně řešitelným problémem.

Automatické přístupy doposud neumožňují takto komplexní přístup k problematice generalizace, zpravidla využívají pouze podmnožinu vazeb, vztahů či konfliktů, jejichž efektivitu lokálně ověřují, tj. provádí její ohodnocení prostřednictvím předem daných kritérií. Cílem procesu generalizace je splnění nějakého zástupného kritéria, popř. kombinace těchto kritérií, tj. dosažení stavu odpovídajícího nejlepšímu možnému (či alespoň dostatečně dobrému) ohodnocení, při kterém nedochází ke vzájemnému konfliktu grafických prvků. Při automatizované generalizaci tedy dochází k transformaci cíle a způsobu jeho ohodnocení. Strategie řízení generalizace tedy vede k algoritmickým přístupům na bázi heuristiky, často používaným pro problémy, u kterých není znám exaktní postup řešení.

1.2.1 Optimalizace

Z obecného hlediska může být kartografická generalizace chápána jako určitá forma optimalizačního problému, prováděná nad množinou kartografických prvků, operace s nimi jsou popsány prostřednictvím kartografických operátorů. Jednotlivé elementární kroky, operace a postupy jsou svázány s konkrétními typy prvků, druhy map, či měřítka. Do procesu kartografické generalizace tak vstupují různá omezení, vyjádřitelná výše uvedenými podmínkami (Constraints). Z konceptuálního hlediska tedy ruční kartografická generalizace vede k podmíněné optimalizaci prováděné kartografem zpravidla na základě vizuálního hodnocení interakcí jednotlivých obsahových prvků mapy.

Konstrukce omezujících podmínek a hodnotících funkcí vede k multikriteriální optimalizaci, minimalizovaná funkce je zřejmě multimodální (disponuje více lokálními minimy). Nalezení jejího globálního minima představuje NP problém řešitelný nedeterministickými optimalizačními technikami, které jsou výpočetně náročné, a rychle řešitelné pouze pro problémy s malou dimenzí a malým rozsahem stavového prostoru. Nejčastěji používané optimalizační techniky pro problémy malého rozsahu představuje řízené procházení stavového prostoru, známé jako simplexová metoda. Pro problémy středního a velkého rozsahu lze použít genetické algoritmy, např. diferenciální evoluci. Vzhledem k výše uvedeným důvodům se nalezení suboptimální varianty jeví jako postačující řešení problému. V takovém případě lze použít nelineární MNČ.

Analytické vyjádření funkcí.

Klíčovým krokem pro využití optimalizačních metod představuje transformace kritérií do analytické formy. Představuje proces, při kterém je kartografický operátor transformován nejprve do geometrického tvaru, následně je hledáno jeho analytické vyjádření.

Požadujeme, aby tento funkční vztah, v závislosti na zvolené optimalizační technice, splňoval některé předpoklady.

- *Spojitosť funkce a jejich prvních derivací*

Tato podmínka je důležitá u konvexní optimalizace prováděné nelineární MNČ. Nevhodné chování funkce, místa nespojitostí druhého druhu mohou vést k selhání optimalizačního procesu.

- *Znalost gradientu funkce*

Je důsledkem předchozí podmínky, znalost gradientu funkce je nezbytným předpokladem při aplikaci nelineární MNČ. Při použití stochastických technik není znalost gradientu nutná.

- *Malá citlivost vůči šumu*

Funkce by měla být málo citlivá vůči šumu a úloha podmíněná.

Analytické vyjádření podmínek.

Druhý nezbytný krok představuje nalezení analytické formy vedlejších podmínek. Obvykle vede ke složitějším vztahům, některé podmínky nemusejí být tak snadno analyticky vyjádřitelné a použitelné v optimalizačních kritériích. Požadavky kladené na omezující funkce jsou stejné jako u optimalizované funkce.

Model mapy.

Předpokládejme, že generalizovaná mapa M , kde $M = \{E_i\}_{i=1}^n$, je tvořena n prvky E_i (elements), kde každý prvek může být reprezentován 0D-2D entitou.

1.2.1.1 Analytické vyjádření grafického konfliktu

Analytické vyjádření grafického konfliktu závisí na způsobu, jakým je prvek v mapě reprezentován. Pro některé typy jednoduchých reprezentací je analytické vyjádření míry grafického konfliktu k dispozici, lze ho tedy vyjádřit funkčním vztahem. V takovém případě lze pro nalezení optima použít některou z metod konvexní optimalizace. Pokud neexistuje analytické vyjádření grafického konfliktu, pro optimalizaci je nutné použít některou ze stochastických metod (optimalizovaná funkce nemusí být spojitá).

1.2.1.1.1 Grafický konflikt min-max box boxů

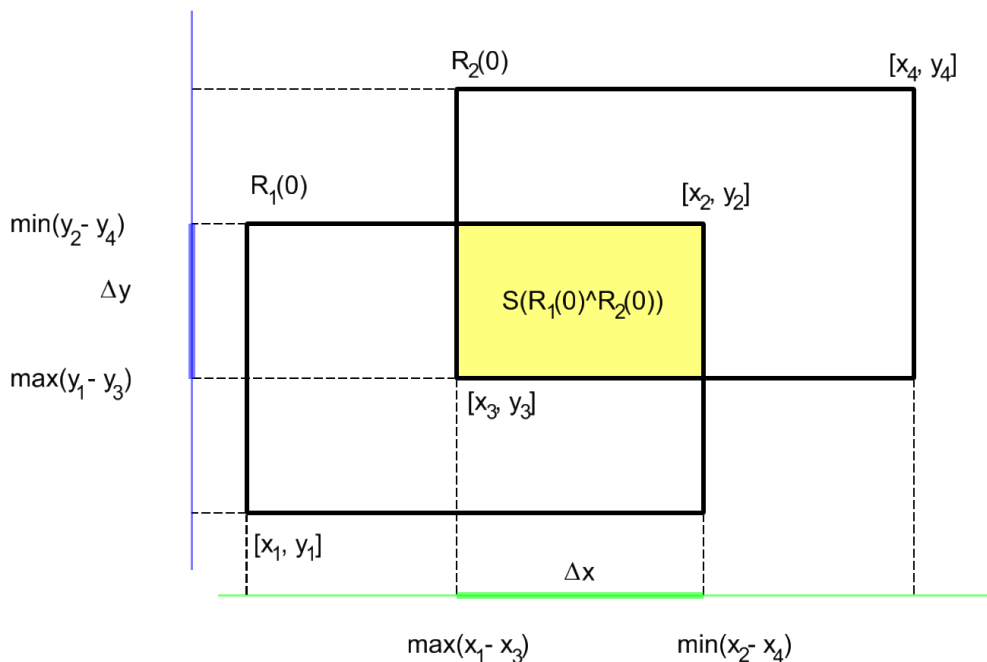
Nejjednodušší variantu problému představuje reprezentace prvku prostřednictvím min-max boxu. Nechť $R = \{R_i\}_{i=1}^n$ představuje množinu n obdélníků R_i , $R_i = \{x_i, y_i, w_i, h_i\}$, kde x_i, y_i jsou souřadnice levého dolního rohu a w_i, h_i jeho šířka a délka. Každý min-max box R_i je tvořen obdélníkem zkonstruovaným nad prvkem mapy E_i . Předpokládejme existenci dvou různých min-max boxů R_j, R_k , $j \neq k$, které jsou v grafickém konfliktu. Pak existuje jejich průnik

$$R_j \cap R_k \neq \emptyset.$$

pro jehož plochu platí

$$S(R_j \cap R_k) \geq 0.$$

Průnik dvou obdélníků může být tvořen prázdnou množinou, bodem, úsečkou, obdélníkem. U posledních varianty je plocha průniku nenulová, grafické konflikty však pravděpodobně mohou nastávat i v případech, kdy se prvky pouze vzájemně dotýkají. Tento problém je vhodné řešit min-max boxem s přesahem, v praxi postačuje cca 10% přesah. Označme $R_i(0)$ jako obdélník v základní poloze a $R_i(s)$ jako obdélník posunutý o hodnotu $s = (s_x, s_y) \in \mathbb{R}$, kde $\|s\|_2 = \sqrt{(s_x^2 + s_y^2)}$, viz obr. 2.1↓.



Obrázek č.2.1 Ukázka průsečnice dvou obdélníků R_1, R_2 v základní poloze.

Plochu průsečnice obdélníků $R_1 = \{x_1, y_1, x_2, y_2\}$, $R_2 = \{x_3, y_3, x_4, y_4\}$, kde

$$[x_2, y_2] = [x_1, y_1] + [w_1, h_1], \quad [x_4, y_4] = [x_3, y_3] + [w_2, h_2],$$

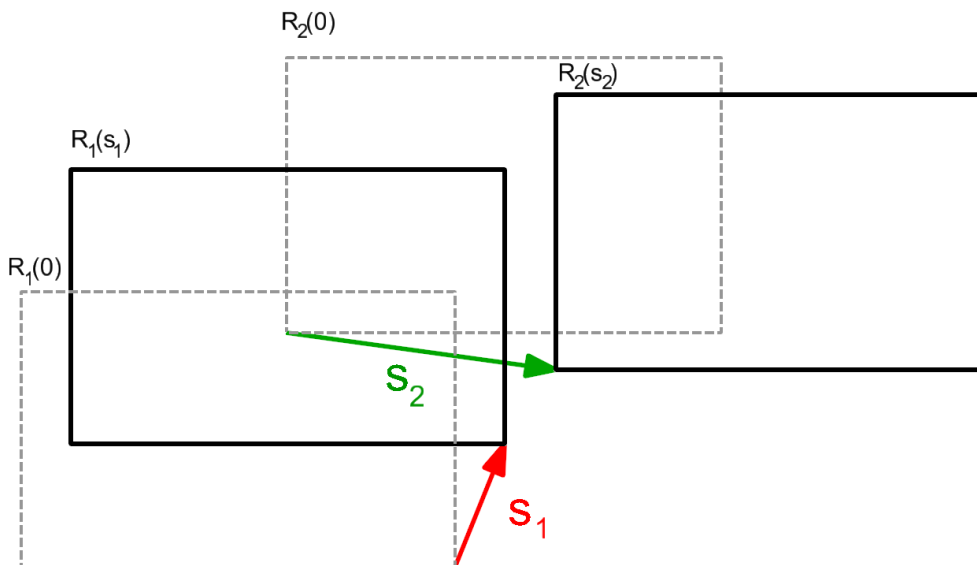
lze určit z projekce intervalů do souřadnicových os x, y

$$\Delta x = \min(x_2, x_4) - \max(x_1, x_3), \quad \Delta y = \min(y_2, y_4) - \max(y_1, y_3). \quad (2.1)$$

Pokud $\Delta x > 0 \wedge \Delta y > 0$, pak $S(R_1 \wedge R_2) > 0$, pak celkovou plochu průsečnice lze určit z

$$\begin{aligned} S(R_1 \wedge R_2) &= \Delta x \Delta y, \\ &= \max(0, \min(x_2, x_4) - \max(x_1, x_3)) \cdot \max(\min(y_2, y_4) - \max(y_1, y_3)), \\ &= \max(0, \min(x_1 + w_1, x_3 + w_2) - \max(x_1, x_3)) \cdot \max(\min(y_1 + h_1, y_3 + h_2) - \max(y_1, y_3)). \end{aligned}$$

Výhodou je snadný výpočet v čase $O(1)$, jsou použity pouze základní aritmetické operátory $+/-/*$.



Obrázek č.2.2 Určované posuny s_1, s_2 obdélníků R_1, R_2 .

Řešení pro 4 neznámé posuny.

V tomto případě posuneme oběma obdélníky tak, aby plocha průniku byla nulová. Označme minimalizovanou funkci ϕ , jejíž argumentem jsou hledané posuny s_1, s_2 jako

$$\phi(s_1, s_2) = S(R_1(s_1) \wedge R_2(s_2)).$$

Hledáme nejlepší odhady posunů \hat{s}_1, \hat{s}_2 minimalizující plochu průniku

$$[\hat{s}_1, \hat{s}_2] = \arg \min_{\forall s_1, s_2} \{\phi(s_1, s_2)\},$$

kde

$$[\hat{s}_1, \hat{s}_2] = [[\hat{s}_{1,x}, \hat{s}_{1,y}], [\hat{s}_{2,x}, \hat{s}_{2,y}]].$$

Pro dva obdélníky R_i, R_j určíme 4 neznámé posuny $[\hat{s}_{1,x}, \hat{s}_{1,y}], [\hat{s}_{2,x}, \hat{s}_{2,y}]$ ve směru souřadnicových os, dimenze problému je $D=4$. Označme posunuté souřadnice levých dolních rohů obou obdélníků jako

$$\begin{aligned} [x_1, y_1] &= [x_1, y_1] + [s_{1,x}, s_{1,y}], & [x_2, y_2] &= [x_1, y_1] + [w_1, h_1] + [s_{1,x}, s_{1,y}], \\ [x_3, y_3] &= [x_3, y_3] + [s_2^x, s_2^y], & [x_4, y_4] &= [x, y_3] + [w_2, h_2] + [s_2^x, s_2^y]. \end{aligned}$$

Dosadíme-li za $\Delta x, \Delta y$ z (2.1↑), minimalizovaná funkce má tvar

$$\begin{aligned} \phi(s_1, s_2) &= \Delta x \Delta y, \\ &= \max(0, \min(x_1 + w_1 + s_{1,x}, x_3 + w_2 + s_{2,x}) - \max(x_1 + s_{1,x}, x_3 + s_{2,x})) \cdot \\ &\quad \max(\min(y_1 + h_1 + s_{1,y}, y_3 + h_2 + s_{2,y}) - \max(y_1 + s_{1,y}, y_3 + s_{2,y})). \end{aligned} \quad (2.2)$$

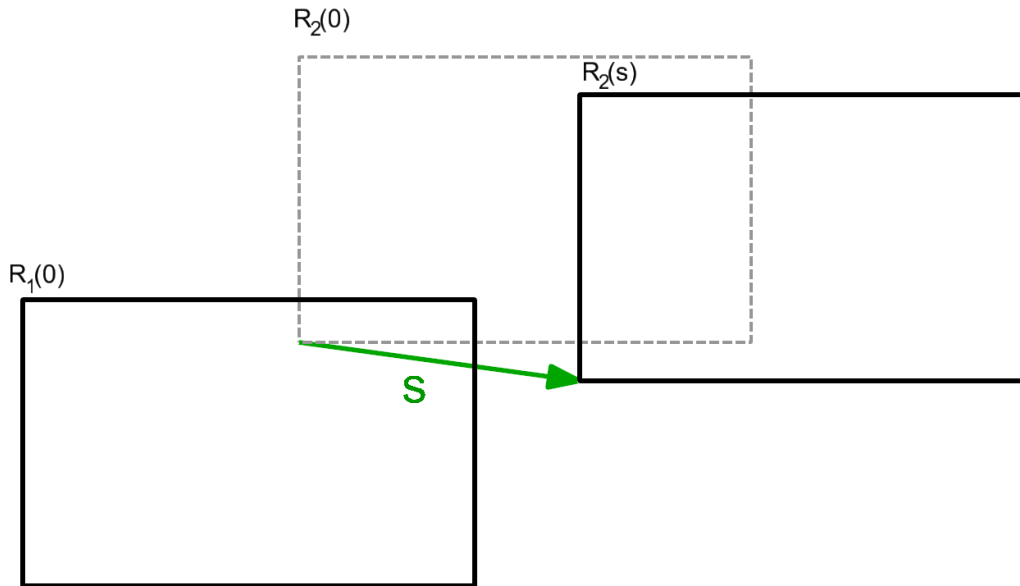
Tato varianta je použitelná za předpokladu, že řízení konfliktů je provedeno s využitím metody inkrementálního vkládání. Řešíme-li současně grafické konflikty mezi větším množstvím prvků, je nutné minimalizovanou funkci zobecnit pro n prvků. Minimalizovaná funkce má tvar

$$\phi(s_1, \dots, s_{2n}) = \sum_{i=1}^n \sum_{j=i+1}^n \phi(s_i, s_j),$$

je zřejmě multimodální. Při optimalizaci, je určováno $2n$ parametrů; dimenze problému je $D = 2n$.

Řešení pro 2 neznámé posuny.

V tomto případě posuneme pouze jedním z obdélníků tak, aby plocha průsečnice byla nulová.



Obrázek č.2.3 Určovaný posuny s , obdélníků R_1, R_2 .

Označme minimalizovanou funkci ϕ , jejíž argumentem je hledaný posuny s jako

$$\phi(s) = S(R_1(0) \wedge R_2(s)) \cup S(R_1(s) \wedge R_2(0)).$$

Pro odvození vztahu předpokládáme, že budeme posouvat pouze s druhým obdélníkem. Hledáme nejlepší odhad posunu \hat{s} minimalizující plochu průsečnice

$$\hat{s} = \arg \min_{\forall s} \{ \phi(s) \},$$

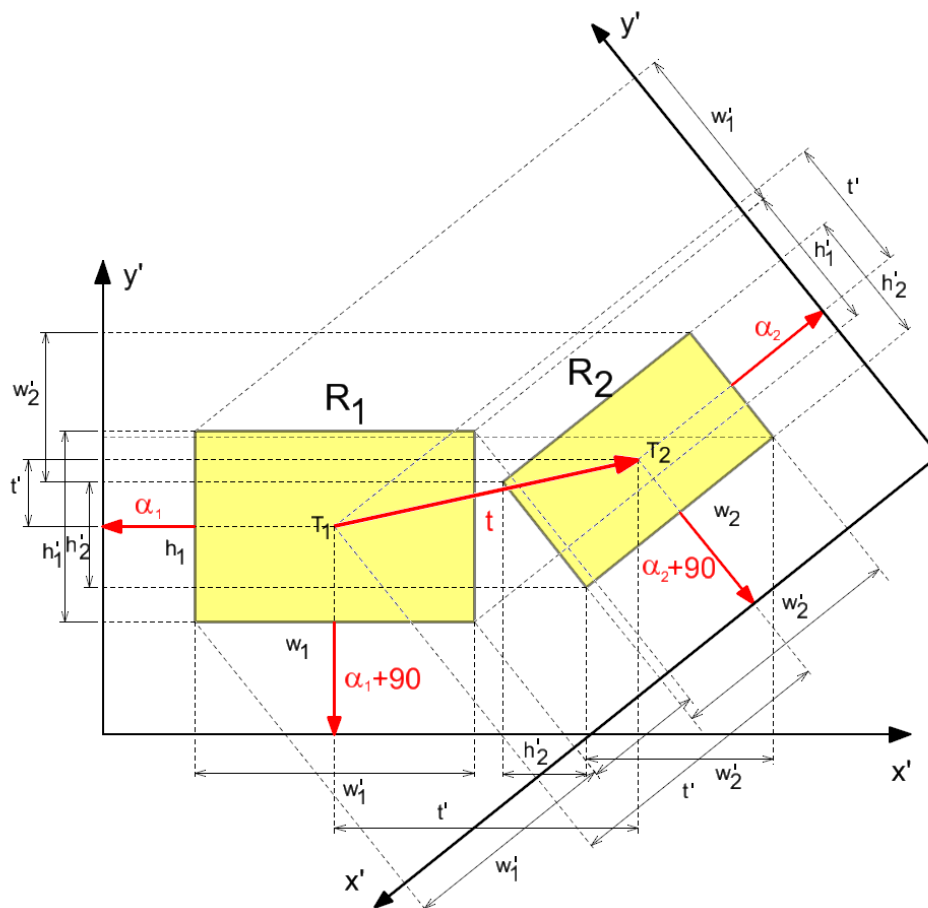
kde $\hat{s} = [\hat{s}_x, \hat{s}_y]$. Pro dva obdélníky R_i, R_j určujeme 2 neznámé posuny \hat{s}_x, \hat{s}_y , ve směru souřadnicových os, dimenze problému je $D = 2$. Minimalizovaná funkce má tvar

$$\begin{aligned} \phi(s) &= \Delta x \Delta y, \\ &= \max(0, \min(x_1 + w_1, x_3 + w_2 + s_x) - \max(x_1, x_3 + s_x)) \cdot \\ &\quad \max(\min(y_1 + h_1, y_3 + h_3 + s_y) - \max(y_1, y_3 + s_y)). \end{aligned} \quad (2.4)$$

Výhodou metody je výrazně rychlejší výpočet za předpokladu, že víme, se kterým obdélníkem budeme posouvat. Pokud tato informace není a-priori známa, lze spočítat pro obě varianty a zvolit tu s kratším posunem. Řešíme-li současně grafické konflikty mezi n prvky, je nutné minimalizovanou funkci zobecnit do tvaru

$$\phi(s_1, \dots, s_n) = \sum_{i=1}^n \sum_{j=i+1}^n \phi(s_i). \quad (2.5)$$

Je nutné si uvědomit, že posunem prvku může dojít ke vzniku nového grafického konfliktu se stávajícími prvky, bez ohledu na fakt, jakým způsobem je realizována strategie řízení grafických konfliktů. V praxi je nutno také provádět testování grafických konfliktů posunutého prvku s ostatními prvky mapy. Lze tak činit metodou hrubé síly či použít vhodnou strategii, např. intervalovou metodu (rozdělením do dlaždic). Opakované hledání kolizí je časově nejnáročnějším krokem algoritmu, vede k značně pomalé implementaci. Řešíme-li úlohu některou ze stochastických metod, z důvodu častého výpočtu hodnoty minimalizované funkce, se úloha stává výpočetně náročnou i pro malé množiny bodů ($n < 1000$).



Obrázek č.2.4 Konflikt MBR, řešení s využitím SAT.

1.2.1.1.2 Grafický konflikt MBR

Na rozdíl od min-max boxů, jejichž hrany jsou rovnoběžné s osami x, y souřadnicového systému, jsou MBR (tj. obdélníky) stočeny o obecný úhel α . Pro nalezení jejich průniku lze použít Separation Axis Theorem. Řešení je založeno na předpokladu, že jsou-li dvě konvexní množiny disjunktní, pak existuje směr, ve kterém jsou jejich průměty disjunktní. Předpokládejme dva obdélníky R_1, R_2 , kde $R_1 = \{x_1, y_1, w_1, h_1, \alpha_1\}$, $R_2 = \{x_2, y_2, w_2, h_2, \alpha_2\}$. Pro těžiště $T_1 = [x_1^T, y_1^T]$, $T_2 = [x_2^T, y_2^T]$, platí

$$\begin{aligned} x_1^T &= x_1 + 0.5(w_1 \cos \alpha_1 - h_1 \sin \alpha_1), & y_1^T &= y_1 + 0.5(w_1 \sin \alpha_1 - h_1 \cos \alpha_1), \\ x_2^T &= x_2 + 0.5(w_2 \cos \alpha_2 - h_2 \sin \alpha_2), & y_2^T &= y_2 + 0.5(w_2 \sin \alpha_2 - h_2 \cos \alpha_2), \end{aligned}$$

vektor t představuje posun obou těžišť

$$t = T_2 - T_1,$$

viz obr. 2.4↑. Označme w'_i, h'_i, w''_i, h''_i průměty stran w_i, h_i, w_2, h_2 obdélníků R_1, R_2 do os souřadnicových systémů $([x_1, y_1], x', y)$, $([x_2, y_2], x', y)$ rovnoběžných se stranami obdélníků. Každou stranu obdélníka promítneme do směrů $\alpha_i, \alpha_i + 90^\circ, \alpha_2, \alpha_2 + 90^\circ$. Průnikem obdélníků je neprázdná množina, pokud pro jejich průměty v tomto směru platí

$$\|t\|_2 \leq 0.5 (\|w'_1\|_2 + \|h'_1\|_2 + \|w'_2\|_2 + \|h'_2\|_2).$$

Obdélníky je nutno posunout ve směru α_i o hodnotu

$$\|s_{\alpha_i}\|_2 > 0.5 (\|w'_1\|_2 + \|h'_1\|_2 + \|w'_2\|_2 + \|h'_2\|_2) - \|t\|_2.$$

Velikost posunu, která zaručuje, že se oba obdélníky neprotínají, představuje maximum ze všech posunů

$$\|s\|_2 > \max_{\forall \alpha_i} \|s_{\alpha_i}\|_2,$$

v korespondujícím směru α_i . Výpočet je numericky stabilní, lze ho zobecnit pro libovolný konvexní polygon.

1.2.1.1.3 Grafický konflikt konvexních obálek

Pro detekci grafických konfliktů a jejich eliminaci lze použít SAT. Každou stranou mnohoúhelníka je vedena přímka, do které jsou promítnuty všechny strany obou konvexních mnohoúhelníků. Následně je určen nejmenší možný posun v daném směru, hledaný posun je maximum nad všemi existujícími. Úloha má kvadratickou složitost vzhledem k počtu vrcholů.

1.2.1.1.4 Grafický konflikt + minimalizace posunů

Další vylepšení metody řešení grafických konfliktů představuje zohlednění velikosti posunů objektu. Jak již bylo zmíněno výše, cílem je eliminace velkých a kartograficky nepřirozených posunů při procházení stavového prostoru. Minimalizovaná funkce $\phi_k(s_1, s_2)$ kombinuje kritérium minimalizující plochu překrytu a čtverec velikostí obou posunů. Pro dvojici obdélníků (min-max boxů) a čtyři určované posuny $s_1 = [s_{1,x}, s_{1,y}]$, $s_2 = [s_{2,x}, s_{2,y}]$, má minimalizovaná funkce tvar

$$\phi_k(s_1, s_2) = \phi(s_1, s_2) + \|s_1\|_2^2 + \|s_2\|_2^2,$$

pro dva neznámé posuny $s = [s_x, s_y]$ má minimalizovaná funkce tvar

$$\phi_k(s) = \phi(s) + \|s\|_2^2.$$

U složitějších prvků (konvexní/nekonzvexní mnohoúhelníky) budou hodnoty posunů vztaheny k těžišti prvku, tvar minimalizované funkce je analogický.

1.2.1.1.5 Grafický konflikt + minimalizace posunů vzhledem k sousedním objektům

Při generalizaci mapy je nutné zachovat kartografické, logické či geometrické vazby, které má generalizovaný prvek vzhledem k ostatním prvkům, primárně k nejbližším prvkům. Jako nejbližší prvky lze chápat k přirozených sousedů generalizovaného prvku. Pro nalezení sousedů lze použít některou z běžně používaných technik, např. Voronoi digram. Označme

$$d_k(E_i, E) = \min_k \{d(E_i, E_j), j = 1, \dots, n, E_i \in E, E_j \in E, j \neq i, \},$$

jako vzdálenost mezi prvkem E_i a k -tým nejbližším prvkem množiny E . Pro generalizovaný prvek E_i , posuny $s = [s_x, s_y]$ má minimalizovaná funkce tvar

$$\phi_k(s) = \phi(s) + d_1^2(E_i, E) + \dots + d_k^2(E_i, E) = \phi(s) + \sum_{l=1}^k d_l^2(E_i, E),$$

a zohledňuje sumu čtverců vzdáleností generalizovaného prvku k jeho k nejbližším sousedům. Chceme-li vzít v úvahu kartografické či logické vazby prvku, minimalizovanou funkci lze vyjádřit jako

$$\phi_k(s) = \phi(s) + \sum_{l=1}^k d_l^2(E_i, E) + \sum_{j=1}^{n_k} d_1^2(E_i, E_j) + \sum_{j=1}^{n_l} d_1^2(E_i, E_j).$$

Poslední dva členy představují sumu čtverců vzdáleností mezi analyzovaným prvkem a n_k prvky, u kterých existuje kartografická souvislost s generalizovaným prvkem. Analogicky, n_l představuje množinu prvků, které jsou s generalizovaným prvkem v logickém vztahu. Tato varianta z kartografického hlediska dosahuje dobrých výsledků, které se blíží ruční kartografické generalizaci.

1.2.2 Omezující podmínky v optimalizaci

Díky existenci omezení vede úloha kartografické generalizace k podmíněné optimalizaci. Tuto úlohu lze řešit s využitím několika strategií, např. s využitím bariérových nebo penalizačních funkcí. V našem případě byla zvolena metoda kvadratické penalty, při které je podmínka $g(x)$ zahrnuta do minimalizované funkce $\phi(x)$ tak, aby její porušení penalizovalo funkční hodnotu minimalizované funkce. Optimalizační úloha

$$\min(\phi(x)), \quad g_i(x) = 0, \quad i = 1, \dots, m,$$

je převedena na

$$\min(\phi(x, \mu)), \quad \phi(x, \mu) = \phi(x) + \frac{1}{2\mu} \sum_{i=1}^m g_i^2(x), \quad \mu > 0,$$

kde μ je penalizační parametr a $\sum_{i=1}^m g_i^2(x)$ penalizační funkce. Pokud je vliv penalizační funkce nulový, nezmění se hodnota lokálního minima $x \equiv x_\mu$. Aby bylo možno metodu použít, omezující podmínky musí být vyjádřitelné v analytickém tvaru $g_i(x) = 0$.

1.2.2.1 Vzdálenostní omezení

Mezi vzdálenostní omezení aplikovaná při generalizaci patří velikost posunu s . Při odsunu prvku je nutné dodržet minimální vzdálenost \underline{d} od jiného prvku či omezit maximální velikost posunu \bar{d} . Penalizační funkce má tvar

$$g(x) = \begin{cases} d, & s < \underline{d}, \\ 0, & s > \underline{d} \wedge s < \bar{d}, \\ d, & s > \bar{d}. \end{cases}$$

Pokud hodnota posunu $s = \sqrt{(s_x^2 + s_y^2)}$ leží v intervalu $[\underline{d}, \bar{d}]$, vliv penalizační funkce se neuplatní.

1.2.2.2 Polohová omezení

Příkladem polohového omezení je zachování vzájemné orientaci jednoho prvku vzhledem k druhému (vlevo/vpravo). Penalizační funkci lze ilustrovat na vzájemné poloze bodu $P = [x, y]$ a přímky $q(P_1, P_2)$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$. Vzájemnou orientaci bodu a úsečky lze určit ze znaménka determinantu

$$t = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix},$$

představující plochu čtyřúhelníku se stranami $u = (u_x, u_y)$ a $v = (v_x, v_y)$, kde

$$\begin{aligned} u_x &= x_2 - x_1, & u_y &= y_2 - y_1, \\ v_x &= x - x_1, & v_y &= y - y_1. \end{aligned}$$

Označíme-li $o = \text{sign}(t)$ orientaci P vzhledem ke q , penalizační funkce má tvar

$$g(x) = \begin{cases} t & o \neq o_{apr}, \\ 0 & o = o_{apr}. \end{cases}$$

Pokud se orientace o prvku po provedení generalizační operace vzhledem k původní orientaci o_{apr} změní, penalizaci představuje čtverec plochy čtyřúhelníku. Analogický princip založený na orientaci vektorů lze aplikovat i na složitější prvky, polygony, či polylinie.

1.2.2.3 Úhlová omezení

Zachování orientace dvou prvků je jedním z nejčastěji použitých úhlových omezení. Úhel ω mezi dvěma liniovými prvky $q(P_1, P_2)$, kde $P_1 = [x_1, y_1]$, $P_2 = [x_2, y_2]$, $r(P_3, P_4)$, kde $P_3 = [x_1, y_1]$, $P_4 = [x_2, y_2]$, lze určit ze vztahu

$$\omega = \arccos \frac{u \cdot v}{\|u\|_2 \|v\|_2}.$$

Při generalizační operaci se vzájemná poloha prvků může změnit, jejich nové souřadnice a příslušné směrové vektory označíme $'$. Penalizační funkce má tvar

$$g(x) = |\omega' - \omega|,$$

kde ω' představuje úhel mezi generalizovanými segmenty

$$\omega' = \arccos \frac{u' \cdot v'}{\|u'\|_2 \|v'\|_2}.$$

Omezující podmínka založená na zachování lokální křivosti je důležitou tvarovou charakteristikou. Pro stanovení lokální křivosti $k(P_i)$ v bodě P_i prvku tvořeného trojicí vrcholů P_{i-1}, P_i, P_{i+1} lze použít např. LLR kritérium

$$k(P_i) = \frac{\|P_{i-1}, P_i\|_2 + \|P_i, P_{i+1}\|_2}{\|P_{i-1}, P_{i+1}\|_2},$$

vyjadřující poměr délky prvku ku vzdálenosti jeho koncových bodů. Po aplikaci generalizačního operátoru lze analogickým způsobem určit křivost $k(P_i)$ vzhledem k modifikovaným vrcholům P_{i-1}', P_i', P_{i+1}' generalizovaného prvku. Pro prvek tvořený n segmenty lze penalizační funkci zapsat ve tvaru

$$g(x) = \sum_{i=2}^{n-1} |k(P_i') - k(P_i)|.$$

Velmi důležitým indikátorem kvality generalizačního procesu představuje schopnost zachovat tvar generalizovaného prvku. Pro popis tvaru prvku E_i lze použít některou z běžných parametrizací, např. Turning funkci, Inner Distance, Fourier Descriptors; tyto

invarianty jsou používány často v oblasti počítačové grafiky. Penalizační funkce vyjadřuje rozdíl tvaru před a po generalizaci, lze ji formálně vyjádřit ve tvaru

$$g(x) = \varphi(E'_i) - \varphi(E_i).$$

Pro zjištění změny tvaru jsou využívány různé metriky, nejčastější jsou L_1 , L_2 , L_p . V rámci projektu byla otestována parametrizace založená na turning funkci, penalizační funkce má tvar

$$g(x) \equiv d_2(E_i, E'_i) = \left\| \Theta_{E_i}(s) - \Theta_{E'_i}(s) \right\|_2 = \left(\int_0^1 \left| \Theta_{P_i}(s) - \Theta_{P'_i}(s) \right|^2 \right)^{1/2},$$

kde Θ_{E_i} , $\Theta_{E'_i}$ představují turning funkce prvku E_i před a po generalizaci.

1.2.3 Zhodnocení optimalizační strategie

Optimalizační strategie pro řešení kartografické generalizace byla široce testována nad různými typy dat. Představovala globální optimalizaci s omezujícími podmínkami, které byly implementovány metodou kvadratické penalty. V rámci projektu byly otestovány všechny výše uvedené omezující podmínky. Řešení grafických konfliktů bylo realizováno metodami hrubé síly a inkrementálním vkládáním. Jako optimalizační strategie byla použita diferenciální evoluce. Celkové zhodnocení optimalizační strategie je nutné provést ve dvou základních rovinách.

1.2.3.1 Kartografický pohled

Z kartografického pohledu dosahovaly algoritmy dobrých výsledků, velký význam mělo zavedení omezujících podmínek. Jako zcela nepostačující se ukázal požadavek minimalizace grafických konfliktů. Při náhodném procházení stavového prostoru docházelo k nepřirozeným posunům prvků, došlo tak ke ztrátě kartografických či logických vazeb mezi jednotlivými prvky, což postihlo zhruba 65% prvků mapy. V extrémních případech byly generalizované prvky posouvány mimo min-max box zkonstruovaný nad původní mapou, tato situace byla typická pro území s velkou grafickou zaplněností mapy. Prostorové rozložení generalizovaných prvků vedlo k výraznému ztížení čitelnosti a přehlednosti mapy, v některých případech mapa zcela ztratila čitelnost.

Zavedení dodatečné podmínky spočívající v minimalizaci posunu prvků vedlo k výraznému zlepšení výsledku, velikost posunů s se výrazně zkrátila. V řadě případů však opět došlo k přerušení logických či kartografických vazeb mezi prvky, které se projevily vznikem kartograficky nepřirozených situací (30% počtu prvků). Na rozdíl od předcházejícího přístupu však nedošlo k výrazné ztrátě přehlednosti či čitelnosti mapy.

Další kvalitativní posun představoval zohlednění vztahu generalizovaného prvku ke svým geometrickým sousedům (tj. přirozeným sousedům), popřípadě k prvkům, se kterými měl logické či kartografické vazby. V generalizované mapě se kartograficky nekorektní či nepřirozené prvky vyskytovaly poměrně málo často, tvořily 10-15% z celkového počtu prvků, stále však byl jejich vliv viditelný.

Zavedení vedlejších podmínek snížilo vliv nekorektně generalizovaných prvků na zhruba 10%, vedlo také k celkovému vylepšení kartografické kvality generalizovaného mapového díla (nevznikaly hrubě nekorektní prvky, pouze nevhodné). Zbývající situace, které nebylo možné úspěšně dořešit automatizovaně, mohou být následně opraveny ruční editací.

Metoda inkrementální konstrukce dosahovala o něco horších výsledků než metoda inkrementálního vkládání, která dříve přidané a generalizované prvky chápala jako neměnné. Nově přidávaný prvek byl generalizován tak, aby jeho nové vlastnosti vyhovovaly již vytvořené části mapy, která se však již tomuto prvku nijak nepřizpůsobovala. Tyto nedostatky se projevily zejména u liniových prvků diagonální orientace, které bylo obtížnější umístit na novou pozici tak, aby korektně navazovaly na

již generalizovanou mapovou kresbu. Velmi dobrých výsledků dosáhla metoda inkrementálního vkládání, u které dochází k opravě již dříve generalizované části mapy.

1.2.3.2 Informatický pohled

Kartografická generalizace, řešená formou nekonvexní optimalizace s řadou vedlejších podmínek, kdy je optimalizovaná funkce multimodální, vede k NP problému. Pro jeho řešení jsou použity nedeterministé techniky, které přinášejí některé níže uvedené nevýhody.

Řada optimalizovaných kritérií je výpočetně složitá, zejména nalezení k přirozených sousedů za předpokladu, že analyzované prvky tvoří 1D/2D entity. Tento krok vede ke konstrukci Voronoi diagramů, jejichž generátory jsou tvořeny 1D/2D prvky.

Další obtížný krok představuje detekce kolizí prvků a jejich posun tak, aby nedocházelo ke grafickým konfliktům. U nekonvexních prvků není možné použít SAT, nelze tedy přímo určit hodnotu a směr posunu s , tuto operaci je nutné provádět opakovaně (stochasticky). Aproximace prvku konvexní obálkou umožňuje problém řešit přímo. Pro velké prvky zejména liniového charakteru s diagonální orientací, však analýza polohy konvexních obálek může vést ke vzniku falešných kolizí. Použití aproximace bufferu vede k výraznému zvýšení výpočetního času spojeného s řešením grafických konfliktů.

Nejvýraznější vliv na výpočetní složitost mělo zavedení tvarových kritérií formou vedlejších podmínek, zejména u rozsáhlejších prvků je nalezení vhodné parametrizace a aplikace srovnávací metriky výpočetně náročným krokem, počet operací je zpravidla kvadratickou funkcí počtu vrcholů.

Stochastické metody optimalizace vyžadují častý výpočet hodnoty optimalizované funkce, která musí zohlednit veškerá minimalizovaná kritéria i omezení. V každém dílčím kroku výpočtu hodnoty optimalizované funkce je tedy nutné detekovat grafické konflikty, nalézt přirozené sousedy, prvky logicky či kartograficky spojené s generalizovaným prvkem, zjistit existenci omezujících podmínek vzhledem k těmto prvkům, realizovat je, provést tvarovou parametrizaci prvku a následně navrhnout nové posuny splňující výše uvedená kritéria.

Takto prezentovaná optimalizační strategie je vhodná pouze pro malé množiny prvků ($n < 1000$), výpočetní časy se pohybují v řádech hodin. Optimalizační metodu nelze použít pro generalizaci rozsáhlejších území, výpočetní nároky na hardware jsou příliš vysoké. Tyto výsledky se potvrdily při práci množinami prvků představovaných mapovým listem ZM 10. Doba běhu algoritmu je navíc výrazně ovlivněna grafickou zaplněností mapového listu.

V praxi se ukazuje, že stochastická optimalizace multimodální funkce s podmínkami je výpočetně náročný proces, jehož některé kroky by mohly být zjednodušeny vhodnou heuristikou. V takovém případě nemusí být výsledkem generalizačního procesu nejlepší řešení, avšak zpravidla bude postačovat některé ze suboptimálních řešení. V dalších kapitolách proto zmíníme techniky, které se snaží problematiku kartografické generalizace řešit přibližnou formou. Tento přístup však má některé nevýhody, za cenu snížení výpočetní náročnosti může dojít ke zhoršení kvality nalezeného řešení. Pro většinu aplikací však bude suboptimální varianta postačující, případné nedořešené situace lze opravit manuálním zásahem operátora.

1.2.4 Batch Processing

Batch processing představuje nejjednodušší metodu řízení generalizačních operací. Jedná se o historicky nejstarší strategii řízení kartografické generalizace. Z kartografického pohledu neprovádí komplikované řízení generalizačního procesu. Jednotlivé generalizační operátory jsou aplikovány sekvenčně, pořadí operátorů je zpravidla předem dané a vychází z přístupu používaného v klasické kartografii. Postup lze vyjádřit následujícím schématem:

```

applyOperator_1(items)    //Aplikace generalizačního operátoru 1
...
applyOperator_k(items)    //Aplikace generalizačního operátoru k
    
```

Připomeňme, že u generalizačních operací není zachována komutativita; aplikací dvou operátorů v různém pořadí vede zpravidla k rozdílnému výsledku. Pořadí vychází zpravidla z empirické zkušenosti. Každý generalizační operátor je aplikován pouze jednou, nedochází k opakování dílčích kroků. Strategie nedisponuje žádnou vnitřní inteligencí, neprovádí hodnocení výsledků; efektivita jednotlivých generalizačních kroků není posuzována.

Z kartografického hlediska nedosahuje příliš dobrých výsledků, jednorůchodový přístup není schopen zajistit optimální rozložení generalizovaných prvků na mapě. Při ruční kartografické generalizaci jsou jednotlivé operátory zpravidla aplikovány opakovaně, pořadí jejich použití není pevné, ale závisí na řadě kartografických faktorů (měřítko, typ, obsah mapy, charakter generalizovaného území). Metoda je implementačně jednoduchá, lze ji přirovnat k jednomu iteračnímu kroku prováděnému v rámci optimalizační strategie. Nedosahuje dobrých výsledků, díky absenci opakování nelze efektivně řešit grafické konflikty, a to zejména v místech s vyšší grafickou zaplněností mapy.

1.2.5 Rule Based Generalization

Tato technika se snaží dosáhnout nalezení optima simulací chování kartografa. Na rozdíl od předchozí metody nepoužívá sekvenční přístup s pevně daným pořadím, umožňuje dle požadavku jednotlivé generalizační operátory aplikovat opakovaně. Strategie provádí dílčí ohodnocení každého kroku s cílem dosáhnout celkového řešení s vysokým ohodnocením. Generalizace je prováděna v souladu s přístupem používaným při ruční generalizaci; pořadí volání jednotlivých generalizačních operátorů respektuje kartografická pravidla.

```

while res decreases          //Opakuj, dokud existuje zlepšení
    if (condition_1)         //Je -li splněna podmínka, aplikuj
        operátor
            applyOperator_1(items);
    else                       //V opačném případě aplikuj jiný
        operátor
            applyOperator_2(items);
    res = φ(items)           //Ohodnot kvalitu řešení

    ...
    if (condition_k)
        applyOperator_k(items);
    else
        applyOperator_k(items);
    res = φ(items)
    
```

Na základě ohodnocení dílčího kroku algoritmus umožňuje zvolit další postup, na rozdíl od předcházející metody se kód může větvit. Pořadí prováděných operací, stejně tak jako počet opakování, se mohou dle potřeby měnit. Ohodnocení dílčích operací využívá obecnou nákladovou funkci φ .

Algoritmus probíhá tak dlouho, dokud aplikace generalizačních pravidel způsobuje snižování hodnoty nákladové funkce. Generalizační operátory zpravidla nevyužívají stochastický přístup. Pro řízení grafických konfliktů používají strategie uváděné v kapitole 1.2↑.

Z kartografického hlediska dosahuje výrazně lepších výsledků než dávkové zpracování. Opakované volání generalizačních operátorů v různém pořadí umožňuje řešit i složitější grafické konflikty, které při jednoručkové variantě nelze zvládnout. Při řešení grafických konfliktů jsou používány přibližné metody řešení grafických konfliktů, velmi často je využíván try-catch přístup. Řešení dílčího kroku je hledáno opakovaně, a to za použití heuristiky tak, že dochází ke změně hodnoty hledaného parametru s pravidelným přírůstkem dokud dochází ke grafickému konfliktu.

Z implementačního hlediska bývá Rule Based generalization označována jako *while-if-else* strategie. Jedná se o jednoduchou deterministickou optimalizační techniku založenou na direct-search přístupu, opakované volání vede ke stejnému výsledku. Strategie vede k nalezení lokálního minima, negeneruje špatná řešení (avšak ani řešení velmi dobrá), zpravidla vyžaduje předzpracování dat. Výpočetní časy jsou výrazně lepší než u optimalizačních strategií, tato technika může být použita i pro rozsáhlejší datové množiny. V místech s vysokou grafickou zaplněností mapy (intravilán) nemusí být snaha nalézt deterministické řešení úspěšná, popř. může být generováno řešení, které není kartograficky korektní.

Pro hodnocení efektivity dílčích kroků mohou být používány analogické funkce jako v případě optimalizačních strategií, hledání řešení minimalizující optimalizovanou funkci však probíhá jednodušším způsobem.

Rule Based Generalization představuje transformaci kartografických pravidel do formalizovaného jazyka. Vzhledem k faktu, že řada kartografických pravidel má ryze empirický charakter, neexistuje přímý způsob, jak úspěšnost této operace verifikovat. Tento fakt, jak již bylo výše zmíněno, negativně ovlivňuje efektivitu generalizačních operací.

1.2.6 Multi Agent Systems

Multiagentní systémy představují jednu z pokročilých technik řešení rozhodovacích problémů, u kterých je optimální řešení vzhledem ke komplexitě problému obtížně určitelné. Představují jedno ze základních paradigmat informatiky. Jedná se o jednu z technik umělé inteligence, která při řešení problému používá přístupy běžné lidskému myšlení (konkurence, spolupráce). Postup řešení složitého problému je založen na jeho dekompozici na problémy jednodušší, které lze řešit snadněji. Řešení původního problému se snažíme nalézt řešením dílčích podproblémů autonomními procesy, tzv. *agenty*.

Řešení problému je hledáno na základě kooperace jednotlivých agentů, každý agent disponuje vlastní autonomií. Na základě vstupních informací a vnitřního rozhodovacího procesu provádí agent dílčí řešení problému, ke kterému je kompetentní. Dynamická koordinace jejich činnosti vede k nalezení optimálního řešení problému. Veškeré operace agenta jsou reakcí na stav, ve kterém se objekt nachází. Klíčovou vlastností agenta je schopnost reakce agenta na změnu stavu a výběr akce.

Zjednodušení problému.

Aby bylo možné multiagentní systémy použít, je nutné do problematiky vnést dvě základní omezení:

- diskretizace prostředí na konečný počet stavů $E = \{e_1, \dots, e_m\}$, na které agent reaguje,
- diskretizace reakcí agenta na konečný počet akcí $A = \{a_1, \dots, a_m\}$.

Tento krok lze provést pro libovolný problém s libovolnou mírou podrobnosti vzorkování. Je však nutné vzít v potaz, že restrikce počtu stavů a reakcí vede k nepřesnému chování agenta, které je důsledkem použitého konceptuálního modelu. Předpokládáme, že následující stav prostředí je funkcí předchozích stavů prostředí a předchozí akce agenta. Dále předpokládáme, že prostředí je nedeterministické; není jasné, jaký stav po vykonání akce nastane.

1.2.6.1 Princip agentního systému

Prostředí ε je definováno jako uspořádaná trojice

$$\varepsilon = (E, s_0, T),$$

kde S je množina funkčních stavů, s_0 je počáteční stav, a T je transformační funkce stavu prostředí. Předpokládejme, že R je množina všech posloupností nad E a S a R_A podmnožina posloupností R , která končí akcí A . Transformační funkce T je definována jako $T:R_A \rightarrow 2^E$. Pokud $T(r) = \emptyset$, nejsou k dispozici žádné další stavy, a systém ukončil svůj běh.

Činnost agenta A_g v prostředí $E = (S, s_0, T)$ označujeme jako následující posloupnost kroků:

1. s_0 je poč. stav E
2. $a_0 = Ag(s_0)$
3. pro $\forall u > 0$:
4. $e_u \in T(r)$.
 $a_u = Ag(e_u)$

Prostředí E nachází v nějakém výchozím stavu e_0 . Na základě tohoto stavu agent zvolí akci a_0 , jejímž důsledkem je přechod prostředí do nového stavu e_1 . Důsledkem funkce agenta je střídající se posloupnost stavů a akcí

$$r = \{e_0, a_0, e_1, a_1, \dots, e_m, a_m\}.$$

Každý agent je do jisté míry autonomní, nese informaci o tom, co má udělat, nikoliv jak, řešení konkrétní situace zpravidla obsahuje prvek náhodnosti. Pro posouzení efektivity existuje účelová funkce $\varphi: E \rightarrow \mathbb{R}$, umožňuje ohodnocovat jednotlivé stavy. Cílem je provedení takové akce, která vede k co největšímu snížení hodnoty účelové funkce mezi dvěma či více stavy.

1.2.6.2 Agentní systém v oblasti kartografické generalizace

Výše navržený přístup lze využít i při automatizaci kartografické generalizace. Předpokládáme, že představuje natolik složitý problém, jehož komplexní a univerzální řešení nelze nalézt běžnými optimalizačními technikami.

Proces kartografické generalizace lze dekomponovat na elementární operace realizované generalizačními operátory; každý generalizační operátor může být modelován agentem. Agent má k dispozici informace o tom, na jaké jaký může být generalizační operace aplikována, jaký je její výsledek, v jakém intervalu měřítek může být použita, které vazby mezi prvky musí být zachovány (popř. mohou být vypuštěny), pro který druh map či charakter území je generalizační operátor typický. Každý agent tedy disponuje pouze omezenou informací o svých vlastnostech a možnostech vzhledem k okolním prvkům, nemá informaci o celkovém řešení problému. V důsledku realizace generalizačních operací se mění obsah i vlastnosti mapy, na změnu stavu jednotliví agenti následně reagují.

Příkladem použití agentního systému může být implementace operátoru odsunu aplikovaného na prvek. Na základě okolí prvku, kartografických, logických či geometrických vazeb jsou definovány stavy, kterých prvek může dosáhnout. Kartografická pravidla lze tímto přístupem snadno formalizovat. Uvedme několik příkladů: Pokud se v okolí prvku nachází prvek s vyšší prioritou, prvek s vyšší prioritou se neposouvá. Pokud by při posunu prvku určitým směrem došlo ke grafickému konfliktu generalizovaného prvku s jiným prvkem, zvolíme jiný směr posunu. Pokud má posouváný prvek geometrickou, kartografickou, či logickou vazbu na jiný prvek, nemůže být posunut tak, aby tyto vazby zanikly.

Problematika agentních systémů a jejich využití v kartografii představuje téma, které svojí komplexitou přesahuje tento materiál. Z kartografického pohledu dosahuje dobrých výsledků, dekompozice problému na elementární úlohy, princip agentů, jejich autonomie,

vzájemná konkurence či soutěživost agentů, zavedení prvku náhodnosti, vedou k návrhu řešení, které je blízké kartografickému řešení. Z implementačního hlediska je negativem dlouhá doba běhu, která závisí na množství stavů a akcí agenta a na počátečním stavu.

1.2.7 Heuristika

Další strategii, kterou lze použít jak pro řízení kartografické generalizace, tak pro implementaci dílčích generalizačních operátorů, či řešení grafických konfliktů, představuje heuristika. Používá se při řešení problémů, u kterých není znám exaktní algoritmus. Heuristiky procházejí stavovým prostorem a hledají přípustné řešení, tj. řešení, které není nejlepší možné, ale zároveň není špatné.

1.2.7.1 Greedy strategie

Greedy strategie představují jednu z nejčastěji používaných heuristických strategií. Algoritmus se snaží nalézt optimum (globální/lokální) nákladové funkce tak, že v každém okamžiku používá lokálně optimální krok. V každém kroku ze skupiny řešení vybírá takové, které má minimální sumu ohodnocení. Tato strategie je účinná při hledání lokálního minima, pro hledání globálního minima již méně. Greedy strategie nepřipouští dočasné zhoršení stavu, které může zamezit generování řešení v oblasti kolem lokálního minima.

Pro účely kartografické generalizace byly implementovány jednoduché heuristické strategie:

- *Try and catch*

V rámci omezujících podmínek je vygenerováno náhodné řešení, jeho kvalita je následně ověřena účelovou funkcí φ . Z praktického pohledu se jedná o implementaci přístupu pokus-omyl. U graficky zaplněných míst, zejména v oblasti intravilánu, může být počet pokusů generující přípustné řešení poměrně vysoký, výpočet ohodnocovací funkce se vzhledem k počtu podmínek stává výpočetně náročným. Typický příklad použití představoval posun nekonvexních prvků tak, aby nebyly v grafickém konfliktu s ostatními prvky. Opakovaně je generováno náhodné řešení, či řešení ovlivněné výše definovanými podmínkami tak dlouho, dokud není ohodnoceno jako správné.

- *Snižující se krok*

V rámci omezujících podmínek je zvoleno řešení blízké maximální hodnotě omezující podmínky. Pokud toto řešení nevyhovuje, hledá se takové, jehož hodnota (vlastnost) je menší o předem danou hodnotu kroku. Dokud správné řešení není nalezeno, popř. jsme neprošli všechny možné varianty, je hledáno nové, bližší řešení. Typickým příkladem může být implementace odsunu, kdy zvolíme jako výchozí polohu takovou vzdálenost, při které jsou zachovány existující kartografické či logické vazby. Dokud není nalezeno řešení eliminující grafické konflikty, hodnotu posunu neustále snižujeme o předem zadaný krok. U graficky zaplněných míst nemusí být takové řešení nalezeno, v takovém případě je nutno provést ruční dořešení situace.

- *Propagační přístup*

Pokud předchozí strategie nevedou k cíli, je možné generalizační operátor propagovat vzhledem k okolním prvkům. Při tomto principu se jeho vliv uplatní i na okolní prvky a to v závislosti na vzdálenosti od generalizovaného prvku. Myšlenka vychází z empirického přístupu k řešení problému. Pokud není možné nalézt řešení pro jeden prvek, provede se úprava jeho okolí, a operace se opakuje, dokud není dosaženo požadovaného cíle. Míra propagace může klesat lineárně, kvadraticky, či exponenciálně se vzdáleností od generalizovaného prvku. Pro prvky ovlivněné propagací lze použít výše uvedené optimalizační strategie.

Jednoduché heuristické techniky dosahovaly dobrý poměr cena/výkon a demonstrovaly svoji použitelnost jak při řešení lokálních konfliktů, tak i pro řízení strategie kartografické generalizace, zejména ve spojení s inkrementální metodou.

1.2.7.2 Simulované žíhání

Simulated Annealing představuje komplexní optimalizační strategii založené na stochastickém přístupu. Na rozdíl od jednoduchých heuristik umožňuje zabránit uváznutí řešení v oblasti lokálního minima, připouští dočasné zhoršení stavu modelovaného prostřednictvím účelové funkce. Vychází z fyzikálního modelu, při kterém jsou u zahřívání tělesa pohyby atomů funkcí jeho teploty, při ochlazení se velikost kmitů zmenšuje, a atomy se usazují v rovnovážné poloze. Jako rovnovážná poloha je chápáno globální/lokální minimum optimalizované funkce. Strategie je výpočetně náročná, dosahuje horších výsledků než genetické algoritmy. Pro účely projektu nebyla tato strategie použita.

1.2.7.3 Genetické algoritmy

Genetické algoritmy patří mezi nejúčinnější optimalizační metody. Vznikly jako náhrada simulovaného žíhání, které u složitých optimalizačních úloh (multimodální funkce, velká dimenze problému) selhávalo. Genetické algoritmy pracují s populací možných řešení, na které jsou aplikovány zákony evoluční biologie. V dalších iteracích přežívají pouze kvalitní jedinci, kvalita se měří prostřednictvím fitness funkce. V každé iteraci jsou na jedince aplikovány operátory křížení a mutace, při kterých z několika jedinců vzniká jedinec lepších vlastností s celkovým cílem vygenerovat kvalitnější populaci.

Pro automatické řízení kartografické generalizace byla použita technika diferenciální evoluce, která řešila problematiku podmíněné optimalizace, účelová funkce φ byla multimodální. Jako mutační chéma bylo použito DE/RAND/1 schéma, velikost iniciální populace byla volena jako $NP=10D$, kde D je dimenze problému (tj. počet určovaných parametrů). Pro strategii řízení generalizace založenou na metodě hrubé síly $D=n$, kde n představuje počet všech prvků na mapě. Předpokládáme-li řešení problému nad mapovým listem, kde se počet prvků pohybuje v řádech tisíců, jedná se o optimalizační úlohu středního rozsahu. Strategie inkrementálního vkládání vedla i v místech s velkou mírou grafické zaplněnosti mapy k optimalizačnímu problému malého rozsahu, hodnoty D se pohybovaly v řádek jednotek či desítek prvků.

Tato technika poskytovala dobré výsledky pouze pro malé množiny dat. Již pro středně rozsáhlé problémy ($n > 1000$) se výpočetní časy pohybovaly v dobách hodin či dnů, velkou roli hrálo výpočetně náročné hodnocení grafických konfliktů použité při výpočtu účelové funkce. Pro rozsáhlé množiny dat tato strategie zřejmě nebude prakticky použitelná, v této oblasti je však nutný další výzkum.

1.3 Závěr

Kartografickou generalizaci lze v širším pohledu chápat jako optimalizační úlohu. Vzhledem k zaměření projektu na topografické mapy středních měřítek představoval odsun nejčastější metodu řešení grafických konfliktů. Grafický konflikt lze chápat jako přiblížení dvojice prvků pod limitní vzdálenost (dojde k jejich grafickému slití), nevhodným umístěním jednoho prvku dovnitř jiného, či překrytím části prvku jiným prvkem.

Tento materiál představil základní strategie řešení grafických konfliktů. Kromě metody hrubé síly, která je zaměřena spíše na množiny malého rozsahu, prezentoval inkrementální a heuristický přístup umožňující řízené postupné přidávání prvků původní mapy do mapy generalizované. Nejprve jsou do mapy přidávány prvky, které mohou díky své velikosti či tvaru způsobit větší množství grafických konfliktů, jako poslední malé či tvarově jednoduché prvky. Myšlenka je založena na faktu, že konflikty složitých prvků je jednodušší řešit nad málo zaplněnou mapou než nad mapou zaplněnou.

Důležitou roli hraje způsob detekce grafických konfliktů. U nekonvexních prvků komplexních tvarů je vyhledávání konfliktů založeno na testování průniku prvků či průniku jejich ekvidistant, tzv. bufferů. Tato operace je výpočetně náročná, materiál se snaží prvky a jejich buffery nahradit zjednodušenou reprezentací.

Pokud u dvou prvků dochází ke grafickému konfliktu, je plocha jejich průniku nebo plocha průniku jejich bufferů nenulová. Odstranění grafických konfliktů povede k situaci, kdy celková plocha průníků všech kombinací prvků či jejich bufferů bude nulová. Na této myšlence je založeno geometrické kritérium minimalizující grafické konflikty nad mapou, lze ho vyjádřit prostřednictvím funkce. Tohoto cíle lze dosáhnout několika způsoby: posuny prvků, jejich tvarovou modifikací, či kombinací obou postupů. Vzhledem k měřítku a typu mapy je převládající operací posun prvků. Taková úloha má nekonečně mnoho řešení, vede k nepodmíněné optimalizaci. Při stochastickém procházení stavového prostoru mohou být generována taková řešení, kdy jsou hodnoty posunů prvků tak velké, že dojde ke ztrátě vztahů mezi generalizovanými prvky.

Jeden ze způsobů, jak tento problém zmírnit, představuje zahrnutí absolutní velikosti posunu prvku do minimalizované funkce či zahrnutí velikosti posunu prvku vzhledem k jeho přirozeným sousedům nebo prvkům, se kterými je v nějakém vztahu. Zohlednění těchto podmínek vede k výraznému vylepšení výsledků generalizačních algoritmů. Důležitým krok představuje zahrnutí základních omezujících podmínek kartografického charakteru, jedná se zejména o úhlová, tvarová, či polohová kritéria. Úloha kartografické generalizace je transformována na optimalizační úlohu s omezením, jednotlivá kritéria jsou zohledněna prostřednictvím penalizační funkce (metoda kvadratické penalty). Takto navržený princip je robustní, avšak výpočetně velmi složitý, představuje NP problém. Doba nalezení vhodného řešení metodou diferenciální evoluce pro mapový list se pohybovala v řádech hodin či dnů (dle grafické zaplněnosti). Místo exaktní optimalizace bylo nutné přistoupit k nějaké jednodušší optimalizační metodě. Z hlediska poměru cena/výkon se jako optimální jeví zejména použití heuristik (zejména greedy strategií), kdy jsou grafické konflikty řešeny metodami try-catch či postupným snižováním velikosti kroku či propagačním přístupem.

Materiál si neklade za úkol vyřešit problematiku kartografické generalizace map středních měřítek v komplexní formě, ale upozornit na některé zajímavé, důležité a doposud opomíjené postupy a přístupy související s kartografickou generalizací. Chápe kartografickou generalizaci jako určitou formu optimalizace, vzhledem ke komplexitě problému se však nesnaží nalézt řešení nejlepší, ale spokojí se s řešením optimálním.

2. Stávajícího stav implementace automatizované generalizace

Tato kapitola byla sestavena jako výsledek studia dostupných materiálů, zejména podkladů workshopů komise ICA pro generalizaci a vícenásobnou reprezentaci [1]. Jako výtečný komplexní materiál shrnující poslední období se ukázala studie EuroSDR [2]. Dalším zdrojem informací byly konzultace s jednotlivými zahraničními řešiteli na workshopech pracovní skupiny, kterých jsme se účastnili [3], [4]. Dalším vstupem k doplnění a aktualizaci informací v [2] byly prezentace jednotlivých národních řešení na workshopu ICA organizovaná v roce 2015 [4]. Obdobně přínosný workshop, zaměřený na řešení na platformě ArcGIS zorganizovala také ESRI Swiss [5]. Využitím těchto podkladů se podařilo sestavit velice aktuální a souhrnný materiál popisující stávající stav v oblasti automatizované generalizace, který předkládáme v této kapitole.

2.1 Rané fáze, 80-léta

Již v raných fázích kartografické tvorby s využitím výpočetní techniky byla řešena potřeba do procesu zařadit generalizační blok. Ten se v prvopočátku zaměřil na operace náročné pro ruční zpracování, například zjednodušení kresby (simplification) či výběr zobrazených prvků (data selection). Z tohoto období pochází velké množství efektivních generalizačních algoritmů, které byly do procesu zpracování řazeny postupně za sebou v tzv. dávkovém zpracování (batch processing). Velice brzo se prosadila potřeba využití kartografických databází jako zdroje dat pro GIS a analytické úlohy. Tyto algoritmy byly proto optimalizovány aby pracovaly rychle i na pomalém hardware a nevyžadovaly velké množství paměti, cenou za to ovšem byla velice nízká kvalita výsledné kresby. Tyto algoritmy se proto prosadily v oblasti zjednodušování dat GIS (modelová generalizace), v oblasti kartografické tvorby nikdy nenašly širšího uplatnění. To souvisí se vznikem a rozvojem oboru geoinformatika v devadesátých letech, jehož zaměření není zcela v souladu s potřebami kartografické tvorby.

2.2 Přejít k digitální kartolitografii, 90-tá léta

S rozvojem schopností systémů GIS v oblasti prezentace dat došlo k postupnému k nahrazení práce karto-litografa výpočetní technikou, kartografické práce se komplexně přesunuly do profese kartograf-redaktor.

To s sebou přineslo kromě úspory pracovní síly zejména vysokou ostrost tisku kresby díky využití osvitových jednotek k tvorbě tiskových podkladů. Následně nahradilo mnoho mapovacích agentur tisku přímými barvami čtyřbarvotiskem. Jeho zásadní omezení, tj. malý rozsah barevnosti a nízká hodnota reálně dosažitelného LPI bylo eliminováno drobnou korekcí barevnosti mapových děl, zejména prvků kreslených ve vlasových čarách (vrstevnice, vodní toky). Přejít na technologii čtyřbarvotisku umožnil zvýšit počet použitých barev mapových děl bez zvýšení nákladů a tím mnohde odlehčil kresbě a snížil potřebu generalizace z důvodu překrytu prvků (fight for space).

Zároveň v tomto období většina agentur dále rozvíjela metody generalizace v rozsahu, který mu umožňovala příslušná softwarová platforma. V tomto období bylo velice obtížné stavět produkční linku z různorodých komponent, protože akvizice hlavního systému s sebou přinášela stanovení výrobního celého řetězce (například ArcInfo=Unix SVR6=HP Workstations=plotry Xerox ...). Výměna dat mezi systémy, jejich zařazení do jedné počítačové sítě a podobné požadavky dnes zcela běžné, byly v tehdejší technickém kontextu zcela namyslitelné.

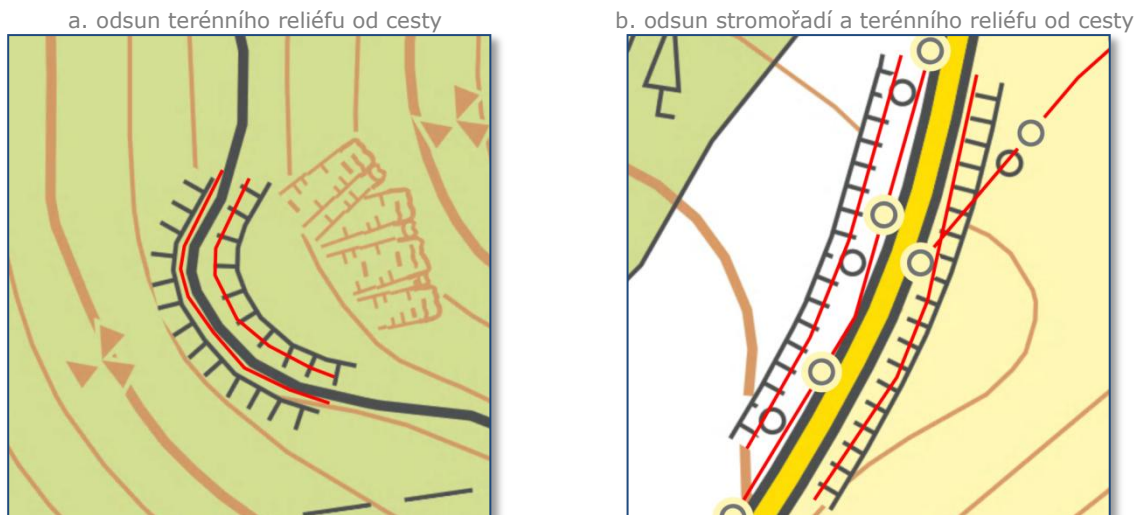
2.2.1 Generalizace založená na pravidlech

Postupem času bylo původní dávkové zpracování modernizováno do generalizace založené na pravidlech (Rule Based Generalization). Při této strategii, velice zjednodušeně, definujeme sekvenci dvojic pravidlo-akce. Poté postupně procházíme jednotlivá pravidla a na prvky, které dané pravidlo splňují aplikujeme příslušnou akci. Tato metoda umožňuje aplikovat více akcí na tytéž prvky, pokud splňují více pravidel.

Zároveň, definicí obecných a upřesňujících pravidel je možno poměrně jednoduše definovat případné hierarchické vazby.

Definice podmínky může být jak velice jednoduchá "pro všechny pozemní komunikace", tak složitější "propustky kde se kříží jednočarý vodní tok a pozemní komunikace" i velice sofistikovaná. Obdobně akce může být jednoduchá "proved' vyhlazení kresby linie" či složitější "vypočti parametry natočení a šíře značky propustku a zaznamenej je v parametrech výjimky pro kartografickou reprezentaci v geodatabázi".

Princip generalizace založené na pravidlech si demonstrováme na jednoduchém příkladu odsunu terénního reliéfu od cesty, respektive postupného odsunu. V červené barvě a symbolické značce je naznačen skutečný průběh prvků.



Obrázek č.1 Odsun liniových prvků

Obě situace kartograf vyřeší aplikací operátoru odsun na příslušné liniové prvky ve správném pořadí.

Definice pravidel může vypadat následovně:

č.	Pravidlo	Operace
RB A	Jestliže se kresba jednoho terénního reliéfu zasahuje do kresby cesty	Odsuň kresbu terénního reliéfu jednoduchým algoritmem <i>Displace A</i> .
RB B	Jestliže se překrývají kresby více prvků mezi sebou a lze je seřadit podle sestupně podle důležitosti prvků a polohy (zde silnice, stromořadí a reliéf)	Postupně odsuň kresby prvků sofistikovaným algoritmem <i>Displace B</i> .

Tabulka č.1 - Pravidla a akce pro situace 1a a 1b

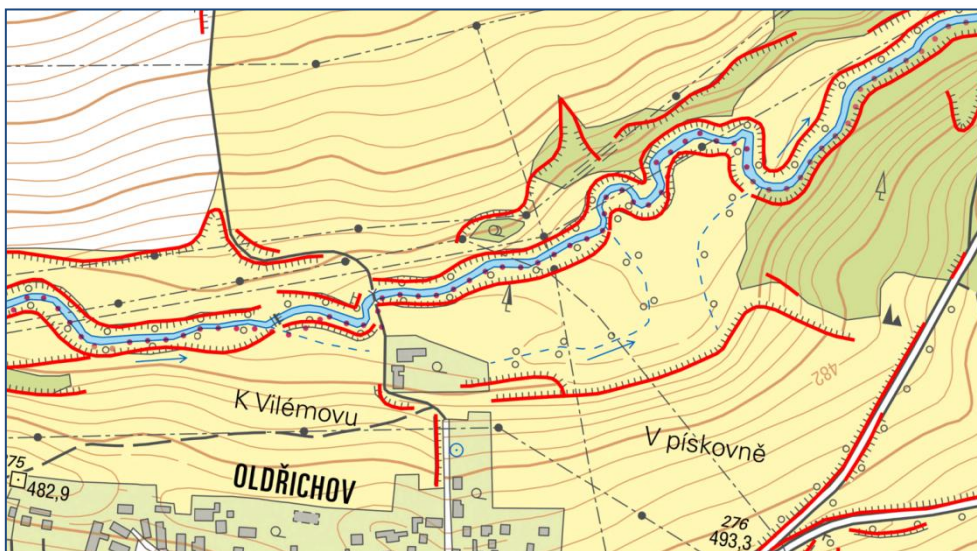
Je zřejmé, že podmínku A budou splňovat pouze prvky z obrázku 1a. Podmínka nebude splněna na obrázku 1b, protože jde o kolizi více prvků. Z hlediska implementace algoritmů obě pravidla vedou k implementaci stejného operátoru, jehož implementace je ovšem pro variantu 1b poměrně složitá z důvodu křížení kresby stromořadí s reliéfem vpravo. Implementovat pravidlo A je poměrně jednoduchá geoprostorová operace, zatímco pravidlo může obnášet několik měsíců práce.

Ověřený postup návrhu generalizace založené na pravidlech je poměrně přímočarý:

1. navrhne základní definici pravidla konfigurace a jeho řešení
2. pravidlo upřesňujeme, v případě potřeby vylepšíme algoritmus řešení
3. opakujeme bod 2 dokud:
 - a. máme dostatek kapacit
 - b. jsme spokojeni s výsledkem
 - c. je zjevné, že dotvoření výsledku operátorem bude efektivnější než náklady na další vývoj

Vlastností generalizace založené na pravidlech je jeho úspěšnost a efektivita při řešení málo komplexních situací. Čím je vyšší požadovaná komplexnost řešení, tím rostou nároky na upřesňování pravidel (rule explosion) a jejich složitost. Zároveň často roste potřeba upravovat algoritmy operátorů.

Zásadní výhodou této strategie je kromě okamžitého prvního výsledku vývoje skutečnost, že velké množství situací v kartografických dílech je nekomplexního charakteru, který však z hlediska pracnosti pro kartografa-redaktora znamená potenciál velké úspory kapacit. Přestože tedy tato strategie zdaleka nepokrývá všechny potřeby, pokrývá z velké části potřeby kapacitně náročné.



Obrázek č.2 - Běžná konfigurace liniových odsunů mimo zástavbu

2.2.2 Generalizace založená na modelování

S rozvojem generalizace založené na pravidlech započaly snahy o komplexní řešení generalizace pomocí modelování a následného řešení situací. Obě situace lze modelovat, ve shodě s myšlením kartografa stejným požadavkem (tzv. constraintem).

č.	Požadavek na kresbu mapy (constraint)	Operace
MB	Kresba prvků ve stejné barvě se nesmí překrývat	Odsuň kresbu ve vhodném pořadí algoritmem <i>Displace B</i> .

Tabuka č.2 - Situace 1a a 1b popsaná pomocí modelu constraints

Porovnání tabulek svádí k mylnému dojmu, že generalizace založená na modelování je vždy daleko vhodnější:

- definice kartografického požadavku je pouze jedna
- tato definice je naprosto shodná s popisem, který k problému uvede kartograf
- již prvotní implementace požadavku donutí implementační tým realizovat sofistikovanou verzi každého operátoru

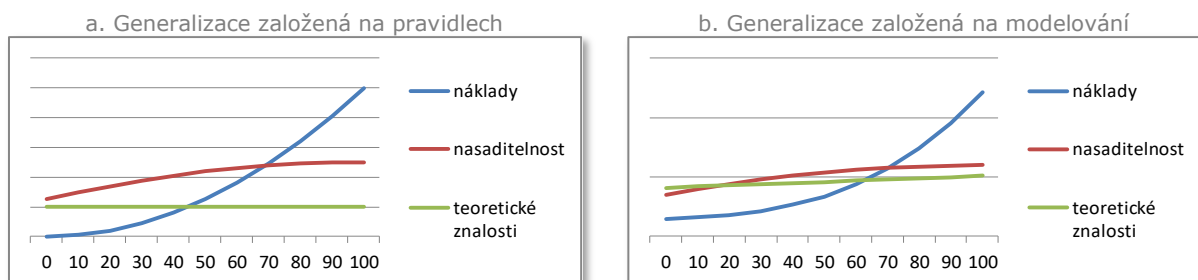
Klíčovým rozdílem je fráze *odsuň* ve sloupci operace.

č.	Pravidlo/Požadavek na kresbu mapy (constraint)	Operace
RB A	Jestliže se kresba jednoho terénního reliéfu zasahuje do kresby cesty	Odsuň kresbu terénního reliéfu jednoduchým algoritmem <i>Displace A</i> .
RB B	Jestliže se překrývají kresby více prvků mezi sebou a lze je seřadit podle sestupně podle důležitosti prvků a polohy (zde silnice, stromořadí a reliéf)	Postupně odsuň kresby prvků sofistikovaným algoritmem <i>Displace B</i> .
MB	Kresba prvků ve stejné barvě se nesmí překrývat	Odsuň kresbu ve vhodném pořadí algoritmem <i>Displace B</i> .

Zatímco implementovat výběr prvků podle pravidel RB A a RB B je poměrně přímočaré, formulace "odsuň kresbu ve vhodném pořadí" je daleko složitější. V tomto jednoduchém případě, kdy hovoříme pouze o několika (třech) prvcích v kontextu, možné varianty řešení jsou mezi sebou souměřitelné a funkce popisující splnění požadavků nemají v daném rozsahu žádné lokální extrémy (minima a maxima) je zřejmé, že i metoda nejmenších čtverců zde spolehlivě nalezne správné řešení. Postupem času byly vyvinuty další optimalizační metody (gradual descent search, simulated annealing approach, minimizing energy approach atd) k řešení tohoto problému, blíže viz kapitola XXX.

2.2.3 Generalizace založená na pravidlech nebo modelování?

Jak je patrné z následujících grafů, volba způsobu řešení není jednoznačná.



Obrázek č.3 Schematické znázornění jednotlivých přístupů

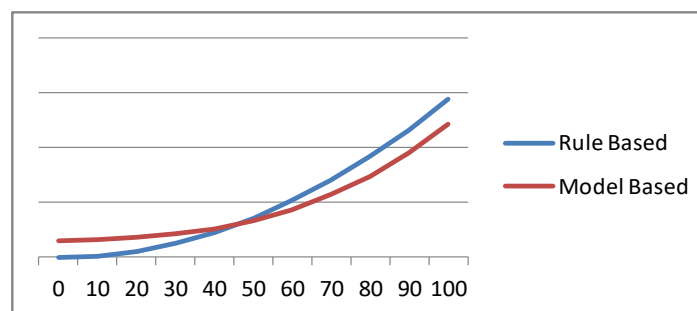
Vlastnosti generalizace založené na pravidlech

- + Nulové počáteční náklady (přináší výsledky již v raných fázích vývoje)
- + Poměrně snadné vyvážení mezi přínosy a náklady na vývoj
- + Je poměrně srozumitelná pro geoinformatiky
- + Obecně jednodušší implementace
- Při složitějších pravidlech převyšují náklady přínosy
- Není vhodná pro vysoké stupně komplexnosti kresby
- Vytváří požadavky na stanovení parametrů, které kartograf nezná (tuhost prvku)

Vlastnosti generalizace založené na modelování

- + Vhodná pro složité konfigurace kresby
- + Jednoduchý a konečný model požadavků na sestavení a kresbu mapy
- + Pouze jedna implementace operátorů
- + Doplnění modelu o požadavek nemá vliv na ostatní požadavky
- Vysoké počáteční náklady
- Vysoké nároky na teoretické znalosti řešitelského kolektivu

Porovnáme-li nejdůležitější parametr, tj. náklady na vývoj, pokud nám postačí úspora zhruba poloviny ruční práce kartografa-redaktora, je vhodná strategie řízení pomocí pravidel, potom je vhodné doplnit o strategii založenou na modelování. Snaha o dosažení vyššího pokrytí automatizovanou generalizací než 75% jsou již příliš nákladné a je vhodné se smířit se sníženou kvalitou produktu (Dutch Cadaster) nebo dokončit práce kartografem-redaktorem (SwissTopo). Ve všech strategiích platí zásada, že všechny metody by měly pracovat podle pravidla "Pokud si nejsme jisti analýzou kontextu, situaci neřešíme".



Obrázek č.4 Závinnost nákladů a úspěšnosti na komplexnosti problému

2.2.4 Technologie

Technologie v tomto období podporovaly generalizaci pouze částečně. Pouze britská firma Laser Scan (dnes 1Spatial) vyvíjela stejnojmenný software, který byl od počátku zaměřen na kartografickou produkci v celém svém komplexu. Velice nadějný software MGE Map Generalizer jako nadstavba modulu Dynamo od firmy Intergraph se ukázal jako příliš složitý v poměru k případným tržním podílům a proto byl jeho vývoj zastaven.

To souvisí se základní vlastností generalizace: jednotlivé značkové klíče a pravidla sestavení a požadavků na kresbu jsou u národních mapovacích agentur účelově sestavena podle místních požadavků. Myšlenka znovupoužití technologie vyvinuté pro mapové jedno národní dílo je proto v podstatě nesmyslná a oblast generalizace není oproti jiným oblastem vývoje pro softwarové firmy v oblasti geoinformatiky perspektivní.

Varianty řešení jsou v podstatě tři:

- Oblast generalizace GIS software vůbec neřeší, případně řeší pouze z úhlu pohledu datové generalizace, což je nejčastější případ;
- investuje se do funkcionality pro velkého odběratele s nadějí, že zakomponováním do COTS bude použitelná v dostatečné míře celosvětově. Příkladem budiž modul Generalization Toolbox v dnešním ArcGIS, umožňující využívat funkcionality vyvinutou pro tvorbu topografických map USA;
- výrobce se zaměří na sestavení systému modelování pravidel pro sestavení a kresbu mapy a implementaci optimalizačních metod. Pro jednotlivé odběratele se již na míru vytvářejí národní funkcionality. Toto je úspěšný model LaserScan, který dodává úspěšnou funkcionality pro IGN France, OS Great Britain, OS Ireland a další.

2.3 Po roce 2010

Od začátku devadesátých let obě skupiny úspěšně s většími či menšími úspěchy vyvíjely svá řešení. Po roce 2010 byly zavedeny nové metody a postupy, na jejichž základě se jeví vhodné pokusit se pro tak malé mapové dílo, jako je to v České republice oba přístupy spojit do jednoho funkčního celku.

2.3.1 Kartografické reprezentace ArcGIS a Model Builder

Zásadním přínosem pro tvorbu map v prostředí systémů GIS bylo nesporně zavedení tzv. kartografických reprezentací, představených ve verzi ArcGIS 9.2. Do té doby byla velká část sil řešitelských týmů čerpána na to, jakým způsobem správně zobrazit (rendering) GIS data tak, aby výsledek odpovídal požadavkům mapového díla. Zdánlivě nesystémovým odklonem datového modelu, kdy jsou kartografické reprezentace uloženy v databázi jako zvláštní binární pole dostupné pouze pomocí prostředků ArcGIS dokázalo snížit počet vrstev v mapových projektech natolik, že rychlost a intuitivita práce a s nimi výrazně narostla. Zároveň s tím byly jednotlivé funkcionality zpřístupněny pomocí nástroje Model Builder, což umožnilo něco do té doby nevídaného: poučený kartograf byl schopen plnohodnotně experimentovat a stát se součástí vývojového týmu bez potřeby přímé podpory programátora. Výsledky jeho práce sice nejsou přímo nasazeny v produkční lince, která neběží jako model ArcGIS, ale jako spustitelná, optimalizovaná dávka systému, nicméně slouží jako plnohodnotné zadání (proof of concept) pro programátora.

Úspěch této kombinace, na rozdíl od již zmíněného Generalization Toolboxu byl takový, že jejich nasazení při tvorbě map Nizozemského katastru a SwissTopo jsou zjednodušeně popisovány větou "zvládli generalizaci přímo pomocí ArcGIS".

2.3.2 Agilní vývoj a scrum

Dalším, tentokrát organizačním, přínosem pro úspěšný vývoj v oblasti automatizované generalizace se ukázal agilní způsob vývoje. Tento způsob vývoje je zcela odlišný od běžného modelu, kdy jsou odhadnuty jednotlivé milníky, doba trvání na vývoj a vyčleněné kapacity. Zvláště v oblasti automatizované generalizace, kde v případě potřeby není možné do týmu podle potřeby začleňovat nové pracovníky (protože jednoduše počet kvalifikovaných pracovníků na národním trhu je konečný) to vede k nedodržení termínů a nespokojenost řídicích pracovníků s rychlostí vývoje.

Agilní metodiky vývoje vycházejí z předpokladu, že máme stanoven vývojový tým, který jsme schopni doplňovat pouze v minimální míře. Ať už proto, že jsem do týmu zahrnuli všechny v oboru kdo do problematiky má co říci, nebo proto že jsme součástí státní správy a máme pevně daný rozpočet. Délka trvání vývoje je fixní, zákazník tedy zná termín kdy by chtěl výsledek vývoje nasadit do provozu. Představu o tom, co chceme vyvinout máme pouze orientační, chceme vyvinout něco co má co nejvyšší přidanou hodnotu. V případě generalizace chceme vyvinout něco, co maximálně ušetří lidské kapacity.

Cílem klasického vývoje je vyvinout předem danou
funkcionalitu danou odhadem detailně specifikovaných potřeb.

vs.

Cílem agilního vývoje je maximálně využít schopnosti vývojového týmu
ve prospěch zákazníka a splnění obecně definovaných vizí a krátkodobých cílů.



Obrázek č.5 Rozdíl tradičního a agilního pojetí vývoje software, převzato z [6]

Nizozemský katastr i SwissTopo s úspěchem využívají metodu Scrum. Tato metoda je, velice zjednodušeně řečeno, postavena na jednoměsíčních iteracích (sprintech), uzavřených prezentací a schválením výsledků zákazníkem (stake holder), včetně návrhu, upřesnění a schválením plánu na další sprint v rámci každoměsíčních project reviews (opravdu jde o přehodnocení, nejen doklad o stavu). To vše zaznamená vedoucí vývojového týmu (scrum master) do zjednodušené evidence (back log), který udržuje aktuální v průběhu sprintu.

Nejvýraznějším přínosem je vývoj v malých iteracích, který dává zákazníkovi jistotu, že se vývoj ubírá správným směrem a může ho průběžně usměrňovat v závislosti na poznání technických a personálních možností v daný okamžik. Přínosem pro řešitelský kolektiv je minimalizace formalizační práce, kterou zabezpečuje scrum master v podobě backlogu a stmelení se zákazníkem. To vede k výrazné motivaci celého kolektivu, protože každý člen týmu prezentuje své výsledky sám za sebe bez prostředníka.

2.3.3 Schopnost postavit široký tým

Posledním významným přínosem v oblasti automatizované generalizace v poslední dekádě bylo úspěšné využití výzkumných kapacit, výrobců GIS software a zejména akademické sféry v jednom celku. Je zřejmé, že požadavky zákazníka, kde typicky 80% vývojových kapacit je vynaloženo pro funkcionality specifickou pro národní mapové dílo a specifický datový model nejsou dobrý obchodní model pro vývoj software GIS. Obdobně, akademická sféra je zaměřená na prvotní výzkum až do fáze ověření a popisu nové myšlenky (proof of concept), což se uživateli samozřejmě jeví poněkud nepraktické.

Jako vhodná cesta se ukázalo umožnit členům výzkumného týmu doktorantské studium na vybraných univerzitách (např. university of Delft, TU, které se problematikou zabývají). Výsledným efektem bylo zásadní zvýšení teoretických znalostí výzkumného týmu. Obdobně se jako užitečná ukázala přímá spolupráce s místním poskytovatelem GIS software na funkcionalitách s přesahem nad národní potřeby.

3. Procesní model řešení generalizace

Při řešení se generalizace jeví jako samostatný proces tvorby mapového díla. Stávající pracovní postup je logicky doplněn generalizačním modulem, který je předřazen ručnímu zpracování kartografem¹. Předpokladem efektivnosti je samozřejmě snížení časových nároků na práci kartografa, zejména minimalizací rutinních, často se opakujících úprav.



Obrázek č.6 Místo automatizované generalizace v procesu zpracování mapy

Vlastní proces generalizace je potřeba chápat jako samostatně spustitelný proces, který běží mimo grafické prostředí GIS software kartografické produkční linky. Proces je optimálně umístěn na aplikačním serveru, kde je možné v případě potřeby rozložit zátěž na více strojů. Spouštění a monitoring procesu je vhodné realizovat v závislosti na technických schopnostech kartografů pomocí vzdáleného přístupu k serveru, případně pomocí jednoduchého webového rozhraní.

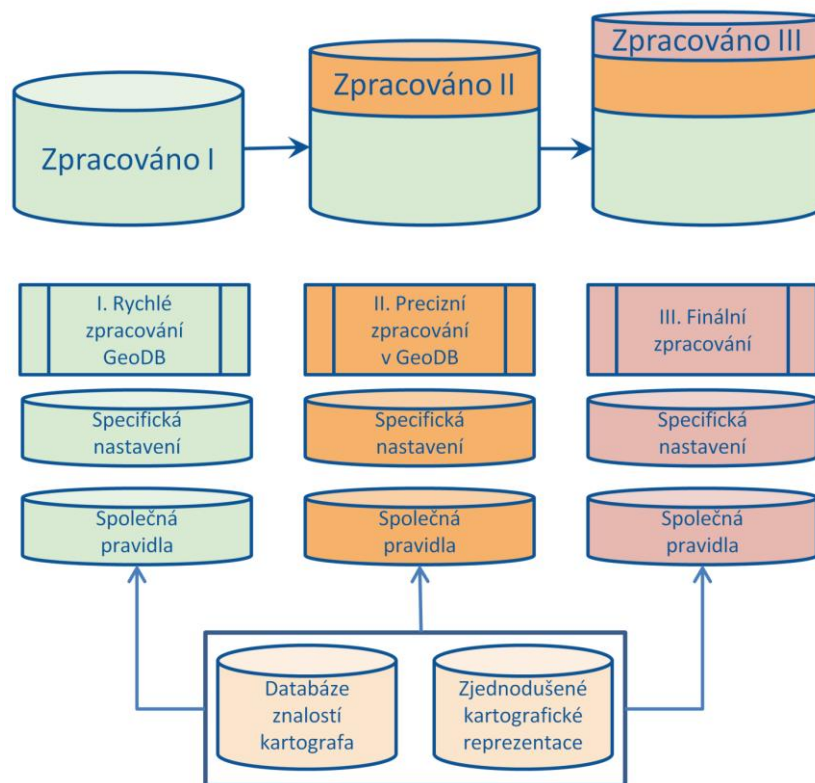
Při procesu generalizace je potřeba vytvářet a udržovat velké množství pracovních informací, které nemají přímou souvislost s kartografickou produkční linkou a jejím datovým modelem. Generalizační operace by navíc mohly být v konfliktu s kontrolami konzistence dat na databázovém stroji. Z tohoto důvodu je vhodné data procesu generalizace oddělit vytvořením pracovní (pomocné) databáze, ve které bude zpracování probíhat. Je zřejmé, že v pracovní databázi je nutné udržovat informace pro začlenění výsledků procesu zpět do produkční databáze.



Obrázek č.7 Životní cyklus dat při procesu automatizované generalizace

¹ redaktor, sestavitel

Proces automatizované generalizace je vhodné rozdělit do samostatných modulů, které pracují nad společnou pomocnou databází. Každý modul databázi zpracuje a vyřeší část generalizačních úloh. Prvky modulem dotčené jsou označeny v příslušném atributu tak, aby mohly být vyřazeny z dalšího zpracování. V rámci projektu TB04CUZK001 byly vypracovány čtyři generalizační moduly, v dalších fázích životního cyklu je samozřejmě možné moduly doplňovat.



3.1 Generalizační modul M1

Generalizační modul M1 je navržen jako specifický modul pro řešení generalizace metodou odsunu prvků v izolovaných a méně komplexních generalizačních situacích. Pro realizaci je navržen a vyvinut komplexní operátor odsunu.

Modul M1 využívá generalizačního přístupu, ve kterém se uplatňují prvky Rule Based Generalization a Constraint Based Modelling, přičemž jednotlivá omezení a podmínky (constraints) jsou omezeny na množinu tzv. hard constraints a vyhodnocení jejich porušení. U tohoto typu constraint je vždy možné jednoznačně rozhodnout, zda v daném kontextu bylo splněno či nikoli. Modul M1 při vyhodnocení akceptuje jen taková řešení, kdy definovaná constraints nejsou v řešeném kontextu porušena.

Cílem řešení je eliminovat konflikty zejména liniových objektů v kartografickém modelu, který bude použit k sestavení mapy. Objekty jsou v kartografickém modelu symbolizovány signaturami (mapovými značkami, mapovými znaky). Signatury bodových, liniových i plošných prvků zařazených do kartografického modelu vytvářejí obraz mapy. V kartografickém modelu lze jednotlivé prvky nahradit stopou (tzv. footprintem) vymezenou jejich signaturou. Tento footprint má pro bodové, liniové i plošné prvky vždy areálový charakter, který lze vyjádřit objektem geometrického typu polygon. Pro účely identifikace konfliktů objektů v kartografickém modelu je každému prvku přiřazen footprint. Ten může věrně kopírovat stopu signatury nebo může být do určité míry zjednodušen. To platí zejména u složených signatur, kdy např. liniový prvek je symbolizován linií určité tloušťky v kombinaci s bodovými signaturami pravidelně rozmístěnými na definiční linii prvku. V takovém případě je možné nahradit přesný footprint zjednodušeným footprintem tvořeným bufferem kolem definiční linie prvku o šíři

velikosti bodové signatury. Rovněž pro signatury bodových prvků může být footprint geometricky zjednodušen (např. kruhový oblouk bude nahrazen lomenou čarou), zejména z výpočetních důvodů. Míru zjednodušení footprintu je vhodné volit tak, aby zbytečně nedocházelo k identifikaci falešných konfliktů nebo naopak aby nezůstaly některé skutečné konflikty opomenuty.

Modul M1 pracuje s obecně definovanými entitami a vztahy mezi nimi. Proces generalizace je řízen parametry, které jsou definovány konfiguračně v XML souboru. Znamená to, že tentýž modul lze použít pro generalizaci libovolného kartografického modelu, který je popsán v sémantice generalizačního modulu M1. Základní entitou tohoto modelu je *element*. Jedná se o prvek kartografického modelu jednoznačně definovaných charakteristik a chování. Jednotlivé elementy jsou výskytem entity *element typ*, přičemž každý element typ má v rámci celého kartografického modelu jedinečné charakteristiky, chování a vztahy k ostatním entitám. Každý element typ je popsán jednoznačným identifikátorem, geometrickým typem, vahou (důležitostí), barvou (v případě areálových prvků barvou obrysové linie), původem (např. v případě vstupního modelu v Esri Geodatabase organizací prvku v rámci struktury Feature Data Set a Feature Class), definicí footprintu a příznakem, zda má topologické vztahy vůči dalším entitám. Popis element typu může být doplněn dalšími vlastnostmi, např. příslušností k sémantickým kategoriím kartografického modelu. Chování element typu je popsáno výčtem operátorů, které lze na element této entity aplikovat. Model je doplněn popisem vztahů mezi jednotlivými entitami, typicky mezi elementy téhož element typu, mezi elementy dvojice element typů či mezi elementy skupin element typů navzájem. Jde především o vyjádření, zda jsou relačně svázané entity při prostorovém překryvu signatur v konfliktu, případně jaká je požadovaná světlost (rozestup, mezera, clearance) mezi signaturami entit. Obecné (generické) vztahy mezi entitami jsou přepisovány specifickými vztahy popsanými v konfiguraci modelu. Příkladem může být definice specifického footprintu prvku terénní stupeň v případě řešení konfliktu s prvkem stromořadí, kdy bodová signatura na linii stromořadí může zasahovat do spádových čárek signatury terénního stupně, ale nikoli do jeho koruny.

Modul M1 využívá prostorového uložení dat v DB PostgreSQL s prostorovým rozšířením PostGIS. Vstupem jsou data výchozího topografického modelu transformovaného do modelu elementů kartografického modelu. V něm jsou elementy reprezentující bodové, liniové a plošné prvky uloženy společně bez ohledu na jejich geometrický typ. Tento způsob uložení umožňuje každému elementu přiřadit a uchovat nejen vstupní geometrii převzatou z topografického modelu, ale rovněž footprint signatury a cílovou geometrii získanou generalizačním procesem. V pracovní části databáze vymezené modulu M1 je vytvořen datový model dle kartografického modelu popsaného v konfiguraci a z výchozího úložiště jsou převzaty všechny elementy popsané v konfiguraci modelu. Do datového modelu jsou zapsány charakteristiky všech definovaných entit, jejich chování a vzájemných vztahů včetně topologických. Pro elementy zařazené do topologie je vytvořen topologický model, v němž jsou jednotlivé elementy vyjádřeny jejich topogeometriemi složenými ze základních konstrukčních entit node, edge a face. Do pracovního datového modelu pro generalizaci jsou zahrnuty topologické i netopologické elementy. Pro všechny elementy pracovního datového modelu jsou vypočteny jejich footprinty, přičemž jeden element může mít definováno v závislosti na konfiguraci modelu jeden či více footprintů.

Proces vlastní generalizace probíhá tak, že jsou postupně procházeny elementy seřazené sestupně podle váhy, která je jednotlivým element typům přiřazena. Pro každý řídicí element (vodící element, master element) je zjištěno výchozí kolizní místo. V konfliktní zóně jsou setříděny všechny elementy dle vzdálenosti od řídicího elementu. Postupně jsou řešeny konflikty řídicího elementu a nejbližšího konfliktního elementu (závislého elementu, slave elementu), přičemž konflikty elementů jsou detekovány vždy vůči aktuální geometrii elementů. Zároveň je zaručeno zachování pořadí závislých konfliktních elementů vůči řídicímu elementu v konfliktní zóně v průběhu řešení konfliktu. Rekurzivně jsou procházeny vzdálenější konfliktní elementy dle principu, že pro následnou iteraci je

původní závislý element zafixován, stává se řídicím elementem a přebírá váhu iniciačního řídicího elementu. Po každém elementárním odsunu je provedeno vyhodnocení, zda se podařilo konflikt vyřešit v souladu s definovanými podmínkami a omezeními (constraints). V případě nesouladu jsou neakceptované úpravy vráceny do předchozího stavu, což je umožněno transakčním způsobem řešení. Po vyřešení všech konfliktů výchozího řídicího elementu jsou iterativně řešeny konflikty elementů všech element typů s výjimkou nekonfliktních. Po projití všech elementů je provedena rekonstrukce topologických elementů a provedené změny jsou promítnuty do cílové geometrie modelu elementů kartografického modelu včetně naplnění stavových atributů.

Při řešení konfliktu elementů odsunem je na základě identifikace konfliktní zóny vypočtena distanční linie, která představuje v konfliktní zóně průběh geometrie závislého elementu, která nezpůsobí konflikt signatur řídicího a závislého elementu. Dále jsou vypočteny vektory posunu geometrie závislého elementu na distanční linii a provedena transformace geometrického průběhu závislého elementu.

Ve specifických vztazích element typů je podporován princip ztotožnění signatur řídicího a závislého elementu v celém jejich průběhu nebo jejich části. Z hlediska vzájemné polohy obou elementů jsou podporovány následující typy ztotožnění signatur: ztotožnění průběhu linie (distanční linie inciduje s geometrií řídicího elementu), ztotožnění hrany průběhu kresby (distanční linie inciduje s obrysovou linií signatury řídicího elementu), ztotožnění průběhu linie na dotyk (distanční linie ve vzdálenosti aritmetického průměru šířek signatur obou elementů od geometrie řídicího elementu) a ztotožnění průběhu linie s rozestupem (distanční linie ve vzdálenosti aritmetického průměru šířek signatur obou elementů od geometrie řídicího elementu zvětšené o hodnotu clearance).

Generalizační modul M1 je vhodný pro řešení konfliktů signatur elementů odsunem v izolovaných a méně komplexních generalizačních situacích. Užití vyvinutého operátoru odsunu není vázáno na předem definovaný kontext generalizační situace.

3.2 Generalizační modul M2

Generalizační modul M2 je navržen jako obecný rozšiřitelný modul pro řešení kartografické generalizace. Rozšiřitelný je ve smyslu možnosti přidávání pravidel, parametrů a algoritmů pro zpracování geometrie kartografických objektů. Jako obecný je označován vzhledem k ambici řešit všechny typy konfliktů kartografické reprezentace a možnosti definovat zpracování pro různé typy a měřítka map.

Modul je postaven na několika základních principech:

1. Znalostně omezený "útok hrubou silou". "Útok hrubou silou" je chápáno generování kombinace přípustných stavů geometrické a klasifikační charakteristiky kartografických objektů získané aplikací generalizačních algoritmů. Tato kombinace je testována na absenci konfliktu (s tím, že je uchováván stav s nejmenší mírou konfliktu) a při prvním takovém stavu ukončena. Znalostní omezení tohoto přístupu je dáno jednak určením přípustných operací nad kartografickými objekty jejich příslušností k třídě, jejím podskupinám a jejich lokální prostorovou konfigurací. Dalším omezením je uspořádání přípustných operací podle míry jejich dopadu a úspěšnosti v dosavadním běhu modulu.
2. Lokální omezení konfliktu k zamezení dominového efektu. Objekty v konfliktu jsou rozděleny do shluků určených aktuálním konfliktem a potenciálním konfliktem vzniklým manipulací dotčených objektů generalizačními operátory. Tento shluk je ovšem omezen prostorovým dosahem hlavního konfliktu, který je určen arbitrárně s ohledem na měřítka.
3. Topometrická měření pro identifikaci a konstrukci pravidel. V rámci prostorového rozsahu konfliktního shluku je provedena soustava topologických a metrických měření, která slouží ke kontrole topologických vazeb a stanovení odvozených pravidel.
4. Kódování znalostí do explicitně vyjádřené, perzistentní soustavy záznamů. Jádrem těchto záznamů jsou následující tabulky:
 - a. **maps** - popis obsahu mapy zahrnující identifikaci tříd a základní parametry jejich symbolizace. Tato tabulka slouží k zjištění parametrů symbolizace objektů a k odvození generických omezení.
 - b. **constraints** - podmínky čitelnosti mapy. Generické podmínky jsou odvozeny z parametrů tabulky maps. Další specifická omezení mohou být doplněna uživatelem.

- c. **rules** - seznam přípustných generalizačních funkcí pro kartografický objekt (skupinu, třídu) v určité situaci. Pravidla mohou být uživatelská nebo odvozená z běhu modulu.
- d. **contexts** - vztah kartografického objektu k jeho okolí a specifikace jeho vlastností vztahující se k pravidlu. Specifikace kontextů je součástí definice pravidel
- e. **structures** - prostorové konfigurace kartografických objektů ovlivňující pravidla, která jdou nad rámec popisu obsahu mapy
- f. **thesaurus** - popis agregovaných identifikací skupin kartografických tříd umožňující definici obecnějších pravidel.

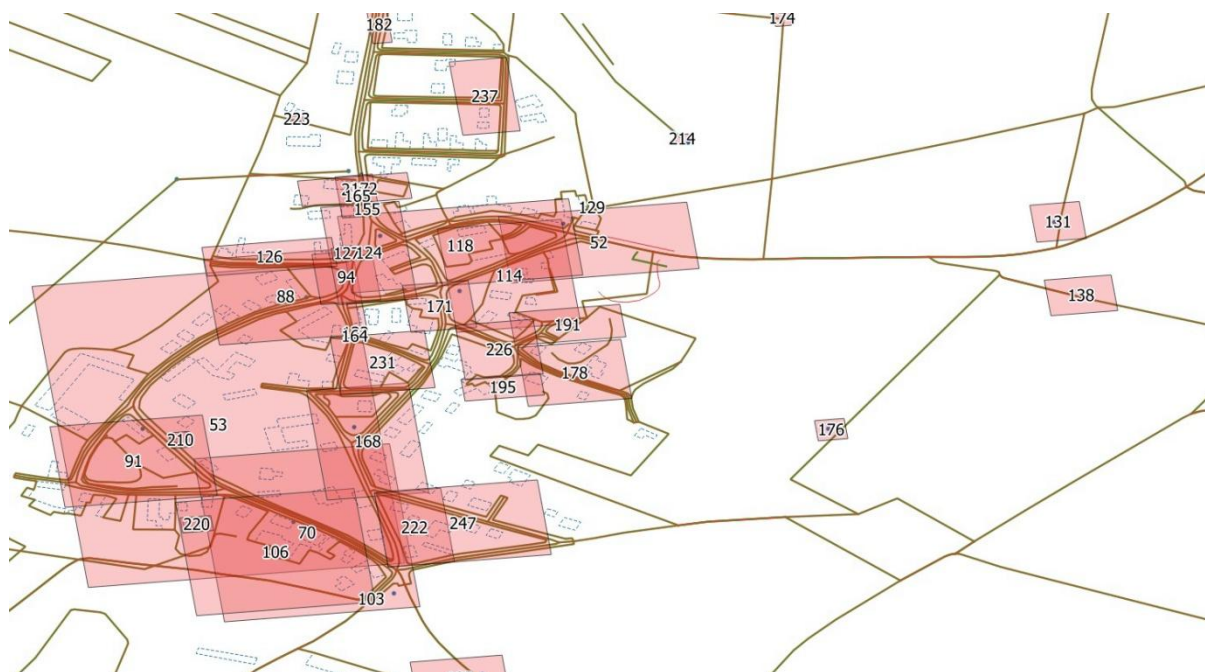
Mimo tyto základní znalosti je udržován seznam dostupných algoritmů a záznam jejich úspěšnosti při řešení konfliktů.

5. Zatřídění a uspořádání konfliktů do 3 skupin:
 - a. **imperceptibilita** - nemožnost korektního zobrazení. Vedle celkové nepatrnosti objektu zahrnuje i problémy s vyjádřením průběhu tvaru. Je řešena jako první pomocí odstranění, kolapsu, zjednodušení a škálování.
 - b. **komplikace** - interakce skupiny objektů, kde existuje jednoznačně definovaný postup úpravy v rámci dané mapy. Vedle explicitně definovaného chování objektů zahrnuje i problémy nahloučení, souběhu a gestaltu. Řešena je vyžádanou úpravou průběhu, typifikací, agregací a vyčištěním
 - c. **kolize** - symbolika kartografických objektů se překrývá. Jedná se o závěrečný běh řešený pomocí odsunů, odstranění, odřezání a deformace.
6. Postupné řešení od nejjednoduššího konfliktu. Modul se snaží primárně vyřešit s nejmenší četností. Předpokládá nenalezení bezkonfliktního stavu a pamatuje si nejlepší dosažený stav. Na výstupu je možné stanovit threshold pro akceptaci řešení. V modulu je také možné nastavit časový zámek pro dobu řešení konfliktního shluku.
7. Modul předpokládá běh ve dvou režimech:
 - a. striktní - běh je založen na uživatelských pravidlech
 - b. adaptivní - modul za běhu definuje nová pravidla na základě úspěšných řešení, která postupně nahrazují uživatelská pravidla na bázi podobnosti kontextů nově definovaných pravidel.

V rámci certifikace se předpokládá testování striktního režimu, adaptivní režim vyžaduje k efektivnímu použití rozsáhlejší testování a stabilní implementaci CBR rezolučního stroje.

V rámci výše zmíněného konceptu probíhá zpracování mapy následujícím způsobem:

1. vyhledání a řešení imperceptibility
2. vyhledání struktur zmíněných v pravidlech pro komplikace
3. řešení komplikací a případné sdružení vyřešených objektů
4. identifikace kolizních shluků a jejich uspořádání
5. topometrická měření
6. sestavení plánu řešení kolizních shluků (nalezení pravidla, algoritmu a parametrického rozsahu)
7. postupná aplikace na algoritmů na objekty zohledňující minimalizaci změny
8. kontrola omezení po každé jednotlivé změně
9. zápis výsledku, odvozeného pravidla, statistiky průběhu zpracování



Obr. 1 Zóny konfliktních shluků identifikovaných kolizí

3.3 Generalizační modul M3

Generalizační modul M3 je navržen jako obecný modul pro modelování a řešení generalizace.

Požadavky na kartografické zpracování jsou reprezentovány využitím multi-agentního systému pomocí agentů, který není pro tento účel chápán jako striktně hierarchický (meso agents, micro agents, element agents).

Agent může být vázán na různé entity, například:

č.	Entita	Agent
1.	Grafický element	Prvek trigonometrický bod musí být zobrazen beze změny polohy
2.	Grafický element	Průběh silnice musí být zakreslen s přesností 0,8 mm v měřítku mapy
3.	Část elementu	Při souběhu dvou liniových kreseb (jednočarý element, lemovka výplně, lemovka dvoučaré kresby) v délce více než 2 cm, oscilaci méně než 0,3 mm a vzdálenosti menší než 5 mm v měřítku mapy musíme zabezpečit souběh průběhu.
4.	Část elementu	Kresba rohů budovy by měla být patrná a bodové značky by je neměly překrývat.
5.	Skupina elementů	Při generalizaci budov v rámci vnitrobloku musí být zachován jeho charakter. Tato charakteristika je modelována jako skupina agentů: - elementy patřící do vnitrobloku - rozeznané strukturální vzory
6.	Konfigurace objektů	Úhel napojení komunikací v křižovatkách by měl být zachován.

Tabulka č.3 Příklad agentů a jejich vazby na entitu

Základní vlastností každého agenta je schopnost stanovit it hodnotu splnění kritéria, které reprezentuje. Rozdíl hodnot před provedením nějakého kartografické úpravy (aplikováním algoritmu operátoru generalizace) a po něm reprezentuje úspěšnost této úpravy.

Každá z těchto operací může mít vliv na splnění jiných požadavků na stejný element či na elementy okolní. Za účelem komplexního stanovení úspěšnosti nějaké úpravy hodnotu splnění kritéria převádíme do normalizovaného tvaru tak, aby je bylo možno mezi sebou porovnávat v intervalu [0, 1]. Hodnota 0 označuje stav kdy je požadavek splněn, hodnota 1 první akceptovatelnou hodnotu. Průběh mezi nimi musí být patřičným způsobem linearizován.

Pro průběh kresby silnice v předchozí tabulce v mapě s požadovanou přesností zákresu 0,2 mm v měřítku mapy tedy platí:

2. Přesnost průběhu kresby silnice	Vyhodnocení	Normalizováno
0,1 mm	Spokojen	0
0,2 mm	Spokojen	0
0,5 mm	Akceptovatelné	0,5
0,8 mm	Nespokojen	1
1,0 mm	Nespokojen	1

Tabulka č.4 Příklad vyčíslení požadavku reprezentovaného agentem

Druhou vlastností agenta je seznam algoritmů generalizačních operátorů (odsun, vypuštění apod.), které potenciálně mohou směřovat ke zlepšení splnění kritéria, které agent reprezentuje. Seznam je setříděný tak, aby preferované metody řešení byly první. Vypuštění elementu tedy bývá až na konci seznamu, pokud žádné řešení nevede k akceptovatelnému zlepšení.

Seznam povolených algoritmů je patrný z následující tabulky:

č.	Agent	Povolené algoritmy
1.	Prvek trigonometrický bod musí být zobrazen beze změny polohy	Vrácení do skutečné polohy
2.	Průběh silnice musí být zakreslen s přesností 0,8 mm v měřítku mapy	Vrácení do skutečné polohy Vypuštění elementu
3.	Při souběhu dvou liniových kreseb (jednočarý element, lemovka výplně, lemovka dvoučaré kresby) v délce více než 2 cm, oscilaci méně než 0,3 mm a vzdálenosti menší než 5 mm v měřítku mapy musíme zabezpečit souběh průběhu.	Ztotožnění průběhu linií (paralelizace) Vypuštění elementu
4.	Kresba rohů budovy by měla být patrná a bodové značky by je neměly překrývat.	Vypuštění budovy
5.	Při generalizaci budov v rámci vnitrobloku musí být zachován jeho charakter. Tato charakteristika je modelována jako skupina agentů: - elementy patřící do vnitrobloku - rozeznané strukturální vzory	Viz agenti ve skupině
6.	Úhel napojení komunikací v křižovatkách by měl být zachován.	Modifikace vrcholů do požadovaného rozsahu Vypuštění elementu

Tabulka č.5 Příklad vazby operátorů generalizace na agenty

Řešení požadavků na kartografickou kresbu je možné mnoha různými způsoby. Navržený model je otevřený jak multiagentnímu řešení, metodě nejmenších čtverců, simulated annealing i gradient descent search podle potřeby, kontextu a počtu prvků.

Pro mapy ZM 10 se ukázal jako vhodný následující přístup:

1. Normalizujeme databázi do modelu grafika+symbol
2. Postupně umísťujeme jednotlivé elementy kresby podle priorit
3. Při umísťování elementů vytvoříme model agentů vztažený k tomuto elementu, včetně případných strukturálních vzorů
4. případnou nespokojenost s vyhodnocením řešíme postupně využitím dostupných algoritmů v plném rozsahu
5. při aplikaci operátorů okamžitě reagujeme na zhoršení ostatních agentů v kontextu
6. pokud žádná z dostupných metod nevedla k akceptovatelnému řešení, provedeme totéž iterativně s krokem 5% optimálních hodnot

Je zřejmé, že tento postup lze v dalším projektu optimalizovat, například zabránit opakovanému počítání jednotlivých hodnot a strukturálních vzorů jejich podržením v paměti či ukládáním do pomocné databáze, volit další umístěný prvek co nejbližší stávajícímu apod. Je také patrné, že uvedený postup bude vyhovovat kontextům v určité složitosti konfigurace. Jelikož postup kopíruje základní postup kartografa při ručním sestavení mapy a generalizaci je pravděpodobné, že jejich množství bude veliké. V ostatních případech je vhodné doplnit vytvořený model dalšími, účelovými strategiemi

řešení. Zatímco výčet algoritmů řešení a normalizace hodnot "spokojenosti" je konečný je pravděpodobné, že při aplikaci v menších měřítcích ZM bude potřeba doplňovat algoritmy strukturálních vzorů a stanovovat parametry pro vyčíslení. Strategie řešení je navržena tak, že aplikuje operaci pouze tak, aby nedošlo ke zhoršení jiného kritéria.

Literatura

- [1] ICA Commission on Generalisation and Multiple Representation
<http://generalisation.icaci.org/>
- [2] State of the Art of Automated Generalization in Commercial Software
http://www.eurocdr.net/sites/default/files/uploaded_files/58.pdf
- [3] Konzultace řešení automatické generalizace Národní mapovací agenturou Švýcarska (SwissTopo) a místního zastoupení firmy ESRI
Zpráva z pracovní cesty, Augustýn, 2015
- [4] ICA/EuroSDR NMA workshop, Nizozemí, Amsterdam
Zpráva z pracovní cesty, Augustýn, 2015
- [5] ESRI Swiss, Workshop Generalizace
Zpráva z pracovní cesty, Jindrák, 2015
- [6] Agilní metodiky vývoje software, Diplomová práce, Bc. Tomáš Hajdin, 2005
http://is.muni.cz/th/39440/fi_m/dp.pdf
- [7] Modelling the Overall Process of generalization
W. Mackaness, A. Ruas and T. Sarjakoski 2007
- [8] Automated Processing for Map Generalization with Modular Operator Services
Moritz Neun, Dirk Burghardt, Robert Weibel, Geoinformatica 2009
- [9] Fully automated generalization of a 1:50k map from 1:10k data,
Stoter, Post, Altena, Nijhuisb, Brunsb 2013
- [10] Modelling the Overall Process of Generalisation, L. Harrie and R. Weibel. in A. Ruas, W.A.Mackaness and T. Kilpeläinen (Eds.), Generalisation of Geographic Information: Cartographic Modelling and Applications. Amsterdam: Elsevier Science, 2007, pp 67-87.
- [11] Workflow Management and Generalisation Services, I. Petzold, D. Burghardt and M. Bobzien.in 9th ICA Workshop on Generalization and Multiple Representation, Portland, 2006

Přílohy

1. Optimalizace rychlosti procesů a algoritmů
2. Ověření výkonu prostorového indexu
3. Měření kvality kresby

Optimalizace rychlosti procesů a algoritmů

Oblast vývoje nástrojů pro automatickou generalizaci je velice složitá, protože se jedná o vývoj v mnohém základní, u kterého je však požadováno jeho provozní nasazení.

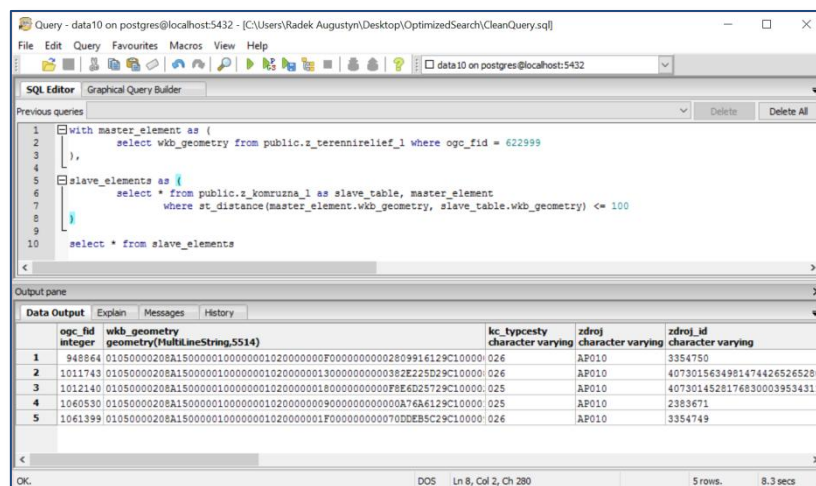
Rychlost dosažených algoritmů (jejich výkon) je negativně ovlivněna rozporupnými požadavky na vývoj:

- Požadavkem uživatele je rychlý vývoj zaměřený na reálné výstupy nasaditelné do provozu, přičemž optimální vývoji se zaměřuje na kvalitu a systémovost;
- požadavky na vývojáře zahrnují nejen výtečné znalosti kartografie a geoinformatiky samotné, ale i dalších podoborů počítačové vědy;
- vývojový cyklus klíčových funkcionalit generalizace je delší než vývojový cyklus používaného software. Vyvinuté metody a algoritmy nemusí být vždy optimální z hlediska použitých frameworků.

Uživatel logicky očekává, že jeho vývojový tým je natolik kvalitní, že kromě vývoje funkcionality zvládne zabezpečit její kvalitu "tak nějak samozřejmě". Problém, který vznikne si demonstrováme na zjednodušeném příkladu vyhledání kandidátů na kolize a slícování pro prvek terénního reliéfu.

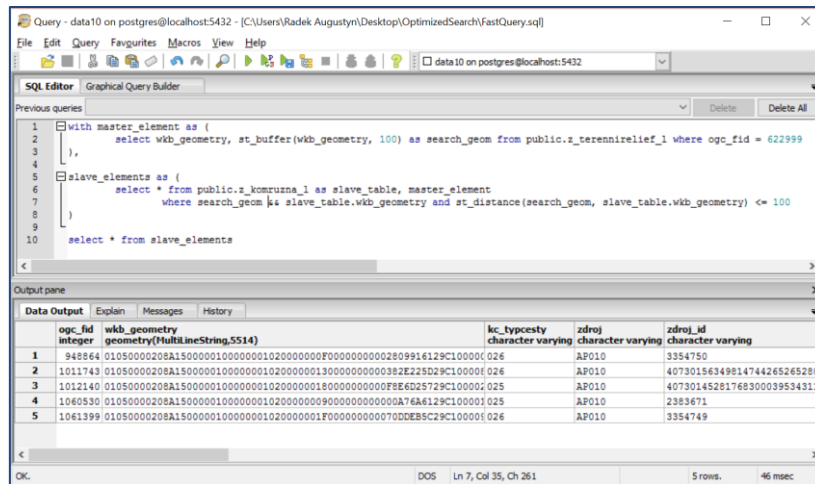
Mějme liniový prvek (hlavní, *master_element*) terénního reliéfu (*z_terennirelief_1*), symbolizovaný značkou 6060100. Prvky v této značce mohou mít kartografický vztah s prvky ve vrstvě komunikace (*z_komruzna_1*) do vzdálenosti 100 m. Při jeho umístění tedy hledáme jako kandidáty pro sestavení vazeb všechny prvky ve vrstvě komunikace, které jsou blíž k hlavnímu elementu než 100 m.

Poměrně snadno sestavíme a vyladíme prostorový dotaz, který námi zadanou úlohu splní. Dotaz je jednoduchý, snadno čitelný a plně funkční. Klíčovou podmínkou je fragment *st_distance(master_element.wkb_geometry, slave_table.wkb_geometry) <= 100*. Vyhledání prvků trvá na rychlém hardware 8.3 s a je zobrazeno na následujícím obrázku.



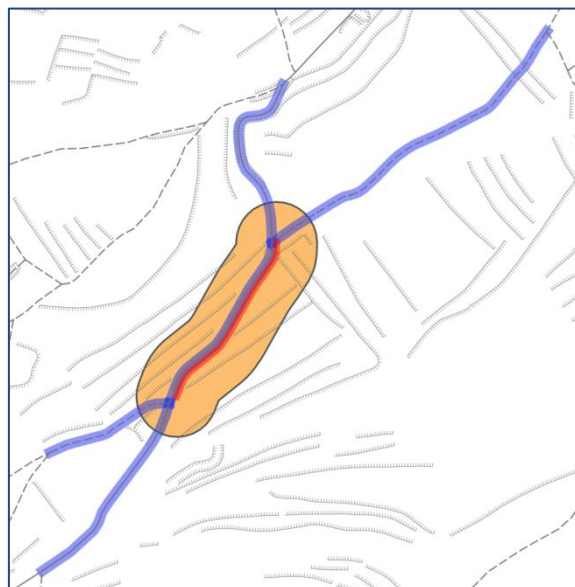
Obrázek č.8 Přehledný a pomalý dotaz

Pokud po návrhu a odladění funkcionality dostane stejný dotaz specialista na příslušnou databázi, upraví ho poměrně jednoduchým způsobem, kdy před poměrně složitý výpočet vzdálenosti mezi jednotlivými prvky nejprve předradí výběr prvků, jejichž obálka (Minimal Bounding Rectangle) protíná obálku stometrové zóny okolo hlavního elementu. To ve výsledku způsobí zení který způsobí jeho dvoustenásobné zrychlení na 46 ms.



Obrázek č.9 Optimalizovaný a rychlý dotaz

Dotaz je zjednodušeně znázorněn na následujícím obrázku, kdy hlavní prvek je zobrazen červeně, stometrová zóna 100 okolo něho je v oranžové barvě a vybrané prvky podbarveny modrou lemovkou.



Obrázek č.10 Prvek zájmu a k němu vyhledané komunikace v okolí 100 m

Je jasné, že v jiném prostředí či v nové verzi může databáze tuto úlohu optimalizovat "tak nějak sama, ovšem potřeba průběžně monitorovat výkon (profiling) vyvinutého kódu a optimalizovat ho na rychlost zůstává. Očekávat od celého vývojového týmu znalosti takovýchto, z pohledu specialisty na optimalizaci databáze "triviálních" postupů je nereálné. To ilustruje důležitost optimalizace výsledných algoritmů před jejich nasazením, případně i v jejich průběhu. I přes dramatické navyšování výkonu počítačů totiž zůstává tzv. "upočitatelnost" vyvinutých řešení a dosažení akceptovatelného výsledku v reálném čase jedním z problémů automatické generalizace.

Osvědčeným řešením tohoto problému je začlenění optimalizace jako samostatného vývojového kroku, odděleného od vývoje funkcionality a jeho nasaditelnosti. To je podrobně popsáno v kapitole "Sestavení vývojového týmu, plánování a řízení vývoje".

Související literatura

- [1] Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, 2008

Ověření výkonu prostorového indexu

V rámci ověřování metod a postupů proběhlo ověření předpokladu, že využitím geoprostorové databáze a zejména její indexace eliminuje potřebu řešit prostorové dotazy nad malým prostorem.

Test byl vytvořen nad daty Data10 pro tři náhodně liniové vybrané prvky terénního reliéfu takto:

```
Pro      náhodně      vybrané      elementy      terénního      reliéfu:
      Zvol výřez okolo středu o rozměru 500 m, 1000 m, 5000 m, 10 km, 20 km, 30 km, 50 km:
      Pro každý výřez vytvoř novou tabulku s obsahem hledaných vrstev
      (z_komruzna_l,      z_voda_l,      z_komsilnice_l      a      z_komsilnice_l):
      Zjistí dotazem všechny hledané elementy bližší než 100 m od vybraného
      Zjistí optimalizovaným dotazem všechny hledané elementy bližší než
      100 m od vybraného
```

Výstupy testů

```
C:\Python27\python.exe ./TB04CUZK001_CartoModel\dataconnectors/windowmovetest.py
```

```
Module dataconnectors initialized.
```

```
Spatial window speed for element 200397 (-866921.898491, -1016517.056684)
```

```
-----
Dist  Select range                               Elm count  Time          Fast query
500   ( -867172, -1016767, -866672, -1016267) --> 24 elements  2.766000s    0.000000s
1000  ( -867422, -1017017, -866422, -1016017) --> 56 elements  2.750000s    0.000000s
5000  ( -869422, -1019017, -864422, -1014017) --> 865 elements 2.688000s    0.000000s
10000 ( -871922, -1021517, -861922, -1011517) --> 2710 elements 2.816000s    0.015000s
20000 ( -876922, -1026517, -856922, -1006517) --> 10124 elements 3.187000s    0.000000s
30000 ( -881922, -1031517, -851922, -1001517) --> 22484 elements 2.781000s    0.000000s
50000 ( -891922, -1041517, -841922, -991517)  --> 54139 elements 2.828000s    0.000000s
```

```
Spatial window speed for element 175151 (-576543.180379, -1123473.534348)
```

```
-----
Dist  Select range                               Elm count  Time          Fast query
500   ( -576793, -1123724, -576293, -1123224) --> 8 elements  2.187000s    0.016000s
1000  ( -577043, -1123974, -576043, -1122974) --> 22 elements 2.188000s    0.000000s
5000  ( -579043, -1125974, -574043, -1120974) --> 526 elements 2.219000s    0.000000s
10000 ( -581543, -1128474, -571543, -1118474) --> 2571 elements 2.534000s    0.000000s
20000 ( -586543, -1133474, -566543, -1113474) --> 10070 elements 2.219000s    0.000000s
30000 ( -591543, -1138474, -561543, -1108474) --> 22449 elements 2.172000s    0.016000s
50000 ( -601543, -1148474, -551543, -1098474) --> 53792 elements 2.219000s    0.000000s
```

```
Spatial window speed for element 772893 (-817573.869888, -1126753.275287)
```

```
-----
Dist  Select range                               Elm count  Time          Fast query
500   ( -817824, -1127003, -817324, -1126503) --> 16 elements  2.828000s    0.000000s
1000  ( -818074, -1127253, -817074, -1126253) --> 68 elements  2.796000s    0.000000s
5000  ( -820074, -1129253, -815074, -1124253) --> 887 elements 2.968000s    0.000000s
10000 ( -822574, -1131753, -812574, -1121753) --> 3030 elements 2.609000s    0.000000s
20000 ( -827574, -1136753, -807574, -1116753) --> 10571 elements 2.815000s    0.000000s
30000 ( -832574, -1141753, -802574, -1111753) --> 22888 elements 2.703000s    0.000000s
50000 ( -842574, -1151753, -792574, -1101753) --> 57866 elements 2.875000s    0.015000s
```

```
Process finished with exit code 0
```

Závěry

1. Potvrdil se předpoklad, že není nutné provádět výřezy z databáze za účelem urychlení prostorových dotazů
2. Potvrdil se předpoklad, že optimalizace SQL dotazu má zásadní vliv na jeho rychlost
3. Předpoklad zřejmě platí podobně i pro ostatní moderní geodatabázové systémy

Kód testovacího skriptu

```
1  # -*- coding: utf-8 -*-
2  __author__ = "raugustyn"
3
4  import time
5  from dataconnectors import connection as database
6
7  RETRIEVE_CENTERPOINT_SQL = """
8  with center_point as (
9      select ST_Line_Interpolate_Point ((st_dump(wkb_geometry)).geom, 0.5) as geom from public.z_terennirelief_1 where ogc_fid = 200397
10 )
11
12 select st_Xmin(geom), st_Ymin(geom) from center_point"""
13
14 CREATE_TEMP_TABLE_SQL = """
15 drop table if exists temp_candidates;
16 create temp table temp_candidates as
17     select ogc_fid, wkb_geometry from public.z_komruzna_1 where wkb_geometry && st_GeometryFromText('Linestring(-867921.0
-1017517.0, -865921.0 -1015517.0)', 5514)
18     union all
19     select ogc_fid, wkb_geometry from public.z_voda_1 where wkb_geometry && st_GeometryFromText('Linestring(-867921.0 -
1017517.0, -865921.0 -1015517.0)', 5514)
20     union all
21     select ogc_fid, wkb_geometry from public.z_komsilnice_1 where wkb_geometry && st_GeometryFromText('Linestring(-
867921.0 -1017517.0, -865921.0 -1015517.0)', 5514);
22
23     create index wkb_geometry_idx on temp_candidates using gist (wkb_geometry);"""
24
25 SEARCH_FORCANDIDATES_SQL = """
26 with master_element as (
27     select wkb_geometry from public.z_terennirelief_1 where ogc_fid = 200397
28 ),
29
30 slave_elements as (
31     select * from public.z_komruzna_1 as slave_table, master_element
32     where st_distance(master_element.wkb_geometry, slave_table.wkb_geometry) <= 100
33 )
34
35 select * from slave_elements
36 """
37
38 FASTSEARCH_FORCANDIDATES_SQL = """
39 with master_element as (
40     select wkb_geometry, st_buffer(wkb_geometry, 100) as search_geom from public.z_terennirelief_1 where ogc_fid = 200397
41 ),
42
43 slave_elements as (
```

```
44     select * from public.z_komruzna_l as slave_table, master_element
45         where search_geom && slave_table.wkb_geometry and st_distance(master_element.wkb_geometry, slave_table.wkb_geometry) <= 100
46     )
47
48     select * from slave_elements
49     """
50
51     for elementID in [200397, 175151, 772893]:
52         # Retrieving element center point
53         centerX, centerY = database.executeSelectSQL(RETRIEVE_CENTERPOINT_SQL, { "200397": elementID} ).fetchone()
54
55         print "Spatial window speed for element %d (%f, %f)" % (elementID, centerX, centerY)
56         print "-----"
57         print "Dist    Select range                                Elm count    Time        Fast query"
58         for distance in [500.0, 1000.0, 5000.0, 10000.0, 20000.0, 30000.0, 50000.0]:
59             # Calculating view extent
60             delta = distance/2
61             minX = centerX - delta
62             minY = centerY - delta
63             maxX = centerX + delta
64             maxY = centerY + delta
65
66             # Creating temporary table
67             database.execute(CREATE_TEMP_TABLE_SQL,
68                 {
69                     "-867921.0" : minX,
70                     "-1017517.0" : minY,
71                     "-865921.0" : maxX,
72                     "-1015517.0" : maxY
73                 }
74             )
75
76             # Reading selected row number
77             rowCount = database.executeSelectSQL("select count(*) from temp_candidates;").fetchone()[0]
78
79             # Querying via normal query
80             startTime = time.time()
81             candidates = database.executeSelectSQL(SEARCH_FORCANDIDATES_SQL, { "200397": elementID} )
82             elapsedTime = time.time() - startTime
83
84             # Querying via optimized query
85             startTime = time.time()
86             candidates = database.executeSelectSQL(FASTSEARCH_FORCANDIDATES_SQL, { "200397": elementID} )
87             fastElapsedTime = time.time() - startTime
88
89             # Printing results
90             print "%5.0f (%8.0f, %8.0f, %8.0f, %8.0f) --> %5d elements %fs %fs" % (distance, minX, minY, maxX, maxY, rowCount, elapsedTime,
fastElapsedTime)
91         print
```

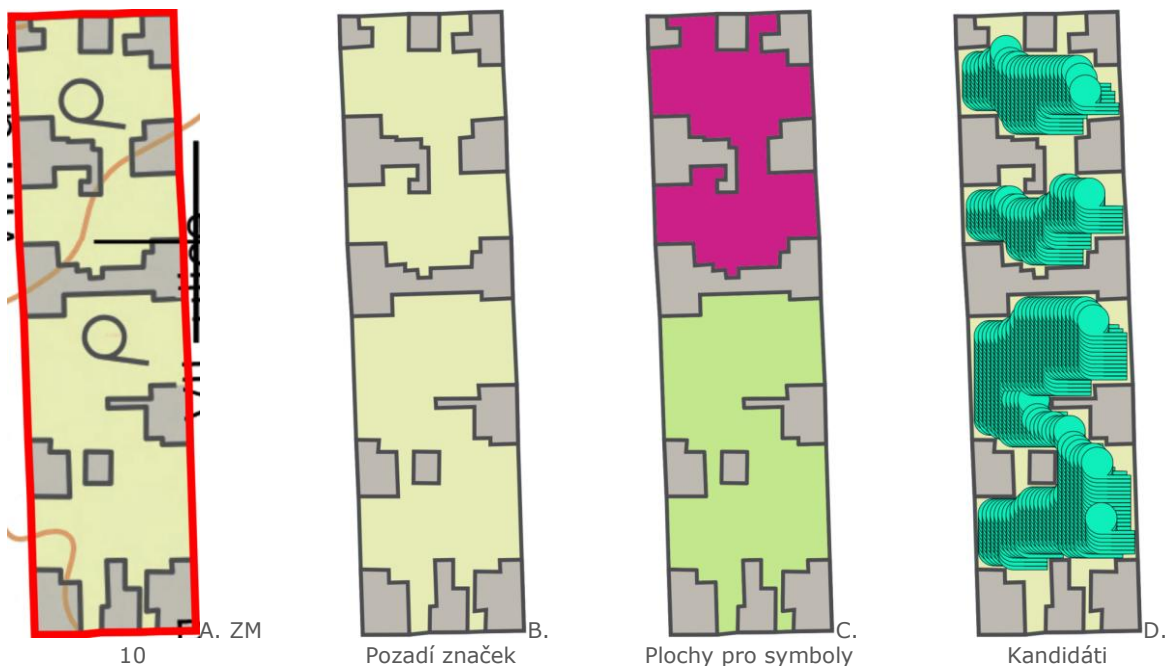
Měření kvality kresby

Generalizace je komplexní proces kdy se kartograf snaží vzájemně sladit velké množství často protichůdných požadavků na mapovou kresbu jednotlivých prvků tak, aby dosáhl optimálního znázornění. Proces generalizace nemá až na výjimky jedno řešení, jedná se o optimalizační proces, kdy se snažíme hledat nejvhodnější řešení v daných podmínkách. Kartograf je limitován množstvím času, který může sestavením a generalizací mapového listu strávit. Podobně při automatické generalizaci jsme limitováni výpočetním výkonem, velikostí paměti, množstvím dostupných algoritmů apod.

Jednoduché měření

Na rozdíl od generalizace operátorem je v prostředí automatizované generalizace nutné zabezpečit automatické vyhodnocení úspěšnosti řešení ve všech jeho variantách. Princip komplexního měření modelu si vysvětlíme na zjednodušeném příkladu umístování značky druhu porostu parků. Jednotlivé plochy mohou být zakryty či rozděleny kresbou dalších objektů jako jsou budovy, cesty, silnice a podobně. Na každou plochu chceme co nejlépe umístit značku parku, přičemž máme možnost tří velikostí značky. Úloha je pro kartografa po krátkém zácvičku poměrně nenáročná. Výsledné umístění značek vidíme na obrázku A.

Z vektorové kresby podkladových prvků (B) získáme jednoduchou analýzou dvě plochy (C), do kterých bychom rádi umístiti značky. Zjednodušíme zde pro jednoduchost možné varianty řešení na diskrétní matici bodů o vhodném rozestupu, ve které vygenerujeme kandidáty (D) na umístění značky.



Požadavek na co nejlepší umístění značky můžeme definovat například takto:

- značka by měla být pokud možno v co největším volném prostoru plochy, což můžeme vyjádřit pomocí vzdálenosti obrysu značky k ploše, kterou popisuje (obrázek E). Čím větší vzdálenost, tím lépe
- značka by měla být umístěna co nejbližší středu plochy. Tento požadavek vyjádříme jako vzdálenost vztažného bodu značky od středu plochy (obrázek F). Čím menší vzdálenost, tím lépe

Obrázky E a F jsou zabarveny směrem od bílé k zelené, kde zelená je maximální vzdálenost daného požadavku.



Vidíme že význam barevnosti na obou obrázcích je právě opačný, normalizujeme ho tedy do plynulého přechodu hodnot v intervalu $[0, 1]$ dle následující tabulky:

Barva	Hodnota	Význam
Zelená	0	Optimální splnění požadavku
Červená	1	Poslední akceptovatelná hodnota

Takto normalizované měřené veličiny se často označují jako tzv. happenies/cost funkce, protože začínají na nulové hodnotě, kdy jsme s výsledkem maximálně spokojeni (happy) a končí na hodnotě, kdy důsledky generalizačních úprav, tzn. náklady na generalizaci z hlediska znehodnocení modelu jsou maximální.

Jako optimální polohu značky budeme chápat kandidáta s minimální délkou vektoru definovaného požadavky na něj. V některých případech, přestože máme jednotlivé požadavky normalizovány do lineárního průběhu v jednotkovém vektoru, nejsou jednotlivé požadavky rovnocenné. Zde například do určité míry preferujeme požadavek na volné místo okolo značky před požadavkem na co nejmenší vzdálenost od středu plochy, což můžeme zabezpečit zavedením vah jednotlivých měřených veličin. Tyto váhy je možné určit empiricky, nejlépe analýzou dat vyhodnocených kartografem.

Výpočet hodnoty úspěšnosti řešení určuje následující rovnice

$$\text{cost} = \sum_{k=0}^n k_i * x_i^2$$

kde k_i jsou hodnoty vah jednotlivých veličin x_i mezi sebou.

Složitější měření

Uvažujme situaci na následujícím obrázku, která znázorňuje kartografem odsunutě bodové značky, červenými kruhy jsou znázorněny skutečné polohy objektů.

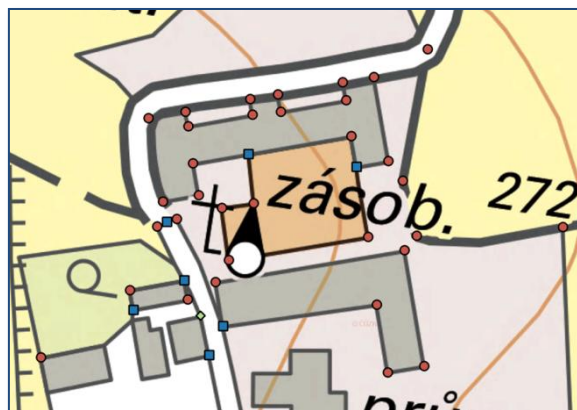


Odsun bodových značek

Cíle sledované kartografem jsou poměrně přímočaré:

1. Kresba značek by se neměla překrývat a měla by být od sebe odsazena;
2. značky by měly být umístěny nejlépe ve skutečné poloze objektů, které reprezentují;
3. orientace značek mezi sebou by měl být dodržena;
4. značky musí být umístěny tak, aby nepřekrývaly charakteristické části podkladové kresby.

Je zřejmé, že postupem uvedeným v minulé kapitole jsme schopni snadno vyčíslit hodnoty prvních tří požadavků. Čtvrtý požadavek však vyžaduje strukturální analýzu podkladové kresby, její rozčlenění nejen z hlediska důležitosti objektů, ale také charakteristických vzorů jako jsou okolí rohy, styčných míst několika prvků a podobně.



Klíčové body podkladové kresby

Je zřejmé, že i jen přiblížení se kvalitě zpracování kartografy vyžaduje často velké množství analytické a programátorské práce s nejistým výsledkem. Vzhledem k omezeným lidským zdrojům je nutné neustálé přehodnocování poměru mezi potenciálním zvýšením kvality a náklady na další krok. To vede k požadavku na vývoj agilními metodami vývoje.