Algorytmy aproksymacyjne dla problemów stochastycznych

Piotr Sankowski

Uniwersytet Warszawski

PhD Open, 5-6 grudzień, 2008

Plan - Wykład III

- Aproksymacyjne algorytmy online
- Aproksymacyjne stochastyczne algorytmy online
- Problem drzewa Steinera online
  - dolne ograniczenia
  - górne ograniczenia
- Stochastyczny problem drzewa Steinera online
  - dolne ograniczenia
  - górne ograniczenia

Algorytmy online

W algorytmach online chcemy rozwiązywać problemy, w których dane wejściowe ujawniane są krok po kroku.

Zazwyczaj zakłada się, że nasze decyzje są nieodwracalne.

Algorytmy dynamiczne wpp.

Badamy wpływ niewiedzy o przyszłości, a nie czas działania.

Interesują nas też algorytmy wykładnicze.

Algorytmy online

Zadaniem *algorytmu online* jest przetworzenie nieprzewidywalnej sekwencji żądań, wykonując każde z nich bez wiedzy o przyszłości.

W *analizie kompetytywnej* algorytmów online badamy zachowanie:

- algorytmu online, który nie zna przyszłości,

porównując go do:

- optymalnego algorytmu offline, który zna całą sekwencję żądań.

Współczynnik kompetytywności

Zakładamy, że sekwencja żądań pochodzi z uniwersum Ω, oraz, że przeciwnik wybiera sekencję $\omega = \omega_1, \omega_2, \ldots$ nieznanej długości. Niech Ω^l oznacza zbiór sekwencji długości l.

Klasyczny *współczynnik kompetytywności* (randomizowanego) algorytmu $\mathcal{A}$ zdefiniowany jest jako:

$$\max_{k} \max_{\omega \in \Omega_k} \frac{\mathbf{E}_r[\mathcal{A}(\omega, r)]}{\mathsf{OPT}(\omega)},$$

gdzie r jest zbiorem losowych wyborów algorytmu.

Drzewo Steinera online

W problemie *online drzewa Steinera* mamy dany:

- graf $G = (V, E)$ o zadanym korzeniu r oraz koszty krawędzi $c : E \to_{\geq 0}$;
- b.s.o. koszty spełniają nierówność trójkąta;
- przeciwnik wybiera sekwencję wierzchołków (z powtórzeniami) $v_1, v_2, \ldots$ z V;
- w każdym momencie czasu t, utrzymujemy spójny podgraf $S_t \subseteq G$ zawierający r oraz wszystkie dotychczas ujawnione wierzchołki $\{v_1, \ldots, v_t\}$;
- zakładamy, że decyzje algorytmu są nieodwracalne, a więc $S_t \subseteq S_{t+1}$.

Algorytm zachłanny

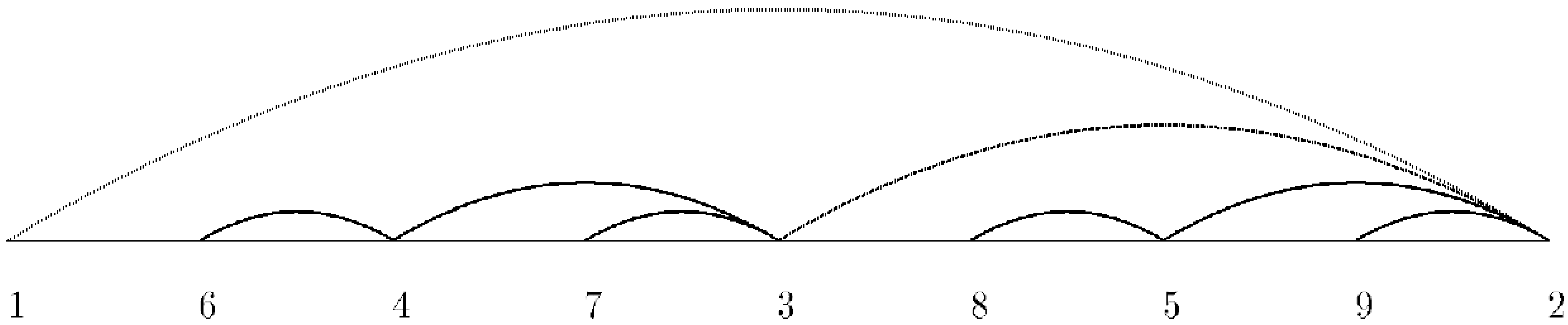
Rozwiązanie zachłanne zawsze podłącza kolejny otrzymany wierzchołek z aktualnie wykupionym podgrafem przy pomocy ścieżki o najmniejszym koszcie.

Twierdzenie 1 (Imase and Waxman '91) *Współczynnik kompetytywności dla dowolnego algorytmu dla drzewa Steinera online wynosi* $\Omega(\log n)$.

Twierdzenie 2 (Imase and Waxman '91) *Współczynnik kompetytywności dla zachłannego algorytmu dla drzewa Steinera online wynosi* $O(\log n)$.

Algorytm zachłanny

Prosty przykład pokazuje, że algorytm zachłanny płaci co najmniej $\log n$.



Pierwsze wierzchołki w złej sekwencji to skrajny lewy i skrajny prawy, a następne to wierzchołki z środka.

Dolne ograniczenie

W przypadku dowolnych algorytmów.

Zdefiniujmy grafy $G_k = (V_k, E_k)$, dla $k \in Z_0^+$ o funkcji kosztu c_k.

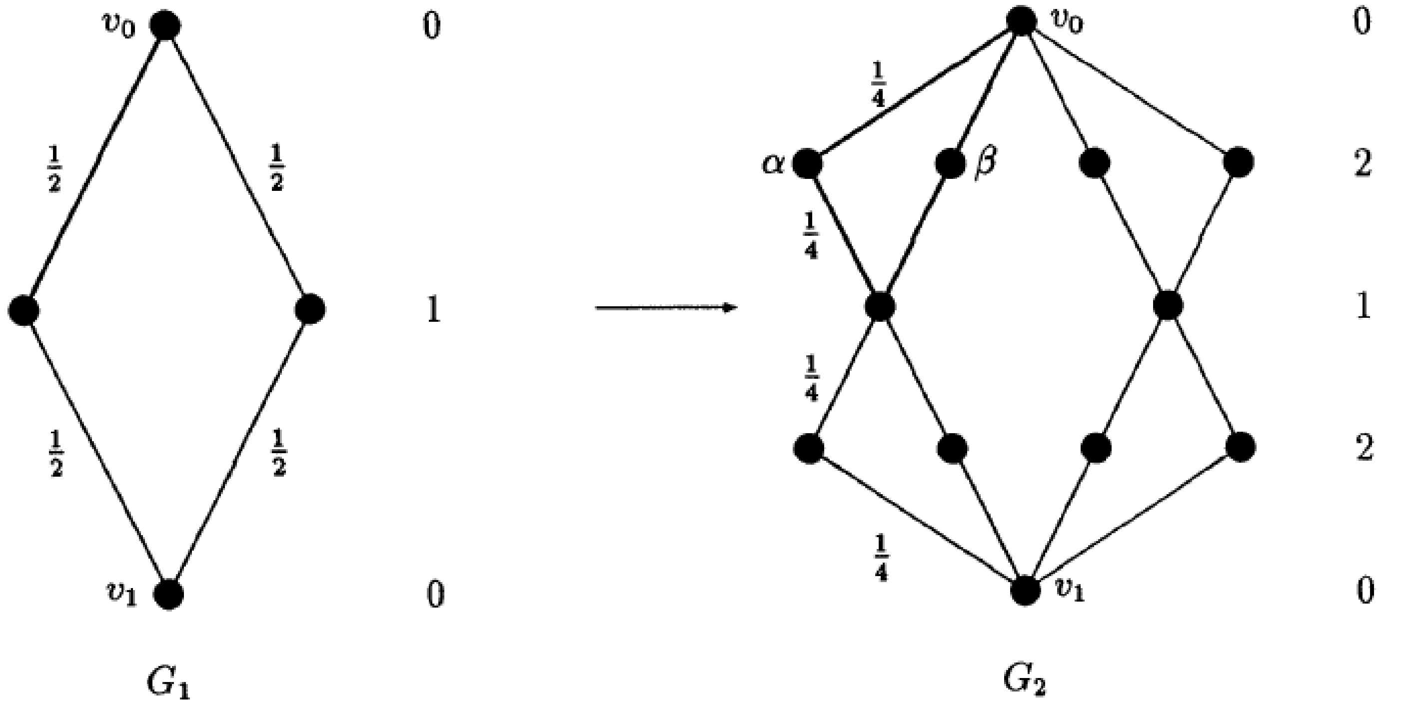
G_0 jest grafem o dwóch wierzchołkach i jednej krawędzi o koszcie 1.

Aby otrzymać G_k każdą krawędź (u, v) w G_{k-1} zamieniamy na dwie ścieżki (u, α, v) oraz (u, β, v) o krawędziach o wadze 2^{-k}.

Wierzchołki dodane w G_k mają poziom k.

Dolne ograniczenie

Sekwencję zaczynamy zadając v_0 i v_1.



Następnie dla każdego poziomu wybieramy wierzchołki jeszcze nie połączone z par α i β.

Dolne ograniczenie

Niech N_i to ciąg żądań na poziomie i.

Niech $T = \{T_0, T_1, \ldots, T_k\}$ będzie sekwencją drzew wygenerowanych przez algorytm dla ciągu żądań $N = \{N_0, N_1, \ldots, N_k\}$.

Niech $\hat{T} = \{\hat{T}_0, \hat{T}_1, \ldots, \hat{T}_k\}$ będzie sekwencją minimalnych drzew otrzymaną z T, i.e., $\hat{T}_i$ jest minimalnym w sensie zawierania drzewem łączącym aktualne wierzchołki.

Pokażemy, że $c(\hat{T}_i) \geq c(\hat{T}_{i-1}) + \frac{1}{2}$, a zatem $c(\hat{T}_i) \geq \frac{1}{2}i + 1$.

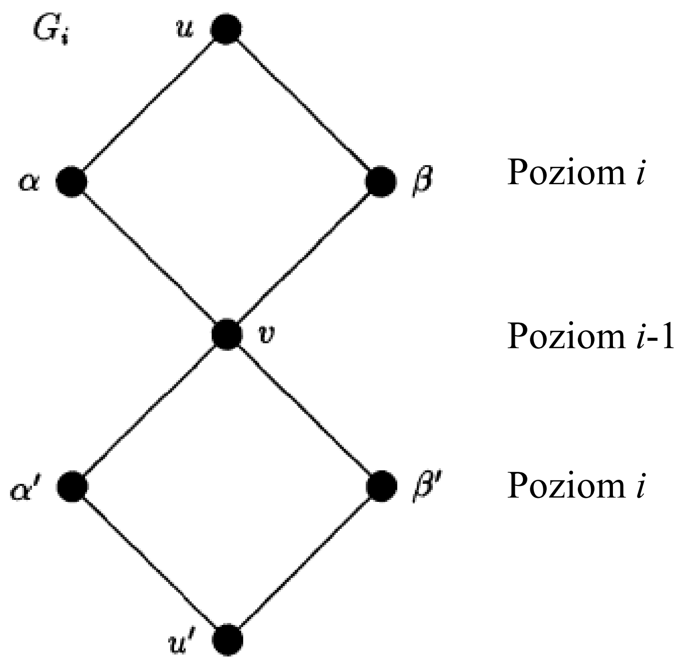
Dolne ograniczenie

Ponieważ $\hat{T}_{i-1}$ jest minimalne to nie zawiera cykli oraz każdy liść musi być elementem jakiegoś N_j.

Rozważmy wierzchołek v z N_{i-1}, jest on sąsiedni z czterema wierzchołkami poziomu i.

Z każdej z sąsiednich par w $\hat{T}_{i-1}$ może być tylko jeden wierzchołek, gdyż inaczej mielibyśmy:

- liść na poziomie i,
- bądź cykl.



Dolne ograniczenie

Dla każdego wierzchołka $v \in N_{i-1}$ wybieramy do N_i sąsiednie wierzchołki z poziomu i które nie należą do $\hat{T}_{i-1}$.

Zauważmy, że jak istniała wcześniej ścieżka o koszcie 1 zawierająca wszystkie żądania, to istnieje ona nadal.

W N_i jest 2^{i-1} wierzchołków i za podłączenie każdego musimy zapłacić co najmniej 2^{-i}.

Tak więc $c(\hat{T}_i) \geq c(\hat{T}_{i-1}) + \frac{1}{2}$.

Algorytm zachłanny

Udowodniliśmy więc pierwsze z twierdzeń.

Twierdzenie 3 (Imase and Waxman '91) *Współczynnik kompetytywności dla dowolnego algorytmu dla drzewa Steinera online wynosi* $\Omega(\log n)$.

Przejdźmy teraz do drugiego dowodu.

Twierdzenie 4 (Imase and Waxman '91) *Współczynnik kompetytywności dla zachłannego algorytmu dla drzewa Steinera online wynosi* $O(\log n)$.

Górne ograniczenie

Aby pokazać górne ograniczenie założymy, że n jest parzyste.

Niech T będzie optymalnym drzewem Steinera.

Rozważmy cykl Eulera E otrzymany z drzewa T, jego koszt wynosi co najwyżej $2c(T)$.

W cyklu tym korzystamy ze wszystkich możliwych skrótów.

Długość E jest parzysta, a więc możemy rozbić go na dwa skojarzenia. Niech M oznacza to które waży mniej niż $c(T)$.

Górne ograniczenie

Rozważmy $(u, v) \in M$. Bez starty ogólności możemy założyć, że u występuje w sekwencji przez v.

Po ujawnieniu v mogliśmy wybrać krawędź (u, v).

Taką możliwość mamy dla połowy wierzchołków, to połączenie tej połowy wierzchołków nie kosztuje więcej niż koszt skojarzenia.

Możemy rozważyć pozostałe $\frac{n}{2}$ wierzchołków i zastosować ten sam argument.

Po $\log n$ powtórzeniach zostanie nam jeden wierzchołek.

Algorytmy stochastyczne online

W problemie stochastycznym online:

- algorytm ma dany jako wejście rozkład prawdopodobieństwa,
- jesteśmy zainteresowani oczekiwanym kosztem algorytmu.

Taki model został dobrze przebadany w przypadki problemów takich jak stronicowanie, czy problem k-serwerów.

Natomiast nie wiadomo wiele o problemach optymalizacji kombinatorycznej, np. drzewie Steinera.

Algorytmy stochastyczne online

W przypadki stochastyczmym adwersarz może wybrać rozkład prawdopodobieństwa $\pi : \Omega \rightarrow [0, 1]$ nad przestrzenią żądań oraz długość sekwencji żądań k.

Sekwencja ω będzie wygenerowana przez k'krotne niezależne losowanie z π.

Będziemy rozważać przypadki kiedy algorytm może znać k bądź π.

Stochastatyczna kompetytywność

W przypadku stochastycznym możemy rozważać dwie miary zachowania algorytmów.

- *ratio-of-expectations* (RoE):

$$\mathsf{RoE}(\mathsf{ALG}) = \max_{\pi} \max_{k} \frac{\mathbf{E}_{\omega \in \pi^k, r}[\mathsf{ALG}(\omega, r)]}{\mathbf{E}_{\omega \in \pi^k}[\mathsf{OPT}(\omega)]}.$$

- *expectation-of-ratios* (EoR):

$$\mathsf{RoE}(\mathsf{ALG}) = \max_{\pi} \max_{k} \mathbf{E}_{\omega \in \pi^k, r} \left[\frac{\mathsf{ALG}(\omega, r)}{\mathsf{OPT}(\omega)} \right].$$

Stochatyczna kompetytywność

Współczynniki te są nieporównywalne ale EoR wydaje się być trudniejszy.

Rozważmy $L+1$ scenariuszy.

$$\begin{aligned}\text{RoE} &= \frac{L\times 1 + 1\times 2L^2}{L\times 1 + L} \\ &= \frac{1+2L}{2} \geq L, \\ \text{EoR} &= \frac{1}{L+1}\left(L\times\frac{1}{1} + \frac{L^2}{L}\right) \\ &= \frac{2L}{L+1} \leq 2.\end{aligned}$$

Rozważmy $2L$ scenariuszy.

$$\begin{aligned}\text{RoE} &= \frac{L\times L + L\times 2L}{L\times L + L} \\ &= \frac{2L^2}{L^2+L} \leq 2, \\ \text{EoR} &= \frac{1}{2L}\left(L\times\frac{L}{L} + L\times\frac{2L}{1}\right) \\ &= \frac{L+2L^2}{2L} \geq L.\end{aligned}$$

Stochastyczne d. S. online

W *stochastycznym problemie drzewa Steinera online* mamy dany:

- graf $G = (V, E)$ o zadanym korzeniu r, oraz koszty krawędzi $c : E \longrightarrow_{\geq 0}$;
- przeciwnik wybiera rozkład prawdopodobieństwa π na V oraz k;
- sekwencja to k wierzchołków wylosowanych niezależnie z π;
- w momencie czasu t utrzymujemy spójny podgraf $S_t \subseteq G$ zawierający r oraz $\{v_1, \ldots, v_t\}$;
- decyzje algorytmu są nieodwracalne $S_t \subseteq S_{t+1}$.

Drzewo Steinera

Twierdzenie 5 (Drzewo Steiner RoE) *Istnieje stochastyczny algorytm online dla problemu drzewa Steinera taki, że* $\mathsf{RoE}(\mathcal{A}) = O(1)$*, gdy jako wejście dane jest* π*.*

Wynik ten można uogólnić także do innych problemów

Twierdzenie 6 *Istnieją stochastyczne algorytmy online dla problemów lasu Steinera, lokalizacji fabryk oraz pokrycia wierzchołkowego takie, że* $\mathsf{RoE} = O(1)$*, gdy jako wejście dane jest* π*.*

Dolne ograniczenia

Można pokazać, że wiedza o rozkładzie, oraz niezależność są konieczne tzn.:

- jeżeli wejście jest otrzymane poprzez niezależne losowania z nieznanego rozkładu,
- bądź losowania nie są niezależne, ale zadane są przez łańcuch Markova,

to istnieje dolne ograniczenie $\Omega(\frac{\log n}{\log \log n})$ w przypadku drzewa Steinera.

Drzewo Steinera cd.

W przypadku gdy algorytm zna k oraz π, to:

Twierdzenie 7 *Istnieje stochastyczny algorytm online dla drzewa Steinera taki, że* $\mathsf{EoR} = O(\log\log n)$.

Otwarty problem: uogólnić ten wynik na przypadek dowolnego k.

Algorytm zachłanny

W przypadku klasycznym najlepsze rozwiązanie to algorytm zachłanny, w którym:

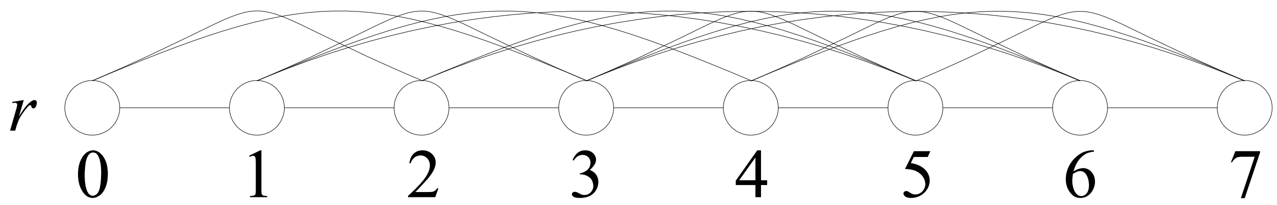
- każde żądanie łączymy najkrótszą ścieżką.

Algorytm ten jest $\Theta(\log n)$ kompetytywny.

Pokażemy teraz, że pozostaje on $\Omega(\log n)$ kompetytywny także w przypadku stochastycznym.

Algorytm zachłanny

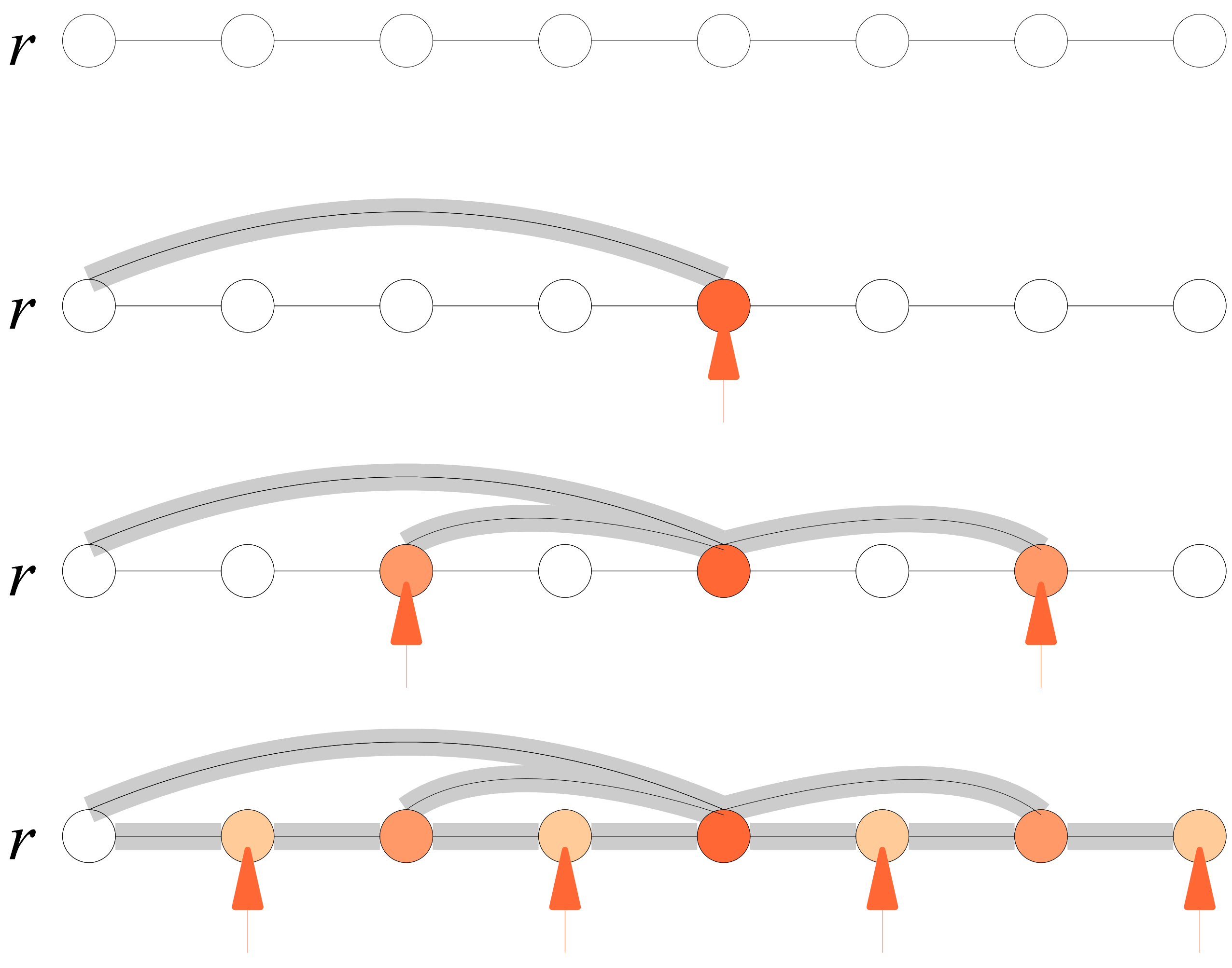
Rozważmy ścieżkę długości n złożoną z krawędzi o koszcie 1, o wierzchołkach ponumerowanych $0, 1, \ldots, n$, wierzchołek 0 jest korzeniem oraz dla pozostałych wierzchołków $\pi(i) = 1/n$.



Dla każdej pary niesąsiednich wierzchołków $\{i, j\}$ dodajmy krawędź o wadze $\ell_{ij} = |i - j| - (i - j)^2/n^3$.

Taka waga krawędzi gwarantuje, że najkrótsza ścieżka to zawsze krawędź „skrót".

Algorytm zachłanny



Algorytm zachłanny

Wybierając wierzchołki z π wiemy, że:

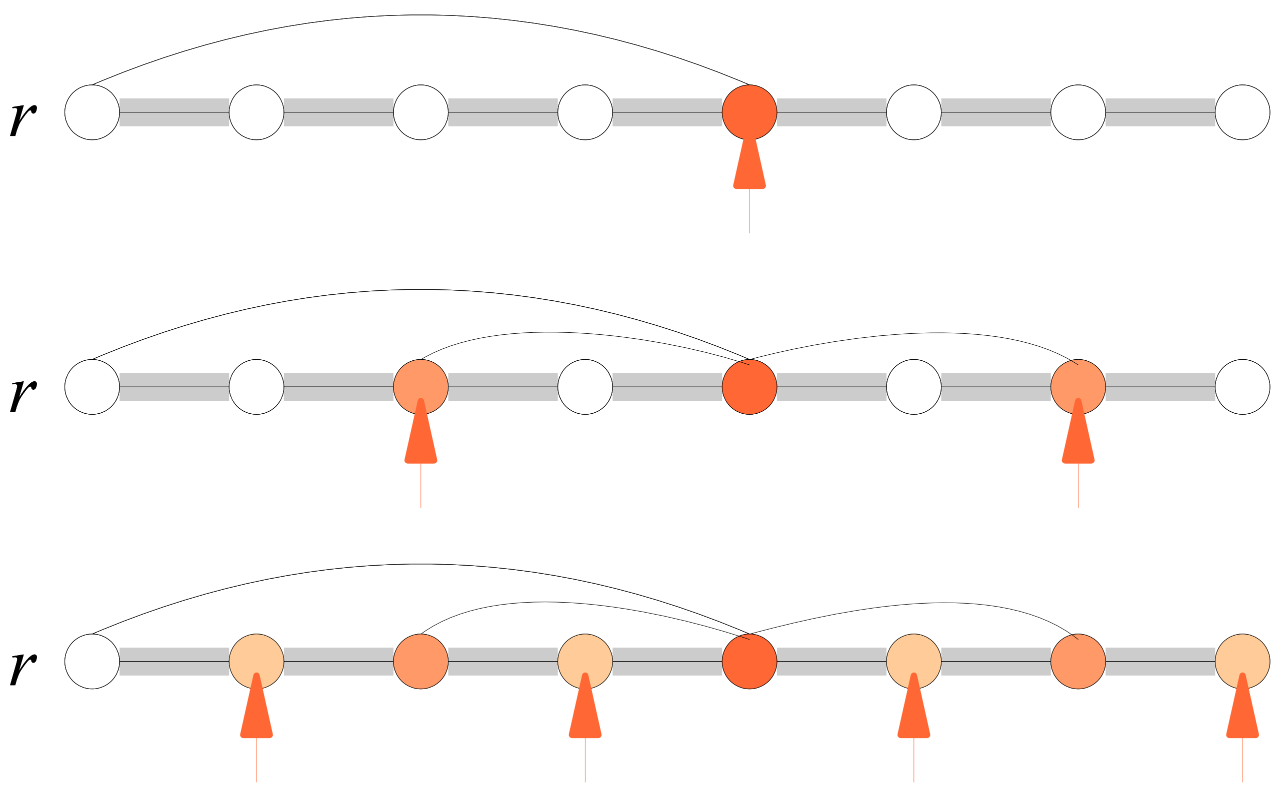
- oczekiwana odległość t'tego wierzchołka do innego najbliższego wynosi około $n/2t$.
- algorytm zachłanny zawsze wykupuje krawędź „skrót".

Otrzymujemy więc oczekiwany koszt $\Theta(n \log k)$ dla k wierzchołków.

Nawet zupełnie naiwna strategia wykupienia całej ścieżki po otrzymaniu pierwszego wierzchołka spisuje się lepiej tzn.,

- płaci ona n, co daje $\mathsf{RoE} = n/(n/2) = 2$.

Algorytm naiwny



Jak to zrobić lepiej

Jeżeli k jest znane to dobrym rozwiązaniem jest wygenerowanie losowo k wierzchołków z rozkładu i zbudowanie na nich dobrego rozwiązania.

Dla przypadku ścieżki, każdy wierzchołek z sekwencji wejściowej ma wierzchołek w oczekiwanej odległości $n/2k$, co daje dodatkowy oczekiwany koszt $n/2$ za podłączenie k wierzchołków.

Najpierw pokażemy algorytm dla zadanego k, a potem przejdziemy do przypadku gdy k jest nieznane.

EoR **dla znanego** k

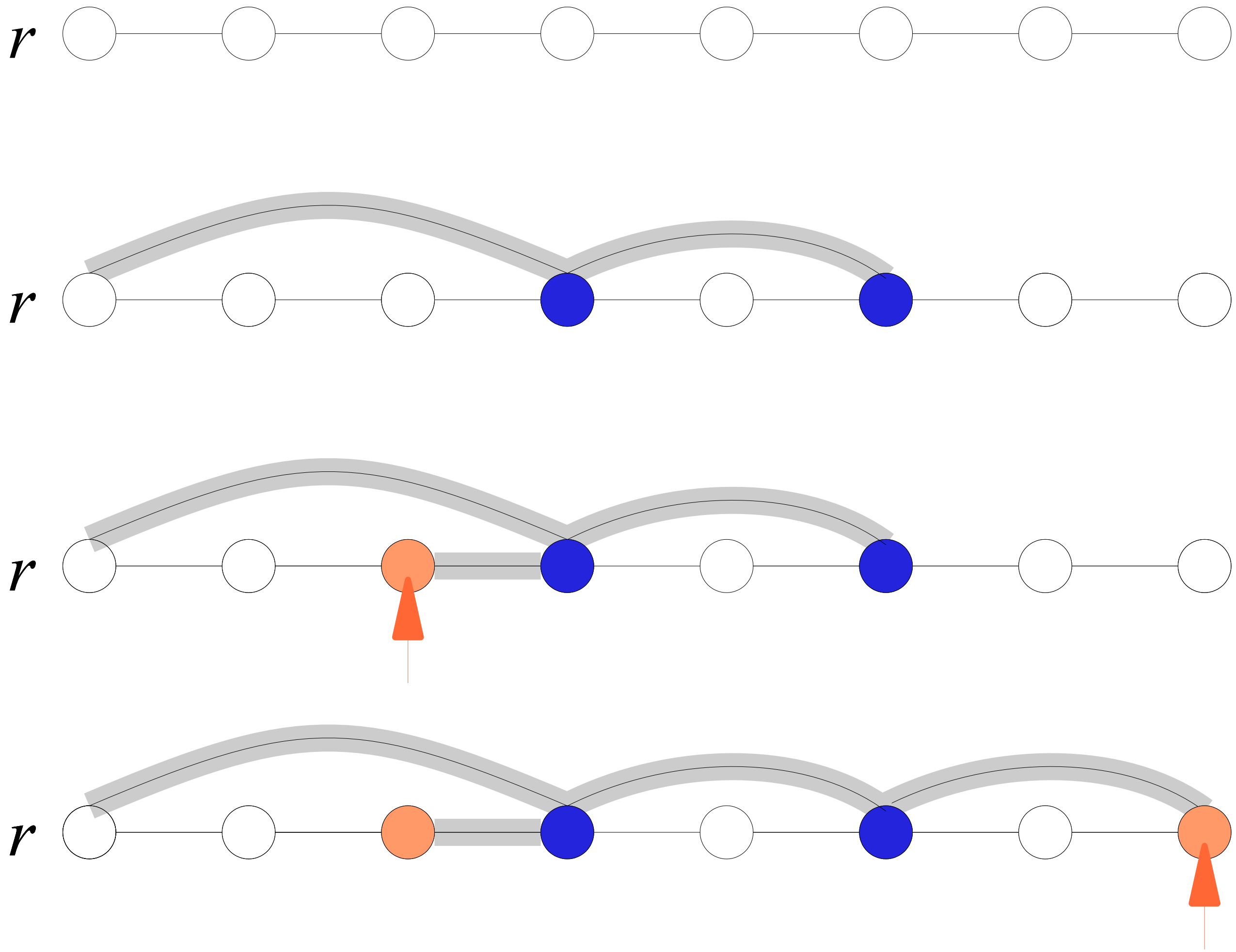
Załóżmy, że długość sekwencji żądań k jest znana. Rozważmy następujący algorytm:

A1. Wybierz zbiór D losując niezależnie k razy z rozkładu π.

A2. Skonstruuj 2-aproksymacyjne drzewo Steinera T_M na zbiorze $D \cup \{r\}$.

A3. Na ujawnionej sekwencji uruchom *algorytm zachłanny* — mianowicie, podłącz każdy wierzchołek do najbliższego w aktualnym rozwiązaniu.

EoR **dla znanego** k



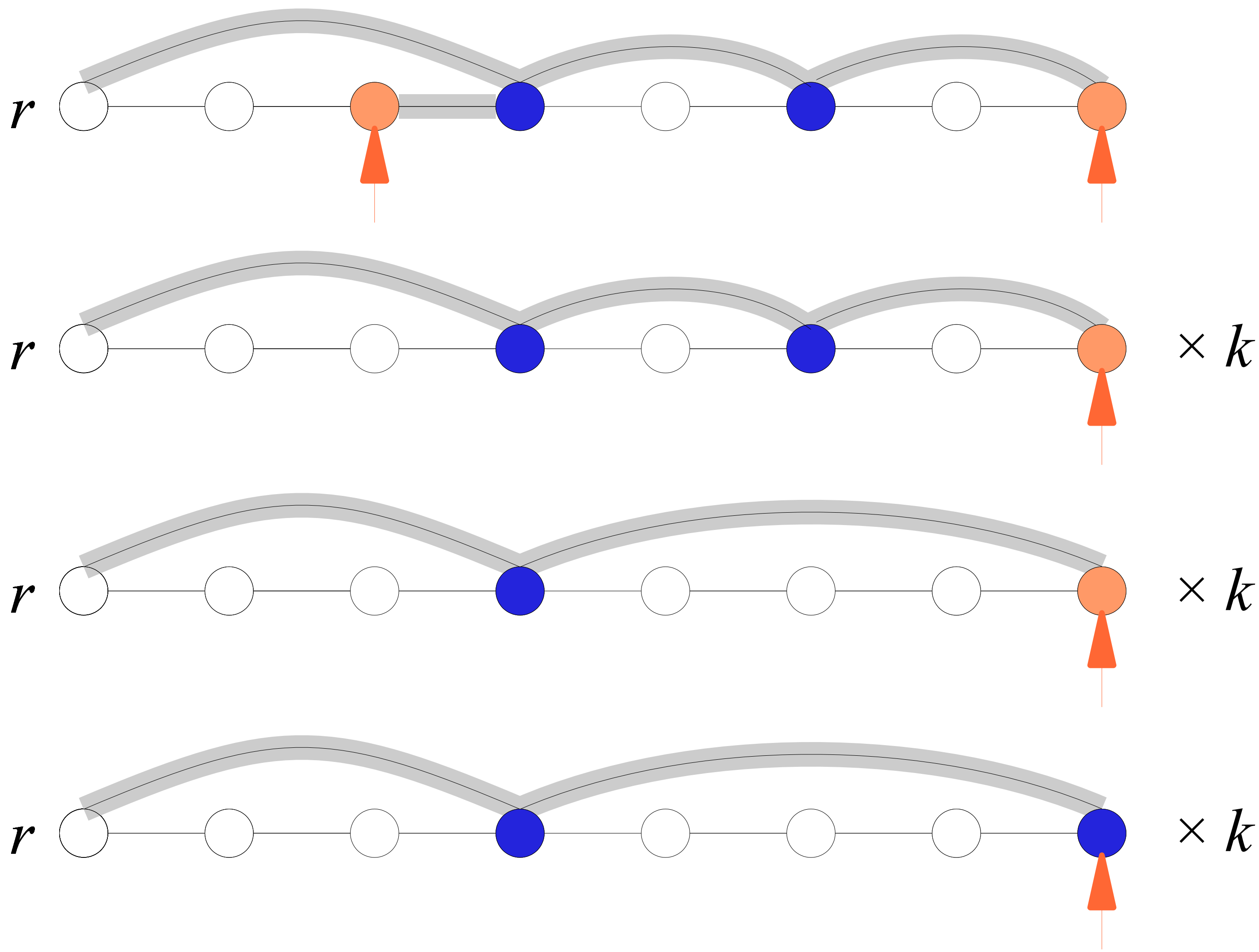
EoR **dla znanego** k

Twierdzenie 8 (EoR dla znanego k) *Współczynnik aproksymacji* RoE *dla powyższego algorytmu wynosi* 4.

Oczekiwany koszt optymalnego drzewa Steinera dla D wynosi $\mathbf{E}[\mathsf{OPT}(\omega)]$, a zatem oczekiwany koszt drzewa T_M wynosi $2\mathbf{E}[OPT(\omega)]$.

Pokażemy, że oczekiwany koszt połączeń wykupionych przez algorytm zachłanny wynosi $2\mathbf{E}[OPT(\omega)]$.

Koszt połączeń



EoR **dla znanego** k

Rozważmy teraz połączenia dodane przez algorytm zachłanny:

- dla i'tego wierzchołka v koszt połączenia jest ograniczony przez $d(v, D \cup \{r\})$, — odległość v do najbliższego wierzchołka z $D \cup \{r\}$.
- ponieważ każdy wierzchołek jest losowany z π, to z liniowości wartości oczekiwanej, oczekiwany koszt rozwiązania wynosi $k \times \mathbf{E}_{v \in \pi, D \in \pi^k}[d(v, D \cup \{r\})]$.
- Ten koszt nie wzrasta jeżeli weźmiemy mniejszy zbiór $D' \in \pi^{k-1}$ zawierający tylko $k - 1$ wierzchołków.

EoR **dla znanego** k

- Wyobraźmy sobie, że wybieramy D' oraz v poprzez wybranie k elementowego D'', a następnie randomly choosing one of them to be the vertex v;
- Ostatecznie wyrażenie $\mathbf{E}_{v\in\pi,D'\in\pi^{k-1}}[d(v,D'\cup\{r\})]$ jest ograniczone przez $\mathbf{E}_{D''\in\pi^k}[\frac{1}{k}\cdot MST(D''\cup\{r\})]$.
  - koszt każdego połączenia jest nie większy niż koszt płacony MST.
- Tak więc koszt połączeń wynosi $\mathbf{E}_{D\in\pi^k}[MST(D\cup\{r\})]$.
- MST jest 2-aproksymacyjne, a więc otrzymujemy 4-aproksymacyjny algorytm.

EoR **dla nieznanego** k

Teraz pokażemy jak pozbyć się założenia, że znamy długość sekwencji.

Zastosujemy metodę skalowania dla oczekiwanego kosztu rozwiązania.

W każdej „skali” będziemy budować rozwiązanie wstępne o koszcie dwa razy większym niż wcześniej.

Każda skala trwa aż otrzymamy tyle żądań ile jest w rozwiązaniu wstępnym.

EoR **dla nieznanego** k

Zdefiniujmy:

$$Z_\ell = \mathbf{E}_{\omega \in \pi^\ell}[\mathsf{OPT}(\omega)]$$

jako oczekiwany koszt optymalnego drzewa Steinera dla sekwencji żądań ω długości ℓ wylosowanej z π.

Skale t_i wybrane są jako najmniejsze takie, że $Z_{t_i} \geq 2^i$.

Jeżeli otrzymamy więcej żądań niż t_i to budujemy 2-przybliżone rozwiązanie wstępne na t_{i+1} losowo wygenerowanych wierzchołkach — $k \leq t_{i+1}$.

EoR dla nieznanego k

W każdej skali koszt rozwiązania wstępnego jest 4 razy większy niż koszt rozwiązania optymalnego, bo:

- oczekiwany koszt dla t_{i+1} wierzchołków jest co najwyżej dwa razy większy niż koszt na k wierzchołkach.

Koszt połączeń także możemy oszacować jako 4 razy większy niż koszt rozwiązania optymalnego.

EoR **dla nieznanego** k

Koszty dla skal tworzą szereg geometryczny.

Tak więc koszt wszystkich skal jest nie większy niż $8 + 8 = 16$ razy oczekiwany koszt rozwiązania dla k wierzchołków.

Twierdzenie 9 (EoR **dla znanego** k**)** *Istnieje wielomianowy algorytm dla stochastycznego problemu drzewa Steinera online taki, że* EoR $= O(1)$.

Zamiast używać dokładnych wartości Z_ℓ wystarczą nam aproksymacje.

Expectation of Ratios

Współczynnik aproksymacji, który jest często bardziej wymagający to *expectation of ratios* (EoR):

$$\mathsf{EoR}(\mathsf{ALG}) = \max_{\pi} \max_{k} \mathbf{E}_{\omega \in \pi^k} \left[\frac{\mathbf{E}_r[\mathcal{A}(\omega, r)]}{\mathsf{OPT}(\omega)} \right].$$

Założymy znowu, że długość sekwencji wejściowej jest z góry znana.

Expectation of Ratios

Będziemy wykonywać dwa algorytmy online równolegle.

Po realizacji żądania przez oba algorytmy, wybieramy ten który dotychczas zapłacił mniej i wykupujemy jego rozwiązanie.

W ten sposób nigdy nie zapłacimy więcej niż dwa razy koszt tańszego z tych dwóch algorytmów.

Expectation of Ratios

Równolegle z algorytmem zachłannym uruchamiamy następujący algorytm:

1. Wylosuj L różnych k elementowych zbiorów $D_1, \ldots, D_L$ z rozkładu π.
2. Dla każdego i, policz MST T_i dla zbioru $D_i \cup \{r\}$, ale nie wykupuj krawędzi.
3. Wybierz takie i^*, że koszt T_{i^*} jest najmniejszy, i wykup te krawędzie tzn. $T_M = T_{i^*}$.
4. Wierzchołki z sekwencji żądań podłączaj zachłannie do najbliższego wierzchołka.

Expectation of Ratios

Następujący lemat wynika natychmiast z symetrii problemu.

Lemat 10 *Z prawdopodobieństwem co najmniej* $1 - \frac{1}{L+1}$*, koszt najtańszego drzewa* T_{i^*} *wynosi co najwyżej* $4\mathsf{OPT}(R)$.

Rozważmy optymalne rozwiązanie T oraz cykl Eulera E skonstruowany z niego.

Podzielmy wierzchołki na E na sekwencje po $3\log Ln$ wierzchołków.

Expectation of Ratios

Dla każdego wierzchołka z prawdopodobieństwem $\frac{1}{n^3}$ z $3 \log nL$ najbliższych jest jeden z D_{i^*}.

Podłączenie pierwszego wierzchołka z sekwencji nie kosztuje więc więcej niż koszt sekwencji w E.

Następnie sekwencja łączona jest algorytmem zachłannym o koszcie $\log(3 \log nL)$.

Lemat 11 *Koszt zachłannych podłączeń żądań wynosi* $O(\mathsf{OPT}(R) \cdot \log\log(nL))$ *z prawdopodobieństwem co najmniej* $1 - \frac{1}{n^2}$.

Expectation of Ratios

Twierdzenie 12 *Dla $L = O(\log n)$, współczynnik* EoR *dla tego algorytmu wynosi $O(\log\log(n))$.*

- Jeżeli jeden z lematów zawiedzie: co dieje się z prawdopodobieństwem $\frac{1}{L+1} + \frac{L}{n^2} \leq \frac{2}{\log n}$. W takim wypadku płacimy koszt zachłanny $O(\log n)$, co daje stały wkład do EoR.
- Jeżeli obydwa lematy są spełnione to płacimy $O(\mathsf{OPT}(R) \cdot \log\log n)$.

Podsumowanie

- W przypadku stochastycznym możemy pokonać klasyczne ograniczenia na współczynnik kompetytywny $\Omega(\log n)$ dla drzewa Steinera.
- Techniki te mogą być zastosowane także w przypadku problemów addytywnych.

Pozostały jednak pewne problemy otwarte:

- przypadek nieznanego k dla miary EoR,
- Jakie innych problemy możemy rozwiązać w takim modelu?