Metody Probabilistyczne i Statystyka

ZADANIE DOMOWE 4

Termin wysyłania (MS Teams): **31 stycznia 2023 godz. 23:59**

Zadanie 1. [5 pkt.] (*Graph exploration / Cover time of a random walk*)

Wprowadzenie. Niech $G = (V, E)$ będzie nieskierowanym grafem, gdzie $V = \{1, \ldots, n\}$ jest zbiorem n wierzchołków, a $E \subseteq \{\{u, v\} \colon u, v \in V\}$ jest zbiorem m krawędzi. Zakładamy, że G jest grafem prostym (bez krawędzi wielokrotnych i pętli, czyli krawędzi prowadzących od pewnego wierzchołka do siebie samego) oraz spójnym, tj. istnieje nieskierowana ścieżka między dwoma dowolnymi wierzchołkami. (mówiąc nieformalnie, dla dowolnych $u, v \in V$ można „przejść po krawędziach" od u do v).

Załóżmy, że graf G opisuje pewną sieć, której węzły reprezentowane są przez wierzchołki, a krawędzie odpowiadają istniejącym połączeniom między węzłami. Początkowo w pewnym węźle $X_0 \in V$ znajduje się jeden agent, który może przemieszczać się pomiędzy incydentnymi wierzchołkami grafu sieci. W jednym kroku (rundzie) agent może przejść przez dowolnie wybraną krawędź od wierzchołka, w którym aktualnie się znajduje, do jednego z sąsiednich wierzchołków. Zakładamy, że początkowo agent nie zna topologii sieci, a jego celem jest eksploracja grafu sieci, tj. odwiedzenie wszystkich węzłów w możliwiej jak najkrótszym czasie (czas rozumiemy tu jako liczbę rund) i przy użyciu jak najmniejszej pamięci (w literaturze rozważa się różne ograniczenia na zasoby, którymi dysponuje taki agent – między innymi rozważane są scenariusze, w których agent nie może zapisywać żadnych informacji w trakcie eksploracji).

Chociaż samo sformułowanie problemu jest dosyć abstrakcyjne[1], motywowane jest ono wieloma praktycznymi zastosowaniami, obejmującymi zarówno systemy złożone z mobilnych urządzeń, które mogą przemieszczać się pomiędzy kolejnymi lokalizacjami, jak i sieci logiczne, w których agenci reprezentują uruchamiane w węzłach procesy przetwarzające gromadzone dane (przykładem może być tutaj oprogramowanie monitorujące aktywność poszczególnych serwerów oraz ruch sieciowy w ich obrębie celem wykrycia nieautoryzowanych prób dostępu i podjęcia ewentualnych akcji mających im zapobiec).

Błądzenie losowe. Jedną z najprostszych, elementarnych strategii eksploracji nieznanego grafu sieci przez agenta jest błądzenie losowe (spacer losowy, ang. *random walk*). Błądzenie losowe na grafie $G = (V, E)$ jest łańcuchem Markowa, w którym przestrzenią stanów jest zbiór wierzchołków V. W podstawowym wariancie – tzw. proste błądzenie losowe (ang. *simple random walk*) – w każdej kolejnej rundzie $t \geqslant 1$ agent wykonujący spacer losowy, znajdujący się w wierzchołku $X_{t-1} = v$ o stopniu $d(v)$, losuje niezależnie z jednakowym prawdopodobieństwem równym $\frac{1}{d(v)}$ jedną z $d(v)$ krawędzi incydentnych (połączonych) z v, którą następnie przechodzi do kolejnego wierzchołka X_t. Jeśli wylosowana krawędź to $\{v, u\}$, to w kolejnym kroku agent znajduje się w wierzchołku u (tj. $X_t = u$), gdzie powtarza całą procedurę. Nieco zmodyfikowana wersja, nazywana leniwym błądzeniem losowym (ang. *lazy random walk*), zakłada możliwość pozostania agenta w wierzchołku, w którym aktualnie przebywa. Mianowicie, w każdej rundzie $t \geqslant 1$ agent znajdujący się w pewnym wierzchołku $v \in V$ z ustalonym prawdopodobieństwem $p > 0$ zostaje w v, a z prawdopodobieństwem $1 - p$ wykonuje jeden

[1] A także mocno uproszczone na potrzeby zadania.

krok prostego błądzenia losowego (stała p może być wybrana dowolnie; w literaturze często przyjmuje się dla uproszczenia $p = \frac{1}{2}$).

Chociaż leniwy spacer losowy na pierwszy rzut oka może wydawać się „mniej interesujący", ponieważ jest wolniejszy od błądzenia prostego, w przeciwieństwie do tego drugiego jest on zawsze procesem nieokresowym (ang. *aperiodic*). Rozważmy siec opisaną przez pewien spójny graf dwudzielny, tj. taki graf $G = (V, E)$, w którym zbiór wierzchołków V można rozbić na dwa niepuste i rozłączne podzbiory $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, takie, że żadne dwa wierzchołki z V_1 (odpowiednio, V_2) nie są ze sobą połączone krawędzią (innymi słowy, możemy pokolorować wierzchołki grafu dwoma kolorami – np. czarnym i białym – w taki sposób, że żadne dwa wierzchołki tego samego koloru nie są połączone). W szczególności każde drzewo jest grafem dwudzielnym. Zauważmy, że jeśli proces prostego błądzenia losowego na takim grafie sieci startuje w wierzchołku białym (odpowiednio, czarnym), to w rundach parzystych zawsze będzie znajdował się w jednym z wierzchołków białych (czarnych), a w nieparzystych – w czarnych (białych). Dla leniwego spaceru losowego ten „problem" nie występuje. W rezultacie leniwe spacery losowe na nieskierowanych grafach spójnych są przykładami ergodycznych łańcuchów Markowa (ang. *ergodic Markov chains*). Jedną z konsekwencji tego jest następujący fakt: bez względu na to, z którego wierzchołka wystartował leniwy spacer losowy, po wykonaniu odpowiednio dużej liczby kroków t (w ogólności zależnej od topologii i rozmiaru grafu sieci), prawdopodobieństwo tego, że po t krokach proces znajdzie się w danym wierzchołku v, „jest bliskie" $\frac{d(v)}{2m}$ (formalnie, prawdopodobieństwo tego, że $X_t = v$, dąży do $\frac{d(v)}{2m}$ wraz z $t \to \infty$, bez względu na to, z którego wierzchołka proces wystartował), tj. jest proporcjonalne do stopnia v (zauważmy, że $\sum_{v \in V} d(v) = 2m$).

Czas pokrycia dla błądzenia losowego. Celem tego zadania jest eksperymentalne zbadanie czasu pokrycia (ang. *cover time*) dla procesu błądzenia losowego dla wybranych rodzin grafów i wierzchołków startowych X_0. W kontekście problemu eksploracji grafu sieci jest to pierwszy moment czasu t, w którym agent wykonujący błądzenie losowe co najmniej raz odwiedził każdy węzeł sieci. Formalnie, niech $(X_t)_{t \in \mathbb{N}}$ będzie procesem błądzenia losowego na grafie $G = (V, E)$, który startuje w wierzchołku $X_0 = v_0$. Wówczas czasem pokrycia nazywamy zmienną losową τ_{v_0} zdefiniowaną jako

$$\tau_{v_0} = \min\{t \in \mathbb{N} \colon (\forall\, v \in V)(\exists\, t' \leqslant t)(X_{t'} = v)\}\,.$$

Eksperymenty. Zaimplementuj proces błądzenia losowego na nieskierowanych grafach prostych. Do reprezentacji grafu możesz użyć np. list sąsiedztwa, macierzy sąsiedztwa, lub skorzystać z dowolnej biblioteki do obsługi grafów. Następnie przeprowadź eksperymentalne badanie średniego czasu pokrycia dla procesu błądzenia losowego dla następujących grafów sieci i wierzchołków startowych (w każdym przypadku $V_n = \{1, \ldots, n\}$):

(a) **klika** (graf pełny, ang. *clique*) $K_n = (V_n, E_n)$, gdzie $E_n = \{\{u, v\} \colon u, v \in V, u \neq v\}$; proces startuje w dowolnym ustalonym wierzchołku,

(b) **ścieżka** (ang. *path graph*) $P_n = (V_n, E_n)$, gdzie $E_n = \{\{v, v+1\} \colon v \in \{1, \ldots, n-1\}\}$; proces startuje „na środku ścieżki", tj. $X_0 = \lfloor n/2 \rfloor$,

(c) **ścieżka** P_n; proces startuje „na końcu ścieżki", tj. $X_0 = 1$ (lub $X_0 = n$),

(d) **zupełne drzewo binarne** (ang. *complete binary tree*) T_n – drzewo binarne, w którym wszystkie poziomy (być może za wyjątkiem ostatniego) są wypełnione, a ostatni poziom wypełniony jest „od lewej" (przyjmijmy, że wierzchołki numerujemy kolejnymi liczbami naturalnymi od korzenia w dół drzewa, a na danym poziomie od lewej do prawej); proces startuje w korzeniu drzewa,

(e) **lizak** (ang. *lollipop graph*) L_n – graf składający się z kliki $K_{\lfloor 2n/3 \rfloor}$ rozmiaru $\left\lfloor \frac{2n}{3} \right\rfloor$ oraz „doczepionej" do jednego z jej wierzchołków ścieżki $P_{n-\lfloor 2n/3 \rfloor}$ rozmiaru $n - \left\lfloor \frac{2n}{3} \right\rfloor$ (por. np. Example 6.1 z rozdziału 6.3 w [MR95]); proces startuje w dowolnym ustalonym wierzchołku kliki $K_{\lfloor 2n/3 \rfloor}$ (wybierz wierzchołek niepołączony ze ścieżką).

Dla każdego $n \in \{100, 150, \ldots, 2000\}$ wykonaj po k niezależnych powtórzeń eksperymentu polegającego na symulacji procesu błądzenia losowego i wyznaczenia czasu pokrycia dla każdego z wyżej opisanych przypadków (dobierz odpowiednią wartość k). Zadbaj o to, aby generator liczb pseudolosowych użyty w symulacjach był „dobry" (tj. miał dobre własności statystyczne).

Po zakończeniu symulacji, korzystając z zebranych danych, dla każdego z badanych przypadków przedstaw na wykresach wyniki poszczególnych powtórzeń (k punktów danych dla każdego n) oraz średnią wartość czasu pokrycia jako funkcję n. Wartość średnią oraz wszystkie wyniki poszczególnych prób nanieś na wspólny wykres tak, aby można było łatwo określić ich koncentrację wokół wartości średniej.

Zaprezentuj wykresy, zwięźle omów uzyskane wyniki (w tym porównaj rezultaty otrzymane dla poszczególnych rodzin grafów i podanych wierzchołków startowych) i przedstaw płynące z nich wnioski. Spróbuj wyznaczyć eksperymentalnie możliwie ścisłe ograniczenia asymptotyczne na średni czas pokrycia w badanych przypadkach (np. czy jest rzędu $\mathrm{O}(n)$, $\mathrm{O}(n \ln n)$, $\mathrm{O}(n^2)$, $\mathrm{O}(n^3)$ itp.). W tym celu możesz poszukać informacji w dostępnej literaturze.

Uwagi dodatkowe i literatura (dla zainteresowanych). Błądzenie losowe na grafach jest jednym z fundamentalnych przykładów łańcuchów Markowa. Elementy teorii łańcuchów Markowa oraz ich zastosowanie w informatyce do projektowania oraz badania algorytmów dla szeregu problemów obliczeniowych przedstawione są m.in. w [Häg02, LPW09, MU17, MR95] (szczegółowo omówione są tam m.in. procesy błądzenia losowego i ich wybrane własności). W [Fei95a, Fei95b] pokazane zostały dolne i górne ograniczenia na czas pokrycia dla procesu błądzenia losowego na grafach.

Eksploracja grafu sieci przez mobilnych agentów (zarówno przez pojedynczego agenta, jak i eksploracja kolaboratywna wykonywana przez wielu agentów) jest jednym z intensywnie badanych tematów w dziedzinie algorytmów rozproszonych. W literaturze rozważany jest szereg modeli teoretycznych dla tego problemu oraz prezentowane są efektywne algorytmy dla różnych scenariuszy. Omówienie klasycznych oraz aktualnych wyników badań w tym zakresie można znaleźć m.in. w artykułach przeglądowych [Das19, Ilc19].

Rozwiązanie zadania obejmujące

- implementacje symulacji (kod źródłowy w wybranym języku programowania) oraz
- pdf z wykresami, zwięzłym opisem wyników i wnioskami

należy przesłać na platformę MS Teams. Nie należy dołączać żadnych zbędnych plików.

Literatura

[Das19] Shantanu Das. Graph Explorations with Mobile Agents. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 403–422. Springer International Publishing, Cham, 2019.

[Fei95a] Uriel Feige. A Tight Lower Bound on the Cover Time for Random Walks on Graphs. *Random Structures & Algorithms*, 6(4):433–438, July 1995.

[Fei95b] Uriel Feige. A Tight Upper Bound on the Cover Time for Random Walks on Graphs. *Random Structures & Algorithms*, 6(1):51–54, January 1995.

[Häg02] Olle Häggström. *Finite Markov Chains and Algorithmic Applications*. Cambridge University Press, 3rd edition, 2002.

[Ilc19] David Ilcinkas. Oblivious Robots on Graphs: Exploration. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, pages 218–233. Springer International Publishing, Cham, 2019.

[LPW09] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 1st edition, 2009.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1st edition, 1995.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017.