Algorytmy skalujące

Piotr Sankowski

Plan

- Algorytm Gabowa

Plan

- Algorytm Gabowa
- Algorytm Gabowa-Tarjana

Skalowanie

Dany problem określony parametrami y_i.

- Rozwiąż problem dla $y_i' = \lfloor \frac{y_i}{2} \rfloor$.
- Korzystając z tego rozwiązania rozwiąż problem dla y_i.

Skalowanie

Dany problem określony parametrami y_i.

- Rozwiąż problem dla $y'_i = \lfloor \frac{y_i}{2} \rfloor$.
- Korzystając z tego rozwiązania rozwiąż problem dla y_i.

Rozwiązanie dla y_i jest podobne do rozwiązania dla y_i.

Metoda Węgierska

Metoda węgierska działa szybko je,żeli zaczniemy od dobrego pokrycia wierzchołkowego — wartości dualnych.

Metoda Węgierska

Metoda węgierska działa szybko je,żeli zaczniemy od dobrego pokrycia wierzchołkowego — wartości dualnych.

Zdefiniujmy

$$D = \sum_{i \in U \cup V} y_i - w(M^*),$$

gdzie y_i to wejściowe pokrycie wierzchołkowe, a M^* do najcięższe doskonałe skojarzenie.

Lemat o metodzie Węgierskiej

Niech f będzie liczbą wierzchołków wolnych w aktualnym skojarzeniu. Niech Δ_{tot} będzie sumą wartości Δ z wykonanych kroków metody Węgierskiej.

Lemma 1 *W każdym kroku mamy*

$$f\Delta_{tot} \leq D.$$

Dowód lematu

- Rozważmy wagę pokrycia wierzchołkowego $y(V) = \sum_{i \in U \cup V} y_i$.

Dowód lematu

- Rozważmy wagę pokrycia wierzchołkowego $y(V) = \sum_{i \in U \cup V} y_i$.
- Każda zmiana wartości dualnych zmienia $y(V)$ o co najmniej $g\Delta$, gdzie g to liczba wierzchołków wolnych kiedy zmiana miała miejsce.
  - jeżeli i jest wolny to y_i zmniejsza sie o Δ,
  - jeżeli i jest skojarzony to krawędzią ij to $y_i + y_j$ się nie zmienia.

Dowód lematu c.d.

- Liczbą wierzchołków wolnych może się tylko zmniejszyć — po znalezieniu ścieżki powiększającej, więc $g \geq f$.

Dowód lematu c.d.

- Liczbą wierzchołków wolnych może się tylko zmniejszyć — po znalezieniu ścieżki powiększającej, więc $g \geq f$.
- Całkowita zmiana $y(V)$ do aktualnego stanu algorytmu wynosi co najmniej

$$\sum g_i \Delta_i \geq f \sum \Delta_i = f \Delta_{tot}.$$

Dowód lematu c.d.

- Liczbą wierzchołków wolnych może się tylko zmniejszyć — po znalezieniu ścieżki powiększającej, więc $g \geq f$.
- Całkowita zmiana $y(V)$ do aktualnego stanu algorytmu wynosi co najmniej

$$\sum g_i \Delta_i \geq f \sum \Delta_i = f \Delta_{tot}.$$

- Z drugiej strony maksymalna całkowita zmiana $y(V)$ to $D = \sum_{i \in U \cup V} y_i - w(M^*)$, czyli $f \Delta_{tot} \leq D$.

Hopcroft Karp

Będziemy używać algorytmu Hopcrofta Karpa do powiększania skojarzenia.

- Algorytm otrzymuje na wejściu skojarzenie M od którego rozpoczynane będzie wyszukiwanie,
- a zwraca najliczniejsze skojarzenie w grafie.

Algorytm Gabowa

0. Jeżeli wszystkie $w_{ij} = 0$ to zwróć dowolne doskonałe skojarzenie i pokrycie wierzchołkowe $y_i = 0$.
1. Znajdź najcięższe doskonałe skojarzenie oraz pokrycie wierzchołkowe dla grafu $\overline{G}$ w którym $\overline{w}_{ij} = \lfloor \frac{w_{ij}}{2} \rfloor$.
2. Niech M będzie pustym skojarzeniem. Dla każdego $i \in U$ niech $y_i = 2y_i + 1$, a dla każdego $i \in V$ niech $y_i = 2y_i$.

Algorytm Gabowa II

3. Powtarzaj co następuje dopóki M nie jest doskonałym skojarzeniem:

3.1. Uruchom metodę węgierską aż znajdzie ścieżkę powiększającą.

3.2. Niech G_y będzie grafem równościowym dla y. Używając algorytmu Hopcrofta Karpa powiększ M do najliczniejszego skojarzenie w G_y.

4. Zwróć M i pokrycie wierzchołkowe y.

Algorytm Gabowa — Poprawność

Lemma 2 *Algorytm Gabowa zwraca najcięższe doskonałe skojarzenie i najlżejsze pokrycie wierzchołkowe.*

- Krok 0 jest poprawny.
- Po kroku 2 otrzymujemy poprawne pokrycie wierzchołkowe.
- W każdym wykonaniu kroku 3 liczność skojarzenie rośnie.

Alg. Gabowa — Czas działania

W trakcie wykonania kroku 3 zachodzi

$$f\Delta_{tot} \leq D.$$

Przeplot z HK nic nie psuje.

Pokażemy że

Lemma 3

(i) Krok 3 wykonany jest mniej niż $n^{\frac{1}{2}}$ razy.

(ii) Krok 3 wykonany jest co najwyżej $n^{\frac{1}{4}}$ razy dla $f \geq n^{\frac{3}{4}}$.

Alg. Gabowa — Czas działania

Niech $\overline{M}^*$ to najcięższe skojarzenie w $\overline{G}$, a M^* to najcięższe skojarzenie w G.

Wtedy

$$D = \sum_i y_i - w_G(M^*) \leq n$$

bo

$$w_G(M^*) \geq w_G(\overline{M}^*) \geq 2w_{\overline{G}}(\overline{M}^*) = 2\sum_i \overline{y}_i = \sum_i y_i - n.$$

Alg. Gabowa — Czas działania

Lemma 4 (i) *Krok 3 wykonany jest mniej niż $n^{\frac{1}{2}}$ razy.*

Policzmy ilość kroków wykonanych dla:

- $f \geq n^{\frac{1}{2}}$ — z lematu mamy $\Delta_{tot} \leq \frac{D}{f} \leq n^{\frac{1}{2}}$. Ponieważ w każdym kroku $\Delta > 0$, bo nie ma ścieżek powiększających zawierających się w grafie równościowym. Takich kroków jest $\leq n^{\frac{1}{2}}$.
- $f < n^{\frac{1}{2}}$ — ponieważ każdy krok kojarzy co najmniej dwa wierzchołki, wiec tych kroków jest $\leq n^{\frac{1}{2}}$.

Alg. Gabowa — Czas działania

Lemma 5 (ii) *Krok 3 wykonany jest co najwyżej* $n^{\frac{1}{4}}$ *razy dla* $f \geq n^{\frac{3}{4}}$.

Jeżeli $f \geq n^{\frac{3}{4}}$ to wtedy $\Delta_{tot} \leq \frac{D}{f} \leq n^{\frac{1}{4}}$. Ponieważ $\Delta \geq 1$ to liczba kroków jest mniejsza niż $n^{\frac{1}{4}}$.

Alg. Gabowa — Czas działania

Lemma 6 (i) *Krok 3 wykonany jest mniej niż* $n^{\frac{1}{2}}$ *razy.*

Każde uruchomienie metody węgierskiej zajmuje $O(m)$ czasu czyli wszystkie uruchomienia zajmują co najwyżej $O(n^{\frac{1}{2}}m)$ czasu.

Alg. Gabowa — Czas działania

Lemma 7 (ii) *Krok 3 wykonany jest co najwyżej* $n^{\frac{1}{4}}$ *razy dla* $f \geq n^{\frac{3}{4}}$.

Algorytm HK działa w czasie $O(\min(n^{\frac{1}{2}}, a)m)$, gdzie a to liczba znalezionych ścieżek powiększających. Na podstawie *(ii)*

- kroki z $f \geq n^{\frac{3}{4}}$ zajmują co najwyżej $O(n^{\frac{1}{4}} n^{\frac{1}{2}} m)$ czasu.
- kroki z $f < n^{\frac{3}{4}}$ znajdą co najwyżej $\frac{n^{\frac{3}{4}}}{2}$ ścieżek powiększających wiec zajmą $O(n^{\frac{3}{4}} m)$ czasu.

Alg. Gabowa — Czas działania

Liczba wywołań rekurencyjnych to $\lfloor \log W \rfloor + 2$.

Theorem 1 *Najcięższe doskonałe skojarzenie w grafie dwudzielnym może zostać znalezione w czasie* $O(n^{\frac{3}{4}} m \log W)$.

Liczby w zapisie bitowym pokrycia wierzchołkowego są długości $O(\log(nW))$.

Alg. Gabowa — Czas działania

W każdym z $\lfloor \log W \rfloor$ kroków wartość pokrycia zmienia się maksymalnie o $\leq \Delta \leq D \leq n$. Czyli w i-tym kroku wyniki

$$a_{i+1} \leq 2a_i + 1 + n,$$

czyli wynosi najwyżej

$$\left(2^{\lfloor \log W \rfloor + 1} - 1\right)(n+1) \leq W(2n+2).$$

Alg. Gabowa-Tarjana

- Metoda Węgierska — najlżejsze ścieżki.
- Algorytm HK — najkrótsze ścieżki.

Jak znajdować jednocześnie wiele rozłącznych wierzchołkowo ścieżek powiększających, które byłyby jednocześnie krótkie i miały małą wagę?

Najlżejsze skojarzenia

Będziemy szukać najlżejszych doskonałych skojarzeń.

Po zmianie znaków wszystkich wag, najlżejsze doskonałe skojarzenia odpowiadają najcięższym doskonałym skojarzeniom w grafie wejściowym.

Najmniejsze pokrycie wierzchołkowe to najcięższe pokrycie wierzchołkowe takie, że dla każdej krawędzi vw mamy $y_v + y_w \leq w_{vw}$.

Definicje

1-dozwolone skojarzenie to skojarzenie M wraz z wartościami pokrycia wierzchołkowego y, takimi, że dla każdej krawędzi uv zachodzi,

$$y_v + y_w \leq w_{vw} + 1,$$

$$y_v + y_w = w_{vw}, \text{ jezeli } vw \in M.$$

1-optymalne skojarzenie to doskonałe

skojarzenie, które jest 1-dozwolone.

Właściwości

Lemma 8 *Niech M będzie 1-optymalnym skojarzeniem*

(a) *Dla dowolnego skojarzenia P zachodzi* $w(P) \geq w(M) - n$.

(b) *Jeżeli pewne k, $k > n$, dzieli wszystkie koszty $w(e)$, to wtedy M jest najlżejszym doskonałym skojarzeniem.*

Właściwości

(a)

$$w(M) = \sum_{uv \in M} w(uv) = \sum_{v \in U \cup V} y(v) \leq$$

$$\leq \sum_{uv \in P} w(uv) + 1 = w(P) + n.$$

(b) Niech M^* to minimalne skojarzenie, mamy wtedy

$$w(M^*) \leq w(M) \leq w(M^*) + n,$$

czyli $w(M) = w(M^*)$.

Algorytm Gabowa-Tarjana

- Niech $\overline{w}(uw) = w(uw) \times (n+1)$.
- Ustaw $w(e) = 0$, oraz $y(v) = 0$.
- Dla i od 1 do k wykonuj:
  - dla każdej krawędzi $w(e) = 2w(e) + (i\text{–ty bit } \overline{w}(e))$,
  - dla każdego wierzchołka $y(v) = 2y(v) - 1$,
  - znajdź 1-optymalne skojarzenie przy użyciu procedury `scale_match`.

Procedura scale_match

Procedura `scale_match`

- zmienia koszty na

$$w(uv) := w(uv) - y(v) - y(u).$$

- wywołuje procedure `match` do znalezienia 1-optymalnego skojarzenia M i pokrycia y^*,
- dodaje y^* do y.
- zwraca M i y.

M i y to 1-optymalne skojarzenie dla w.

Procedura scale_match

Przed skalowaniem mamy

$$y_v + y_w \leq w_{vw} + 1,$$

Po skalowaniu mamy,

$$y'_v + y'_w = 2y_v - 1 + 2y_w - 1 \leq$$

$$\leq 2w_{vw} \leq 2w_{vw} + (i\text{--ty bit } \overline{w}_{vw}) = w'_{vw}.$$

Czyli puste skojarzenie jest 1-dozwolone oraz

$$w'_{uv} - y'_v - y'_w \geq 0.$$

Procedura scale_match

Wartości y_i przekazane do `match` są ≥ 0.

Z drugiej strony jeżeli uv jest krawędzią 1-optymalnego skojarzenia M to po przeskalowaniu

$$y'(u) + y'(v) = 2y(u) + 2y(v) - 2 =$$
$$= 2w(uv) - 2 = w'(uv) - (i\text{–ty bit } \overline{w}_{uv}) - 2 \geq$$
$$\geq w'(uv) - 3.$$

Procedura match

Mamy więc $w'(uv) - y'(u) - y'(v) \leq 3$ i waga M w wagach po przeskalowaniu wynosi co najwyżej $3n$, czyli najlżejsze skojarzenie waży $\leq 3n$.

Pokażemy teraz procedure `match` która przy założeniach, że wagi ≥ 0 oraz, że istnieje skojarzenie o wadze $\leq 3n$ znajduje najlżejsze doskonałe skojarzenie w czasie $O(\sqrt{n}m)$.

Procedura match — definicje

Wago-długość krawędzi e względem skojarzenia M niech będzie

$$wd(e) = w(e) + \begin{cases} 1 & \text{jeśli } w \notin M \\ 0 & \text{wpp} \end{cases}$$

Całkowita wago-długość zbioru krawędzi S względem M to

$$wd(S) = \sum_{e \in S-M} wd(e) - \sum_{e \in S \cap M} wd(e),$$

to waga S plus liczba nieskojarzonych krawędzi.

Procedura match — definicje

Powiemy, że krawędź uv jest dozwolona jeżeli $y(u) + y(v) = wd(uv)$, tzn. warunek 1-dozwoloności zachodzi z równością.

Krawędzie skojarzone są dozwolone.

Pokażemy, że ścieżki powiększające składające się z dozwolonych krawędzi mają najmniejszą całkowitą wago-długość.

Procedura match

Zainicjalizuj $y_v = 0$ oraz $M = \emptyset$.

Powtarzaj dopóki M nie jest doskonałe

1. Znajdź maksymalny zbiór $\mathcal{A}$ wierzchołkowo rozłącznych ścieżek powiększających. Dla każdej $P \in \mathcal{A}$, powiększ M względem P, oraz dla każdego wierzchołka $w \in V \cap P$ zmniejsz $y(w)$ o 1.
2. Użyj metody węgierskiej do zmiany wartości dualnych (zachowując 1-dozwoloność) i znajdź ścieżkę powiększającą z dozwolonych krawędzi.

Procedura match — poprawność

Kroki procedury match zachowują 1-dozwoloność.

Każde wywołanie metody węgierskiej tworzy ścieżkę powiększającą, która następnie zostanie użyta do powiększenia skojarzenia. Czyli ostatecznie doskonałe skojarzenie zostanie znalezione.

Procedure `match` kończy działanie znajdując 1-dozwolone skojarzenie.

Procedura match — czas działania

Dla dowolnego kroku algorytmu zdefiniujmy

- F to zbiór wierzchołków wolnych w U,
- Δ_{tot} suma wszystkich wartości Δ z kroków metody Węgierskiej.

Zauważmy, że zmiany wartości dualnych są takie, że dla $v \in F$ mamy $y(v) = \Delta_{tot}$, oraz dla wolnego wierzchołka w V mamy $y(v) = 0$.

Procedura match — czas działania

Niech M^* będzie najlżejszym doskonałym skojarzeniem, a M aktualnym skojarzeniem. Zbiór $M \oplus M^*$ składa się z ścieżek powiększających P_v dla każdego $v \in F$, oraz zbioru cykli naprzemiennych C_w.

$$n + w(M^*) - w(M) \geq wd(M^* \oplus M) =$$

$$= \sum_{v \in F} wd(P_v) + \sum_{w} wd(C_w).$$

Procedura match — czas działania

Oszacujmy lewą stronę tej nierówności. Rozważmy ścieżkę naprzemienną od $u \in U$ do $m \in U$, w której u należy do krawędzi nieskojarzonej a m do krawędzi skojarzonej. Wtedy dla krawędzi $vw \notin M$

$$y(v) + y(w) \leq wd(vw),$$

a dla $wn \in M$

$$y(w) + y(n) = wd(wn),$$

więc

$$y(v) \leq y(n) + wd(vw) - wd(wm).$$

Procedura match — czas działania

Korzystając z tej nierówności dla wszystkich par krawędzi na P otrzymujemy

$$y(u) \leq y(m) + wd(P).$$

Z nierówności tej wynika że dla każdego naprzemiennego cyklu C_w mamy

$$wd(C_w) \geq 0.$$

Oraz dla ścieżki powiększającej z $v \in F$ do $t \in V$ mamy

$$y(v) + y(t) \leq wd(P_v).$$

Procedura match — czas działania

Ponieważ metoda węgierska utrzymuje $y(v) = \Delta_{tot}$ oraz $y(t) = 0$, więc $\Delta_{tot} \leq wd(P_v)$ i

$$\sum_{v \in F} wd(P_v) + \sum_{w} wd(C_w) \geq |F|\Delta_{tot}.$$

Ponieważ $c(M^*) \leq 3n$ oraz $c(M) \geq 0$. Więc

$$4n \geq n + w(M^*) - w(M).$$

Ostatecznie otrzymujemy

$$4n \geq |F|\Delta_{tot}.$$

Procedura match — czas działania

Jeżeli pokażemy, że każde wykonanie metody węgierskiej zwiększa Δ_{tot}, to

- Co najwyżej $2\sqrt{n}+1$ iteracji jest wykonanych dla $|F| \leq 2\sqrt{n}$, bo po każdym wykonaniu metody węgierskiej skojarzenie się powiększa.
- Jeżeli $|F| > 2\sqrt{n}$ to $\Delta_{tot} < 2\sqrt{n}$ i takich iteracji też będzie co najwyżej $2\sqrt{n}$.

Procedura match — czas działania

Aby pokazać, że każde wywołanie metody Węgierskiej zwiększa Δ_{tot} wystarczy pokazać, że zmienia ono wartości dualne.

Mogłoby ich nie zmienić tylko wtedy gdyby istniała przed jej wykonaniem naprzemienna ścieżka P składająca się z krawędzi dozwolonych.

Procedura match — czas działania

Ścieżka P przecina się ze ścieżką znalezioną w kroku 1 procedury `match`. P zawiera

nieskojarzoną krawędź vw taką, że w należy do ścieżki z kroku 1, a v nie, oraz $w \in V$.

Ale po kroku 1 krawędź vw nie może być dozwolona bo y_w jest zmniejszone,

$$y_v + y_w \leq w_{vw} + 1.$$

Procedura match — czas działania

Każde wykonanie pętli w procedurze `match` zajmuje $O(m)$ czasu.

- Wyszukiwanie maksymalnego zbioru dozwolonych ścieżek zajmuje $O(m)$ — tak jak algorytm HK.
- Metoda Węgierska może być wykonana w czasie $O(m + n \log n)$ tak jak na ćwiczeniach tydzień temu. Można ten czas zmienić na $O(m)$ zauważając, że wartości w kopcu są z przedziału $1, \ldots, 4n$ i zawsze rosną — kopiec w tablicy.