

Univerzita Karlova
Přírodovědecká fakulta



Algoritmy počítačové kartografie

Úloha 4: Energetické spliny.

Adam Kulich, Markéta Růžicková, Eliška Siegllová

N-GKDPZ

Praha 2023

1 Zadání

S využitím generalizačního operátoru Partial Modification realizujte odsun a částečnou změnu tvaru jednoho blízkého prvku vůči blízkému pevnému prvku (bariéře) tak, aby v generalizované mapě nedošlo k jejich grafickému konfliktu. Hodnotu minimální vzdálenosti prvků d volte v závislosti na měřítkovém čísle mapy (např. 1 mm v mapě). Pro implementaci využijte metodu energetických splínů.

Jako vstupní data použijte existující kartografická data (např. silniční či železniční síť, vodstvo), která budou načítána ze dvou textových souborů ve vámi zvoleném formátu.

Grafické rozhraní realizujte s využitím frameworku QT, výsledky generalizačních operací vizualizujte. Porovnejte dosažené výsledky s ruční generalizací prováděnou kartografickým expertem.

Hodnocení:

Krok	Hodnocení
Partial Modification: 1 prvek a překážka	20b
<i>Partial Modification: 2 blízké prvky</i>	<i>+10b</i>
<i>Partial Modification: 2 blízké prvky a překážka</i>	<i>+15b</i>
<i>Partial Modification: 3 prvky</i>	<i>+15b</i>
<i>Variabilní počet iterací, polohová přesnost menší než grafická přesnost mapy</i>	<i>+5b</i>
Max celkem:	55b

Čas zpracování: 3 týdny.

2 Údaje o bonusových úlohách + body

V rámci tohoto úkolu nebyly zpracovány žádné bonusové úlohy.

3 Popis a rozbor problému + vzorce

3.1 Kartografické generalizační algoritmy

Při kartografické generalizaci dochází k výběru, zjednodušení či zevšeobecnění prvků v mapě v důsledku změny měřítka. Při přechodu z většího do menšího měřítka může být v mapě zobrazena pouze podmnožina původních dat, dochází proto k úpravě jejich prostorové reprezentace. Jeden krok generalizace navíc vyvolá další, například když se generalizují silnice, je nutné upravit i budovy podél nich, protože může nastat situace, že silnice bude budovy protínat.

Změny prostorové reprezentace dat, ke kterým dochází při kartografické generalizaci, se dělí do tří skupin: geometrické transformace, transformace vzájemných vztahů a tematické transformace. Cílem geometrické transformace je úprava geometrických parametrů, čehož se docílí například přemístěním, zjednodušením, eliminací či vyhlazením. Transformace vzájemných vztahů upravuje vzájemnou polohu prvků v mapě, dochází zde k úpravě topologických vztahů. Při tematické transformaci dochází ke změně kartografické reprezentace objektů, například agregací či regionalizací.

Pro automatickou generalizaci máme dvě základní strategie. Jednou je využití umělé inteligence a strojového učení, kdy známe vstupní data a generalizovanou mapu, které lze použít jako trénovací množiny. Druhou strategií je rule-based system, při kterém jsou generalizační pravidla převedena na algoritmy.

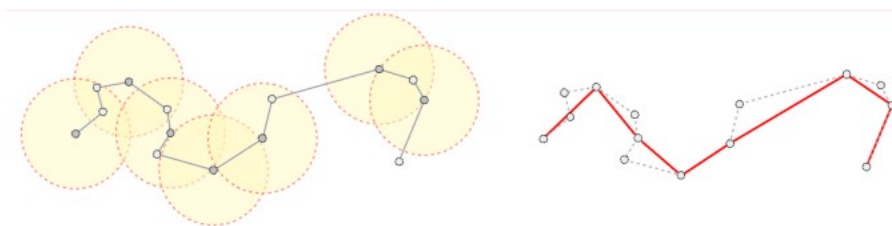
Problémem kartografické generalizace je, že je to NP-problém, u kterého nelze najít nejlepší řešení, pouze se k němu přiblížit. Existuje proto mnoho různých metod, u kterých je obtížné určit, která z nich je nejlepší. Další překážkou je, že je obtížné určit, co je "správná generalizace", protože to je velice subjektivní a záleží na vnímání každého z nás, co je vizuálně "nejhezčí" výsledek.

Při rule-based generalizaci jsou generalizační pravidla převedena na algoritmy, ze kterých je posléze vytvořen model. V těchto modelech vznikají generalizační schémata, které jsou tvořeny generalizačními operátory v určitém pořadí. Mezi nejdůležitější generalizační operátory patří: přemístění (displacement - posun blízkých prvků), eliminace (elimination - odstranění malých prvků), agregace (aggregation - spojení blízkých prvků), spojení (merge - spojení dvou blízkých prvků v jeden), regionalizace (regionalization - převod skupiny prvků na plošnou reprezentaci), selektivní vynechání (selective omission), typizace (typification - odstranění některých prvků), zjednodušení tvaru (simplification), částečná změna tvaru (partial modification - kombinace simplification a displacement), vyhlazení (smoothing).

3.1.1 Simplification algorithms

Při generalizaci linie se vyberou některé lomové body (tzv. critical points) s některými omezujícími podmínkami. Critical points jsou body, které jsou vyhodnoceny jako významné z globálního pohledu. Bývají to lokální minima, maxima, či inflexní body, další parametry, které jsou posuzovány, jsou vzdálenost od dalšího vrcholu, úhel, křivost, a jiné. Při odstraňování nedůležitých vrcholů je přitom důležité dbát na to, že by se neměly navzájem protínat předměty ani se segmenty sousedních prvků, a odstraňovaný vrchol nesmí být počátečním či koncovým bodem.

Jedním z parametrů, který je využíván pro odstranění nedůležitých bodů, je euklidovská vzdálenost. Ve vrcholech je vytvořena kružnice o poloměru d_{min} , a pokud se okolní body nachází uvnitř této kružnice, jsou vyhodnoceny jako nevýznamné. Tento algoritmus zohledňuje pouze vzdálenost, nikoli úhly, nebere proto v potaz tvar prvku. Na obrázku níže je vidět, že strany mohou mít stejné směrnice, tudíž by prostřední bod mohl být eliminován. Tento algoritmus nedosahuje příliš dobrých výsledků a proto v praxi není hojně využíván.



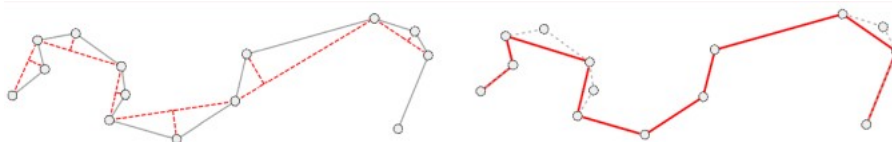
Euklidovská vzdálenost (Bayer, 2023).

Jistou úpravou předchozího algoritmu je kumulativní euklidovská vzdálenost, která odstraňuje ty hrany, jejichž kumulativní vzdálenost je menší než d_{min} , tedy body, které leží příliš blízko sebe. Na obrázku vidíme, že narazíme na stejný problém, kdy není zohledňován tvar prvku a rovněž není v praxi příliš využíváný.



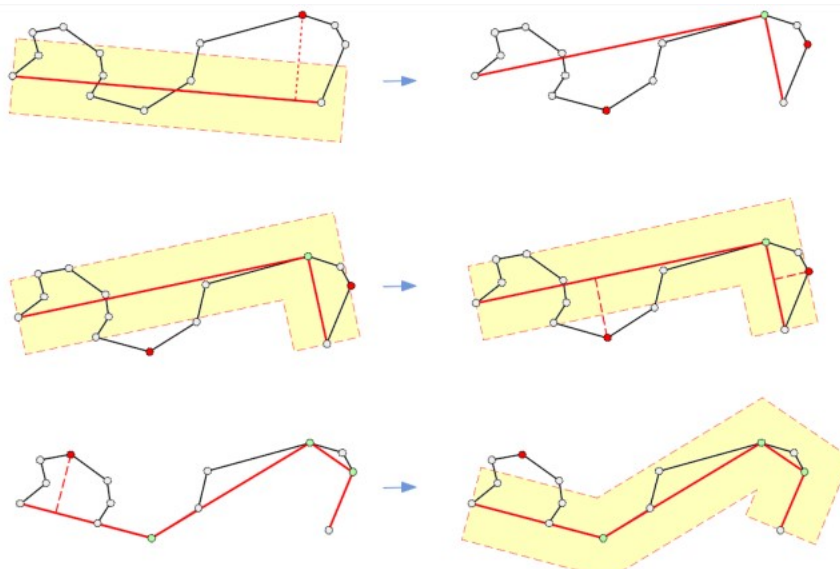
Kumulativní euklidovská vzdálenost (Bayer, 2023).

Další možností je výpočet vzdálenosti bodu p_i od strany $e_i = (p_{i-1}, p_{i+1})$. Pokud $d(p_i, e_i) < d_{min}$, vrchol p_i je vypuštěn. Tento algoritmus odstranil problém, že v datech zůstávají body dále od sebe, ale s podobnou směrnicí, a dosahuje proto poměrně dobrých výsledků.



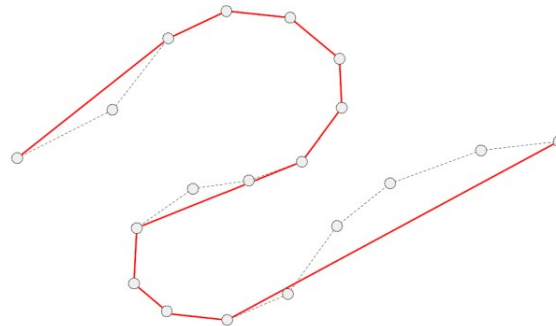
Vzdálenost bodu od strany (Bayer, 2023).

Algoritmy pracující s lokálním rozšířeným okolím, tedy s principem koridorů či bufferů, jsou například Reumann-Witkam, Lang, Min-Max algorithm, či Douglas Peucker algorithm. Tyto algoritmy pracují tak, že okolo strany vytvoří buffer a body buď z datasetu odstraňují na základě toho, že leží příliš blízko, a jsou nedůležité (tedy uvnitř koridoru), nebo začnou s prvním a posledním bodem, a poté do datasetu přidávají body, které leží vně koridor (jsou tedy důležité pro vystihnoutí tvaru linie). Na obrázku je ukázka Douglas-Peuckerova algoritmu, který dosahuje velice dobrých výsledků a je velmi často implementován v GIS. Zachovává ovšem tvar, nikoli křivost, a proto je pro některé aplikace nevhodný (například pro vodstvo).



Douglas-Peuckerova metoda (Bayer, 2023).

Poté máme globální metody simplifikace linií, kterou je například Bend Simplify. Pracuje s detekcí oblouků a vyhodnocením jejich významnosti, pracuje s tzv. compactness index popisující tvar oblouku. Tato metoda preferuje zachování křivosti oproti tvaru a je také hojně implementována v GIS, ovšem pro jiné datasety - právě například vodstvo, či silnice, kde je zachování křivosti důležité. Ukázka výstupu metody Bend Simplify je na následujícím obrázku.



Bend Simplify (Bayer, 2023).

Při generalizaci plošných prvků je možné využít stejné algoritmy, kdy se využije dekompozice prvku na polylinie.

3.1.2 Vyhlašovací algoritmy

Rozdílem mezi generalizačními a vyhlazovacími algoritmy je ten, že generalizační algoritmy pracují pouze s body z původního datasetu. Vyhlašovací algoritmy do dat přidávají nové vrcholy, některé posunují, a některé odstraňují. Cílem je, aby výsledek vypadal vyhlazeněji a přirozeněji, aby nedocházelo k náhlým změnám křivosti.

3.1.3 Displacement

Operátor displacement či odsun se často používá pro bodové znaky, či pro liniové prvky jako kombinace odsunu a modifikace tvaru. Když jsou na mapě dva prvky, které jsou v konfliktu, odsun u dlouhého liniového objektu může způsobit, že bude v konfliktu v jiné části. V praxi se odsun řeší tak, že by se u objektů neměly protínat bounding boxy. Možnosti posunu jsou ale nekonečné, proto k výpočtu využíváme vztažný bod bounding boxu a snažíme se minimalizovat sumy čtverců posunu vztažných bodů tak, aby zároveň plocha překryvu čtverců byla nulová. Displacement se využívá například pro optimalizaci umístění bodových znaků nebo popisů.

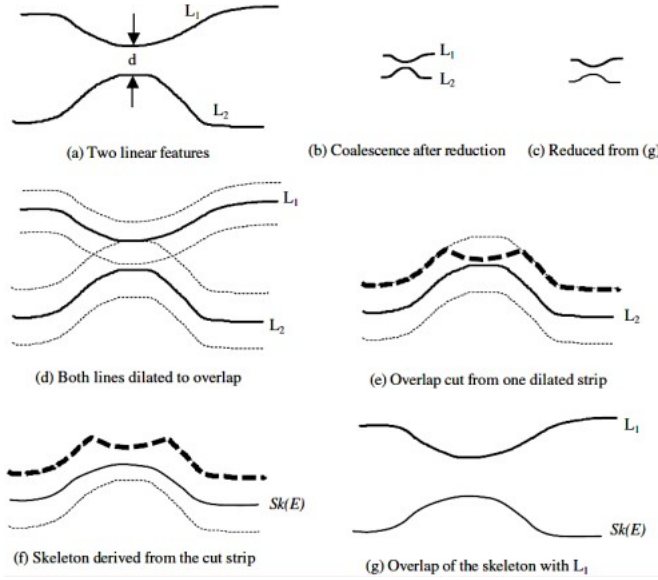
3.1.4 Energetický model

V energetickém modelu si min-max boxy představíme jako magnety a jde o jakousi simulaci fyzikálních sil. Pomocí iteračního procesu se poté model snaží dojít k rovnovážnému stavu.

3.1.5 Partial Modification

Operátor partial modification kombinuje operátory Displacement a Modification, čímž se z něj stává komplexní operátor se složitou algoritmizací. Využívá se zejména pro liniové prvky, kdy posun je nedostatečným operátorem a je potřeba v některých částech i měnit tvar. Využívá se u prvků, kde dochází ke grafickým kolizím a souběhům, jako např. silniční síť, železniční síť, vodní toky. Při použití tohoto operátoru dochází k řetězení reakcí, protože odsunutí a modifikace jednoho prvku vede k jeho přiblížení k prvku jinému. Prakticky se k implementaci Partial Modification využívá topologická kostra (kombinace straight skeleton + odsun), či energetických modelů.

Máme dva prvky, které chci posunout (a). Vytvoří se buffer (d), pokud se buffery nepřekrývají, s prvky se nic nestane, pokud se překrývají, od spodního bufferu se odečte vrchní buffer a vznikne výřez (e). Poté zkonstruujeme skeleton (f), podle kterého se modifikuje tvar.



Partial Modification, topologický model (Bayer, 2023).

Důležité je, aby změna tvaru byla přirozená, a aby se v mapě nenacházely žádné ostré lomy. Postupný náběh změny se nazývá postupná propagace a využívá se například v situacích, kdy vodní tok musí obejít budovu. Postupná propagace má dva parametry: prvním parametrem je buffer, druhým je zóna propagace. Zóna přiblížení je uvnitř bufferu (např. okolo budovy), pokud se objekt (např. vodní tok) nachází uvnitř tohoto bufferu, musí se odsunout. V zóně propagace poté dochází k tvorbě plynulého přechodumezi původním a modifikovaným směrem a pomocí kubické křivky se snaží prvek odsunout tak, aby změna vypadala přirozeně.

3.2 Energetický spline

Energetické spliny jsou občas nazývány "Snakes" (Kaas, 1988) a nejčastější varianta, která se používá, je Partial Modification. Jedná se o složitější modely než se snaží o minimalizaci celkové energie a modifikuje se více prvků současně. Parametrická rovnice energetického splinu vypadá následovně:

$$v(s) = (x(s), y(s)) \quad (1)$$

a je odvozena z podmínky minimalizace energie:

$$E(s) = \int_0^1 E_i(s) ds + \int_0^1 E_e(s) ds. \quad (2)$$

Parametrická rovnice operátoru Partial Modification vypadá následovně:

$$d(s) = (x(s) - (x_{(0)}(s), y(s) - y_{(0)}(s)), \quad (3)$$

kde $x_{(0)}$ a $y_{(0)}$ jsou souřadnice vstupního prvku, x a y jsou souřadnice generalizovaného prvku. Parametr $s \in \langle 0, 1 \rangle$. Energetický model pro křivku L o délce l vypadá následovně:

$$E(d) = \int_l E_i(s) ds + \int_l E_e(s) ds, \quad (4)$$

kde E_i je vnitřní energie (internal energy) a E_e vnější energie (external energy). Tyto ovlivňují tvar splinu.

Vnitřní energie E_i ovlivňuje odchylku od původní linie na základě parametrů napětí (tension) a tuhost (stiffness) následovně:

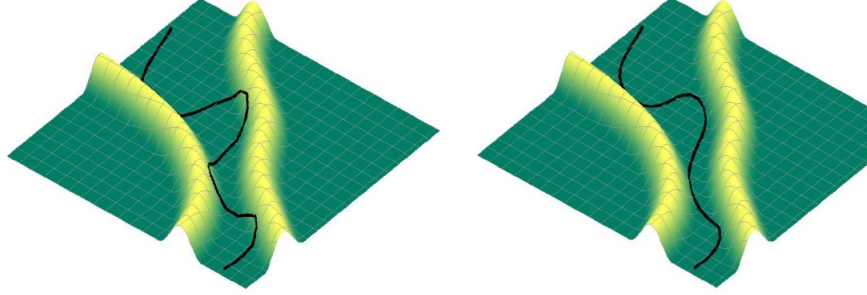
$$E_i(s) = \frac{1}{2} \left(\alpha(s) \|v(s)\|^2 + \beta(s) \left\| \frac{dv(s)}{ds} \right\|^2 + \gamma(s) \left\| \frac{d^2v(s)}{ds^2} \right\|^2 \right), \quad (5)$$

kde α, β, γ jsou konstanty, které modelují parametry tension a stiffness.

Vnější energie je dané vnějšími silami a je definovaná uživatelem, může mít tedy mnoho podob. V případě metody Partial displacement je cílem nepřiblížit se k jinému prvku na vzdálenost menší než d_{min} , cílem je udržet prvek uvnitř nějakého bufferu definovaném jinými prvky. Příklady různých energetických schémat jsou:

$$E_e = \begin{cases} \frac{d_i}{d_{min}} & d_i < d_{min}, \\ 0 & d_i \geq d_{min}, \end{cases} \quad (6)$$

$$E_e = \begin{cases} \frac{(d_{min}-d_i)d_i}{d_i} & d_i < d_{min}, \\ 0 & d_i \geq d_{min}. \end{cases} \quad (7)$$



Vliv vnější energie na výsledný prvek (Bayer, 2023).

V případě metody **Partial Displacement**, pokud se prvek posune na vzdálenost $< d_{min}$ k jinému prvku, dojde k jeho posunu a změně tvaru v zasažené části. V praxi může dojít k jednomu ze tří schémat: (1) částečná modifikace jednoho prvku, (2) částečná modifikace obou prvků a (3) kombinace obou případů.

V případě částečné modifikace obou prvků je jeden prvek bariéra, která je neměnná. U druhého prvku dochází k modifikacím, pokud se nachází ve vzdálenosti $< d_{min}$ ke druhému prvku. V praxi toto může být například silniční síť a vodních toků, kdy vodní toky jsou neměnný prvek a silniční síť může být modifikovaná.

Algorithm 1 Částečná modifikace jednoho prvku

máme generalizovaný prvek $L = (X, Y)$ a bariéru $B = (X', X')$ v maticové reprezentaci,

kde $X = (n, 1), Y = (n, 1), X' = (m, 1), Y' = (m, 1)$

1. výpočet kroku h

výpočet souřadnicových rozdílů δX a δY ($\delta X = x_{i+1} - x_i, \delta Y = y_{i+1} - y_i$)

výpočet vzdáleností mezi vrcholy generalizované polylinie ($H = \sqrt{\|\delta X\|_2 + \|\delta Y\|_2}$)

odvození $h =$ střední hodnota H

2. určení prvků matice A

určíme koeficienty α, β, γ

naplnění matice s využitím koeficientů

3. iterační proces (popsaný v algoritmu č. 2)

4. konverze maticové reprezentace X, Y na spojový seznam vrcholů

Algorithm 2 Částečná modifikace jednoho prvku: iterační proces

```
inicializujeme
   $i = 0$ 
   $L_0 = (X_0, Y_0)$ 
   $X_0 = X, Y_0 = Y$ 
   $\Delta S_0 = (\Delta X_0, \Delta Y_0) = 0$ 
   $max\_iters = 100$  ▷ v praxi je potřeba řádově nižší stovky max. iterací
for  $do_i$  in range( $max\_iters$ )
   $E_x = (n, 1), E_y = (n, 1)$  ▷ vytvoření nových matic
  for  $dop_i$  in  $L$  ▷ pro každý bod  $p_i$  v matici  $L$ 
     $p_i = (x_i, y_i)$ 
    nalezneme nejbližší bod  $q_n \in B$ 
    výpočet parciálních derivací vnější energie podle  $\frac{\partial E_e(x_i, y_i)}{\partial x_i}, \frac{\partial E_e(x_i, y_i)}{\partial y_i}$ 
  end for
  určení hodnot posunů  $\Delta S_i$  podle  $\Delta S_i = (\Delta X_i, \Delta Y_i) = (B^{-1}(\lambda \Delta X_{i-1} - E_{e,x}), B^{-1}(\lambda \Delta Y_{i-1} - E_{e,y}))$ 
  výpočet nových souřadnic vrcholů splinu podle  $L_i = (X_i, Y_i) = L_0 + \Delta S_i = (X_0 + \Delta X_i, Y_0 + \Delta Y_i)$ 
   $i = i + 1$ 
  if  $i \geq max\_iters$  then
    ukonči proces
  end if
end for
```

V případě částečné modifikace obou prvků není ani jeden z prvků pevný, a mohou se modifikovat oba. V praxi toto mohou být například prvky silniční sítě.

Algorithm 3 Částečná modifikace obou prvků

```
máme generalizované prvky  $L = (X, Y)$  a  $L' = (X', Y')$  v maticové reprezentaci,
kde  $X = (n, 1), Y = (n, 1), X' = (m, 1), Y' = (m, 1)$ 
1. výpočet kroku  $h$ 
  výpočet souřadnicových rozdílů  $\delta X, \delta Y, \delta X', \delta Y'$ 
  podle  $\delta X = x_{i+1} - x_i, \delta Y = y_{i+1} - y_i, \delta X' = x'_{i+1} - x'_i, \delta Y' = y'_{i+1} - y'_i$ 
  výpočet vzdáleností mezi vrcholy generalizované polylinie
  podle  $H = \sqrt{\|\delta X\|_2 + \|\delta Y\|_2}$  a  $H' = \sqrt{\|\delta X'\|_2 + \|\delta Y'\|_2}$ 
  odvození  $h$  a  $h' \rightarrow$  střední hodnoty  $H$  a  $H'$ 
2. určení prvků matic  $A$ 
  určíme koeficienty  $\alpha, \beta, \gamma$ 
  naplnění matice  $A, A'$  využitím koeficientů a kroků  $h, h'$ 
   $B = A + \lambda I, B' = A' + \lambda I$ 
3. iterační proces (popsaný v algoritmu č. 4)
4. konverze maticové reprezentace  $X, Y, X', Y'$  na spojové seznamy vrcholů
```

Algorithm 4 Částečná modifikace obou prvků: iterační proces

```
inicializujeme
   $i = 0$ 
   $L_0 = (X_0, Y_0)$ ,  $L'_0 = (X'_0, Y'_0)$ 
   $X_0 = X, Y_0 = Y, X'_0 = X', Y'_0 = Y'$ 
   $\Delta S_0 = (\Delta X_0, \Delta Y_0) = 0, \Delta S'_0 = (\Delta X'_0, \Delta Y'_0) = 0$ 
   $max\_iters = 100$  ▷ v praxi je potřeba řádově nižší stovky max. iterací
for  $do_i$  in range( $max\_iters$ )
   $E_x = (n, 1), E_y = (n, 1), E'_x = (n, 1), E'_y = (n, 1)$  ▷ vytvoření nových matic
  for  $dop_i$  in  $L$  ▷ pro každý bod  $p_i$  v matici  $L$ 
     $p_i \in L, p_i = [x_i, y_i]$ 
    nalezneme nejbližší bod  $q_n \in L'$ 
    výpočet parciálních derivací vnější energie podle  $\frac{\partial E_e(x_i, y_i)}{\partial x_i}, \frac{\partial E_e(x_i, y_i)}{\partial y_i}$ 
     $q_i \in L', q_i = [x_i, y_i]$ 
    nalezneme nejbližší bod  $p_n \in L$ 
    výpočet parciálních derivací vnější energie podle  $\frac{\partial E_e(x_i, y_i)}{\partial x_i}, \frac{\partial E_e(x_i, y_i)}{\partial y_i}$ 
  end for
  určení hodnot posunů  $\Delta S_i, \Delta S'_i$ 
     $\Delta S_i = (\Delta X_i, \Delta Y_i) = (B^{-1}(\lambda \Delta X_{i-1} - E_{e,x}), B^{-1}(\lambda \Delta Y_{i-1} - E_{e,y}))$ 
     $\Delta S'_i = (\Delta X'_i, \Delta Y'_i) = (B'^{-1}(\lambda \Delta X'_{i-1} - E'_{e,x}), B'^{-1}(\lambda \Delta Y'_{i-1} - E'_{e,y}))$ 
  výpočet nových souřadnic vrcholů splinu podle
     $L_i = (X_i, Y_i) = L_0 + \Delta S_i = (X_0 + \Delta X_i, Y_0 + \Delta Y_i)$ 
     $L'_i = (X'_i, Y'_i) = L'_0 + \Delta S'_i = (X'_0 + \Delta X'_i, Y'_0 + \Delta Y'_i)$ 
   $i = i + 1$ 
  if  $i \geq max\_iters$  then
    ukončí proces
  end if
end for
```

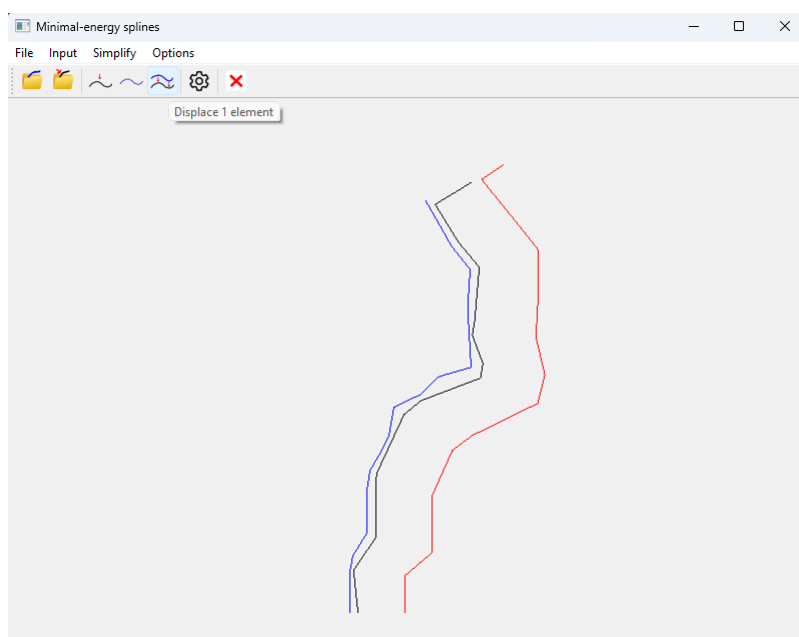
4 Problematické situace a jejich rozbor + ošetření kódu

5 Vstupní data, formát vstupních dat, jejich popis

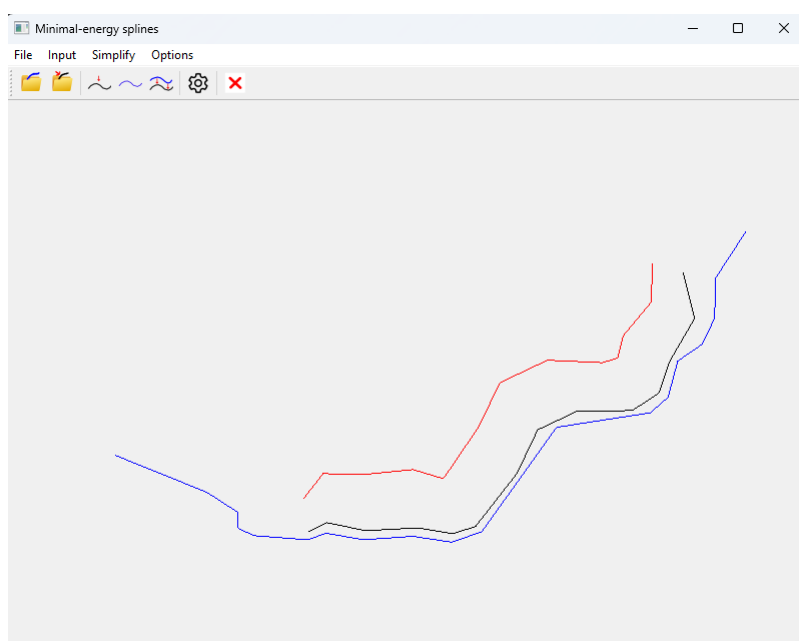
Vstupními daty do aplikace jsou dvě liniové vrstvy ve formátu .shp, nejprve bariéra a následně generalizovaný element. Pro ukázkou funkčnosti algoritmu je třeba použít linie, jejichž vzdálenost je alespoň v jednom místě větší, než parametr minimální vzdálenosti, nastavený v aplikaci. Aplikace umožňuje vybrat ze souboru libovolná data, ale modelová data je možné najít ve složce *data_spline*. Pro ukázkou funkčnosti algoritmu jsou ve složce tři varianty bariéry i elementu. Alternativně je možné vlastní liniová data vytvořit za pomoci ikonek *Barrier* a *Element*. Parametry algoritmu jsou koeficienty *Minimumdistance*, *Alpha*, *Beta*, *Gamma*, *Lambda* a *Iterationnumber*, které je možné změnit v nastavení.

6 Výstupní data, formát výstupních dat, jejich popis a printscreen

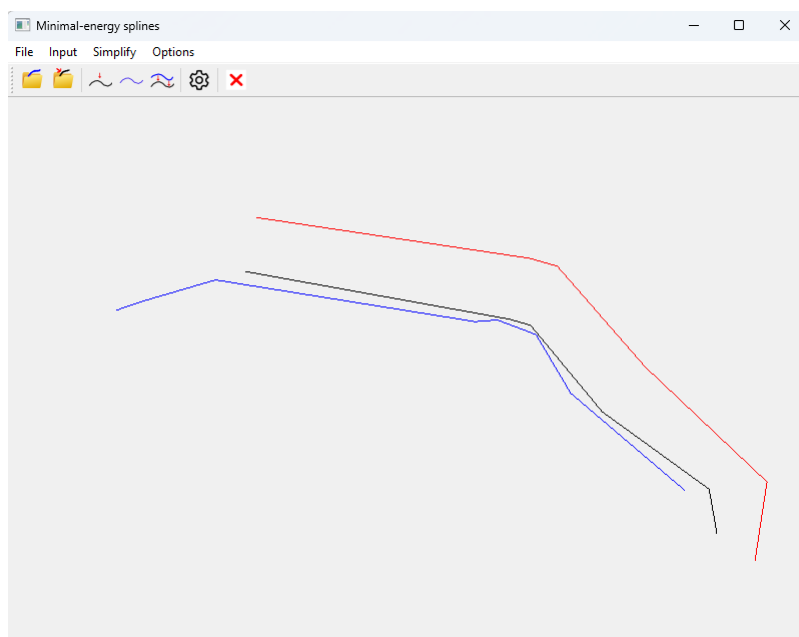
Výstupem aplikace je nová liniová vrstva, vytvořená modifikací generalizovaného prvku. Výstupy při použití různých vstupních dat jsou ukázány níže.



Liniové prvky bariéry (modrá), elementu (černá) a generalizovaného elementu (červená) v první variantě

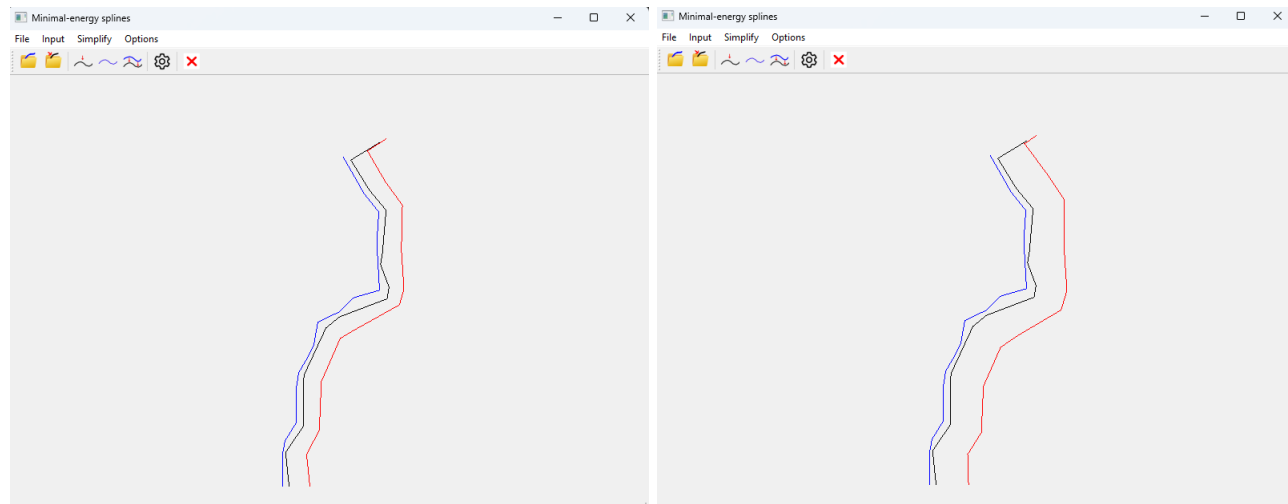


Liniové prvky bariéry (modrá), elementu (černá) a generalizovaného elementu (červená) v druhé variantě

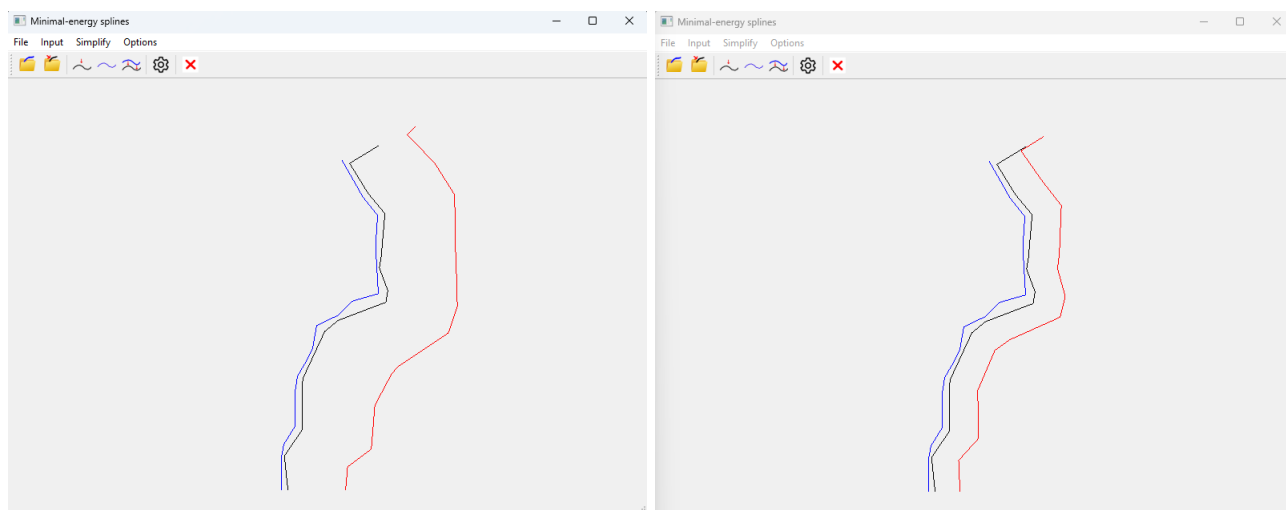


Liniové prvky bariéry (modrá), elementu (černá) a generalizovaného elementu (červená) ve třetí variantě

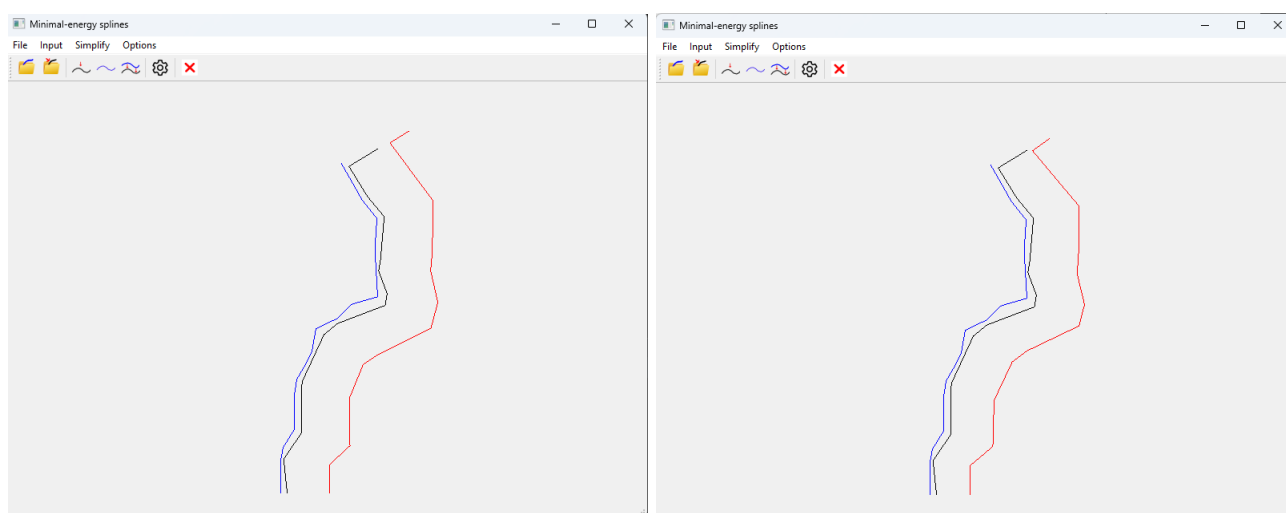
Výše vyobrazené generalizace jsou vytvořeny při ponechání výchozích parametrů výpočtu. Je možné dále experimentovat s parametry, aby bylo dosaženo ideálního výsledku, níže se nachází několik příkladů.



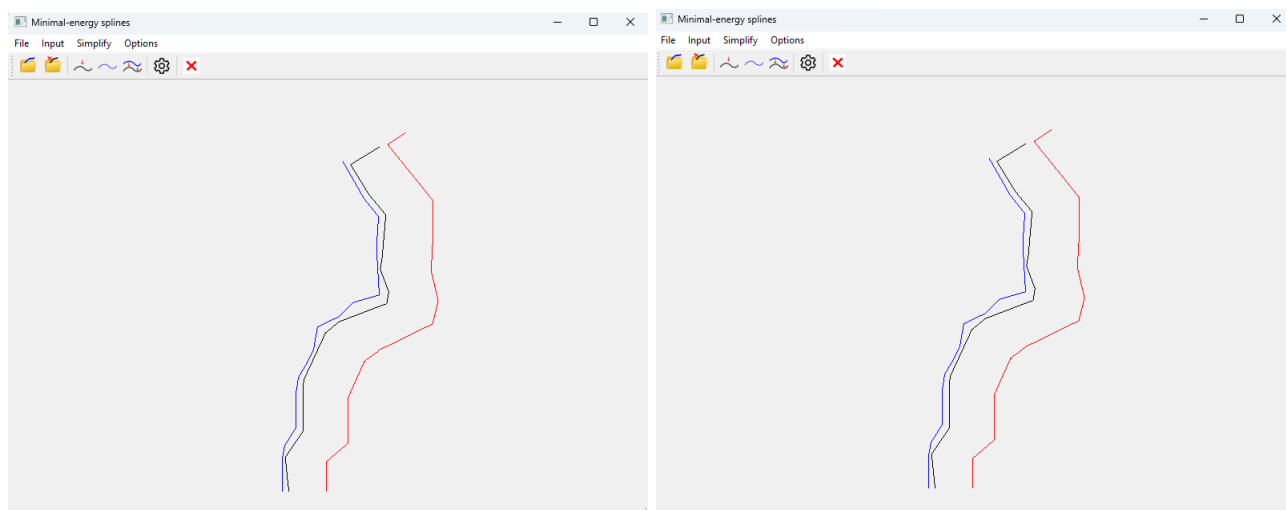
Srovnání výsledků generalizace při nastavení parametru minimální vzdálenosti na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.



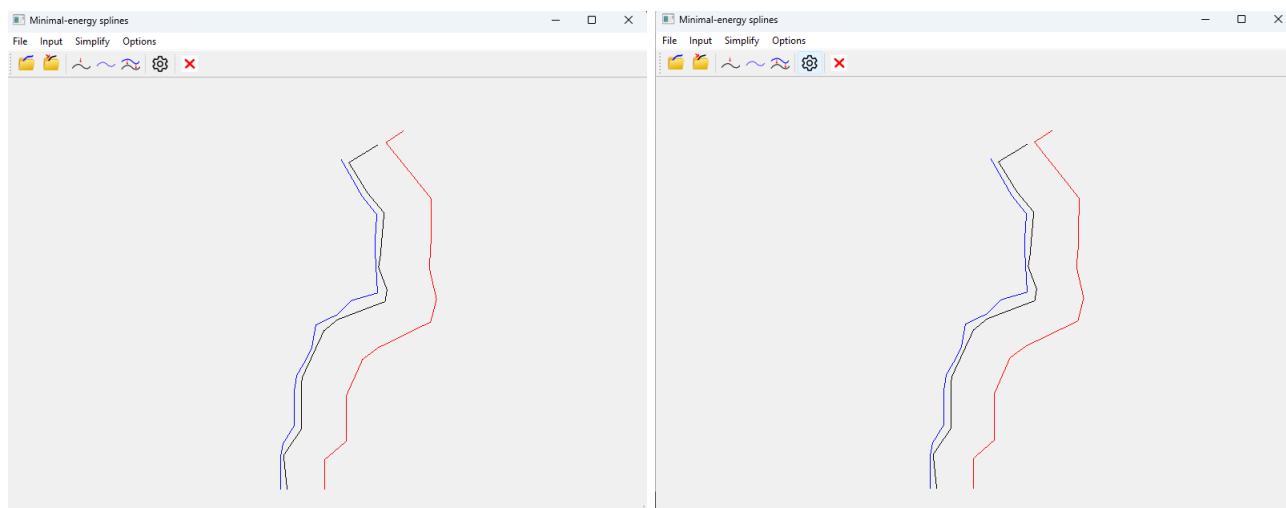
Srovnání výsledků generalizace při nastavení parametru α na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.



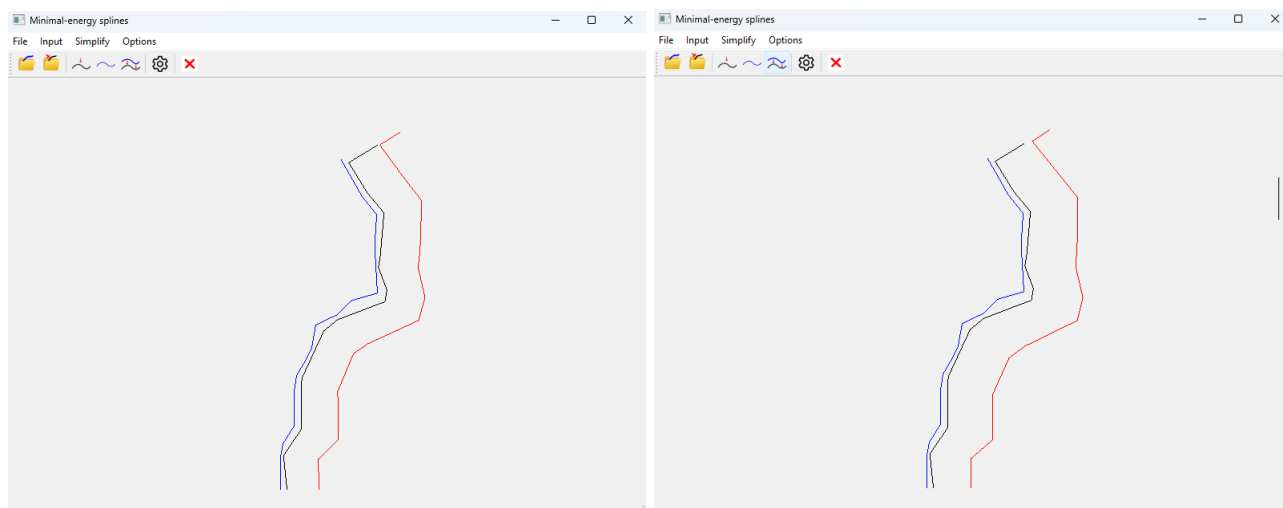
Srovnání výsledků generalizace při nastavení parametru β na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.



Srovnání výsledků generalizace při nastavení parametru Γ na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.



Srovnání výsledků generalizace při nastavení parametru Λ na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.



Srovnání výsledků generalizace při nastavení počtu iterací výpočtu na menší (vlevo) a větší (vpravo) hodnotu. Ostatní parametry byly ponechány na výchozích hodnotách.

Z výsledků je patrné, že při ne příliš drastické změně parametrů $Beta$, $Gamma$ a $Lambda$ nedochází k velkým rozdílům ve výsledné linii. Větší rozdíl je patrný při změně parametru $Alpha$, jehož menší hodnoty implikují větší vzdálenost od původní linie a větší míru zakulacení. Zvětšování parametru minimální vzdálenosti logicky posune celou výslednou generalizovanou linii dále od bariéry a k podobnému důsledku vede i zvyšování počtu iterací. Po zvětšení na určitý počet iterací však již ke změnám ve výsledku nedochází.

7 Dokumentace: popis tříd, datových položek, jednotlivých metod

7.1 Třída MainForm

Tato třída se nachází ve skriptu mainform.py, který slouží ke spouštění programu a jednotlivých algoritmů. Tato třída byla navržena v rozhraní Qt.

7.2 Třída Draw

Třída sloužící k vykreslování jednotlivých objektů. Obsahuje parametry `self._add_L`, `self._L`, `self._B`, `self._LD`, `self._LLoad`, `self._BLoad`, `self._LDLoad`, `self._add_vertex` a parametry splinu `self.dmin`, `self.alpha`, `self.beta`, `self.gamma`, `self.lam`, `self.iters`. Obsahuje metody `mousePressEvent` (na místo kliknutí do widgetu přidá bod do linie), `paintEvent` (vykresluje linie), `switchSource` (mění nastavení `self._add_vertex`, které určuje, jestli se generalizuje bod nebo přidává nový do linie), `getL` (vrací `self._L`), `getB` (vrací `self._B`), `getLLoad` (vrací `self._LLoad`), `getBLoad` (vrací `self._BLoad`), `setLD` (zapisuje vstup do parametru `self._LD`), `setLDLoad` (zapisuje vstup do parametru `self._LDLoad`), `setL` (zapisuje vstup do parametru `self._L`), `setB` (zapisuje vstup do parametru `self._B`), `setSource` (zapisuje vstup do parametru `self._add_L`), `Clear` (maže vstupní data i výsledky generalizace), `setSettings` (funkce spojená s dialogovým oknem ve třídě `Dialog`, která nastaví parametry splinu), `getSettings` (vrací parametry splinu `self.dmin`, `self.alpha`, `self.beta`, `self.gamma`, `self.lam`, `self.iters`), `InvalidInput` (spouští dialogové okno se zprávou o nesprávných vstupních datech v nastavení).

7.3 Třída Algorithms

Třída obsahující metody potřebné pro výpočetní část algoritmů. Obsahuje funkce `getEuclidDistance` (počítá euklidovskou vzdálenost dvou bodů), `getPointLineDistance` (počítá vzdálenost bodu od linie), `getPointLineSegmentDistance` (počítá vzdálenost bodu od segmentu linie), `getNearestLineSegmentPoint` (hledá nejbližší bod na linii k zadanému bodu), `createA` (vytváří maticovou reprezentaci prvků), `getEx` (počítá parciální derivaci vnější energie podle x), `getEy` (počítá parciální derivaci vnější energie podle y), `minEnergySpline` (tvorí novou bodovou vrstvu), `setDefaultSettings` (nastavuje původní nastavení parametrů splinu).

7.4 Třída Dialog

Tato třída obsahuje definované dialogové okno pro vstup uživatelem definovaných parametrů splinu.

7.5 Třída Edge

Třída obsahuje funkce `getStart` (vrací počáteční bod hrany), `getEnd` (vrací koncový bod hrany), `switchOrientation` (vytvoří novou hranu s opačnou orientací), definovaný operátor `==` (slouží ke zjištění, zda jsou dvě hrany shodné).

7.6 Třída Load

Třída slouží k nahrání dat do aplikace. Obsahuje funkce `readPolyline` (přečte polylinii ze shapefilu a vrátí souřadnice bodů), `getPolyBarr` (vrací polygon z linie bariéry), `getPolyElem` (vrací polygon z linie elementu), `numberBarr` (vrací počet bodů v linii bariéry), `numberElem` (vrací počet bodů v linii elementu), `clear` (vymaže načtená data).

8 Závěr, možné či neřešené úlohy, náměty na vylepšení

Aplikace vytvořená v rámci této úlohy umožňuje provádět generalizaci jednoho liniového elementu za pomoci generalizačního operátoru Partial Modification. Generalizovaný element je odsunut a je částečně změněn jeho tvar, tak, aby v mapě nedošlo ke grafickému konfliktu s liniovou bariérou. Uživatel může nahrát vlastní liniovou vrstvu bariéry i elementu, nebo může oba prvky ručně v aplikaci nakreslit. Aplikace také umožňuje nastavení hodnot parametrů pro tvorbu energetických splinů.

Vylepšení aplikace by mohlo spočívat například v možnosti generalizace více blízkých prvků vůči sobě navzájem, nebo více prvků vůči pevné překážce.

9 Seznam literatury

Bayer, Tomáš. 2023. “Kartografické generalizační algoritmy.” Praha. http://web.natur.cuni.cz/~bayertom/images/courses/Adk/adk8_new.pdf.