Algoritam sa fiksnim parametrom za nalaženje minimalne Menhetn mreže

Uroš Ševkušić

Univerzitet u Beogradu, Matematički fakultet

decembar 2022

Uvod

- Autori rada:
  - Kristijan Knauer (Christian Knauer) sa Instituta za informatiku pri Univerzitetu u Bajrojtu (Universität Bayreuth, Institut für Informatik)
  - Adams Spilner (Adams Spillner) sa Instituta za matematiku i informatiku pri Univerzitetu u Grajfsvaldu (Universität Greifswald, Institut für Mathematik und Informatik)
- Rad je objavljen 2011. godine u Journal of Computational Geometry
- Glavni rezultati rada su:
  - konstrukcija algoritma za pronalaženje minimalne Menhetn mreže čija je vremenska složenost $O^*(2^{14h})$, gde je h najmanji broj horizontalnih pravih koje pokrivaju skup P
  - dokaz korektnosti algoritma

Osnovni pojmovi

- *Menhetn mreža* za konačan skup tačaka P u ravni je geometrijski graf $\mathcal{G} = (V, E)$ sa osobinama:
  - $P \subseteq V$
  - grane grafa su vertikalne ili horizontalne duži
  - postoji put od bilo koje dve tačke p i q iz P koji je jednak njihovom L1 rastojanju
  - za svake dve grane $e_1 = \{p_1, q_1\}$ i $e_2 = \{p_2, q_2\}$, ako se duži $\overline{p_1, q_1}$ i $\overline{p_2, q_2}$ seku, njihov presek je $e_1 \cap e_2$
- *Dužina grane* $L(e)$ definiše se kao L1 rastojanje njenih krajnjih tačaka
- *Dužina puta* p sačinjenog od tačaka $p_1, p_2, \ldots p_k$ definiše se kao

$$L(p) = \sum_{i=1}^{k-1} L(\{p_i, p_{i+1}\})$$

- Put p je *monoton* ako važi $L(p) = L(\{p_1, p_k\})$
- *Dužina mreže* $\lambda(\mathcal{G})$ je zbir dužina svih grana u mreži
- Mreža je *minimalna* ako je njena dužina najmanja od svih Menhetn mreža sa skupom tačaka P

Opis algoritma - uvodni pojmovi

- Za svaku tačku q ravni, označićemo njene x- i y- koordinate sa $x(q)$ i $y(q)$ respektivno
- Za skup tačaka P, sortiramo rastuće njene x-koordinate u niz $x_1, x_2, \dots x_l$
- Definišemo skup $X = \{x(p) \mid p \in P\}$ i skupove $P_i = \{p \in P \mid x(p) \leq x_i\}$ za sve $1 \leq i \leq l$
- Sortiramo tačke iz P po y-koordinati u niz $y_1, y_2 \dots y_h$
- Neka je $Y = \{y(p) \mid p \in P\}$ i neka su $L_j = \{q \in \mathbb{R}^2 \mid y(q) = y_j\}$ za sve $1 \leq j \leq h$
- Tačke $v_{i,j} = (x_i, y_j)$ i $V_i = \{v_{i,j} \mid 1 \leq j \leq h\}$

Opis algoritma - uvodni pojmovi

- Skup $R_{i,j}$ je najdešnja tačka $r_{i,j}$, ako takva postoji, u P_i koja leži na L_j
- Za tačku $r_{i,j}$ se kaže da *predstavlja* tačke q takve da je $x(q) \leq x_i$ i $y(q) = y(r_{i,j})$
- Ako u Menhetn mreži za P_i postoji monoton put od tačke q' desno od $r_{i,j}$, onda postoji u toj mreži i monoton put od q' do bilo koje tačke koju $r_{i,j}$ predstavlja
- To znači da, za svaku tačku $v_{i,j}$, kada želimo da proverimo ima li monoton put od nje do neke tačke q, dovoljno je posmatrati tačke q iz skupa $S = \cup_{k=1}^{h} R_{i,k}$
- Za Menhetn mrežu sa tačkama P_i, definišemo skup $R_i = \{j \in \{1, 2, \dots, h\} \mid R_{i,j} \neq \varnothing\}$
- Za skup tačaka P_i, par $\Pi = ((A_1, \dots, A_h), (B_1, \dots, B_h))$ je *prihvatljiv* ako je A_j podskup od $\{j, j+1, \dots, h\} \cap R_i$ i B_j podskup od $\{1, 2, \dots, j-1\} \cap R_i$ za sve $1 \leq j \leq h$
- Osnovna ideja je da se mreža za P_i može opisati prihvatljivim parom kod kojeg su A_j i B_j indeksi k onih tačaka za koje postoji monoton put od $v_{i,j}$ do $r_{i,k}$. Par za koje važi ovo svojstvo se naziva *kanonski* i označava se sa $\pi(\mathcal{G})$

Algoritam

ComputeMMN(P)

Input: a set $P \subseteq \mathbb{R}^2$ of n points
Output: a minimum Manhattan network for P

Compute $X = \{x_1, \ldots, x_l\}$ and $Y = \{y_1, \ldots, y_h\}$.
Initialize an empty collection $\mathcal{C}_1$.
for each pair Π that is admissible for P_1 **do**
    Compute a minimum Manhattan network $\mathcal{N}$ for P_1 and Π.
    Add $\mathcal{N}$ to $\mathcal{C}_1$.
for $i = 1$ to $l - 1$ **do**
    Initialize an empty collection $\mathcal{C}_{i+1}$.
    for each $\mathcal{N}$ in $\mathcal{C}_i$ **do**
        for each $H \subseteq E_{i+1}$ **do**
            Form the Manhattan network $\mathcal{N}'$ by adding the edges in H to $\mathcal{N}$.
            if $\mathcal{N}'$ is a Manhattan network for P_{i+1} **then**
                if $\mathcal{C}_{i+1}$ contains a Manhattan network $\mathcal{N}''$ with $\pi(\mathcal{N}') = \pi(\mathcal{N}'')$ **then**
                    if $\lambda(\mathcal{N}'') > \lambda(\mathcal{N}')$ **then**
                        Remove $\mathcal{N}''$ from $\mathcal{C}_{i+1}$ and add $\mathcal{N}'$ to $\mathcal{C}_{i+1}$.
                else
                    Add $\mathcal{N}'$ to $\mathcal{C}_{i+1}$.
return a Manhattan network $\mathcal{N}$ in $\mathcal{C}_l$ with $\lambda(\mathcal{N})$ minimum.

Algoritam

- Algoritam radi iterativno - grade se kolekcije $\mathcal{C}_i$ Menhetn mreža za skupove P_i
- Algoritam će na kraju iz kolekcije C_l izdvojiti najjeftiniju mrežu i vratiti je kao rezultat
- Ispostaviće se da je algoritam korektan, tj. da je rezultat algoritma zaista minimalna Menhetn mreža za P
- Prvi korak algoritma izgleda ovako:

1. Compute $X = \{x_1, \ldots, x_l\}$ and $Y = \{y_1, \ldots, y_h\}$.
2. Initialize an empty collection $\mathcal{C}_1$.
3. **for each** pair Π that is admissible for P_1 **do**
4. Compute a minimum Manhattan network $\mathcal{N}$ for P_1 and Π.
5. Add $\mathcal{N}$ to $\mathcal{C}_1$.

- Na samom početku algoritma, računamo skupove X i Y
- Zatim se gradi kolekcija C_1
- Sve tačke iz P_1 su na istoj pravoj, pa je dovoljno samo povezati ih vertikalnim dužima da bi se napravila Menhetn mreža
- Tako izgrađenu Menhetn mrežu ubacujemo u kolekciju C_1

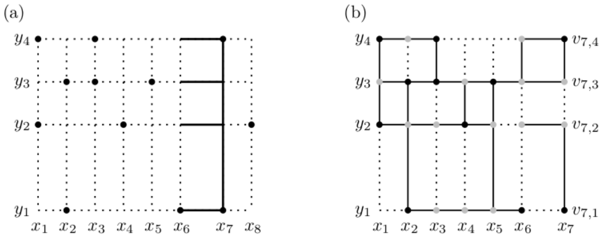
Algoritam

- Sledeća faza je građenje kolekcija C_i za $i \geq 2$:

```
for i = 1 to l − 1 do
    Initialize an empty collection C_{i+1}.
    for each N in C_i do
        for each H ⊆ E_{i+1} do
            Form the Manhattan network N′ by adding the edges in H to N.
            if N′ is a Manhattan network for P_{i+1} then
                if C_{i+1} contains a Manhattan network N″ with π(N′) = π(N″) then
                    if λ(N″) > λ(N′) then
                        Remove N″ from C_{i+1} and add N′ to C_{i+1}.
                else
                    Add N′ to C_{i+1}.
```

- Za svaku mrežu iz prethodne kolekcije, pravimo nove mreže tako što dodajemo nove grane iz skupa grana E_{i+1} u nju
- Skup grana E_{i+1} čine grane $\{v_{i,j}, v_{i+1,j}\}$ i $\{v_{i+1,j}, v_{i+1,j+1}\}$
- Potrebno je proveriti da li mreža ispunjava uslove Menhetn mreža
- Ako je nova mreža jeftinija od neke iz kolekcije sa istim *kanonskim prihvatljivim parom*, onda možemo da izbacimo staru mrežu

Algoritam - ilustracija



- Podebljane linije na levoj slici prikazuju skup grana E_7
- Na desnoj slici smo neke od tih grana izabrali za novu mrežu
- Rezultujuća mreža u ovom koraku nije ispala Menhetn mreža

Redundantni parovi

- Da bismo dokazali korektnost algoritma, potrebno je uvesti još dva pojma
- Kažemo da prihvatljivi par Π čini Π' *redundantnim* ako je:
  - $A'_j \subseteq A_j$
  - $B'_j \subseteq B_j$
  - svaka minimalna Menhetn mreža za P_i i Π' je minimalna Menhetn mreža za P_i i Π
- Za skup P_i, *pokrivač* je skup $\mathcal{A}$ prihvatljivih parova takvih da za sve prihvatljive parove Π za P_i, postoji prihvatljiv par $\Pi' \in \mathcal{A}$ takav da Π' čini Π redundantnim

Dokaz korektnosti

- Na osnovu sledeće tri teoreme, izvodi se dokaz korektnosti algoritma:

Teorema

Neka je i takvo da je $1 \leq i \leq l - 1$. Pretpostavimo da je $\mathcal{A}_i = \{\pi(\mathcal{G}) \mid \mathcal{G} \in C_i\}$ pokrivač i za svako $\Pi \in \mathcal{A}_i$ postoji minimalna Menhetn mreža za P_i i Π u C_i. Tada, za svaki prihvatljivi par Π' za P_{i+1}, postoji Menhetn mreža $\mathcal{G} \in C_i$ i skup $H \subseteq E_{i+1}$ takva da je mreža $\mathcal{G}'$, koja je nastala dodavanjem grana iz H u $\mathcal{G}$, minimalna Menhetn mreža za P_{i+1} i Π'.

Teorema

Neka je i takvo da je $1 \leq i \leq l$. Neka je Π prihvatljiv par za P_i i $\mathcal{G}$ minimalna Menhetn mreža za P_i i Π. Tada $\pi(\mathcal{G})$ čini Π redundantnim.

Teorema

Za svako C_i, $\mathcal{A}_i$ je pokrivač i C_i sadrži minimalnu Menhetn mrežu za P_i i svaki prihvatljivi par Π. Specijalno, C_l sadži minimalnun Menhetn mrežu za P.

Složenost

- Vremenska složenost ovog algoritma je $O^*(2^{14h})$, tj. $O(2^{14h}r(|P|))$, gde je r neki polinom
- Prostorna složenost je $O^*(2^{12h})$
- Dokaz je sproveden u nekoliko faza:
  1. Skupovi A_j i B_j mogu se predstaviti pravougaonicima
  2. Definiše se *kompatibilna šestorka*, struktura koja se sastoji od četiri niza i dva skupa i koja će opisati te pravougaonike
  3. Ispostavlja se da, za svako $\mathcal{A}_i$, postoji injektivno preslikavanje φ iz $\mathcal{A}_i$ u skup kompatibilnih šestorki
  4. Veličina skupa kompatibilnih šestorki je $O^*(2^{12h})$
  5. Primetimo da je $|C_i| = |A_i|$, iz čega sledi da je prostorna složenost $O^*(2^{12h})$
  6. Primetimo da imamo ne više od 2^{2h} podskupova H od E_{i+1}
  7. Vremenska složenost je $O^*(2^{2h \cdot 12h}) = O^*(2^{14h})$