

Algorytmy On-Line

Lista 2

Adrian Herda

2025-04-07

1. Treść zadania

Dla problemu stronicowania rozważamy cache o pojemności k dla zbioru n stron żądanych zgodnie z podanym rozkładem. Zbadaj średni koszt żądania strony dla podanych rozkładów, algorytmów, k i n .

Rozważ następujące rozkłady zmiennej losowej X dla n elementów (ze zbioru $\{1, \dots, n\}$):

- jednostajny $Pr[X = i] = \frac{1}{n}$
- harmoniczny $Pr[X = i] = \frac{1}{i \cdot H_n}$, gdzie H_n jest liczbą harmoniczną,
- dwuharmoniczny $Pr[X = i] = \frac{1}{i^2 \cdot \hat{H}_n}$, gdzie $\hat{H}_n = \sum_{i=1}^n \frac{1}{i^2}$ jest n -tą liczbą dwuharmoniczną,
- geometryczny $Pr[X = i] = \frac{1}{2^i}$, dla $i < n$, i $Pr[X = n] = \frac{1}{2^{n-1}}$.

Zastosuj następujące metody obsługi cache'a:

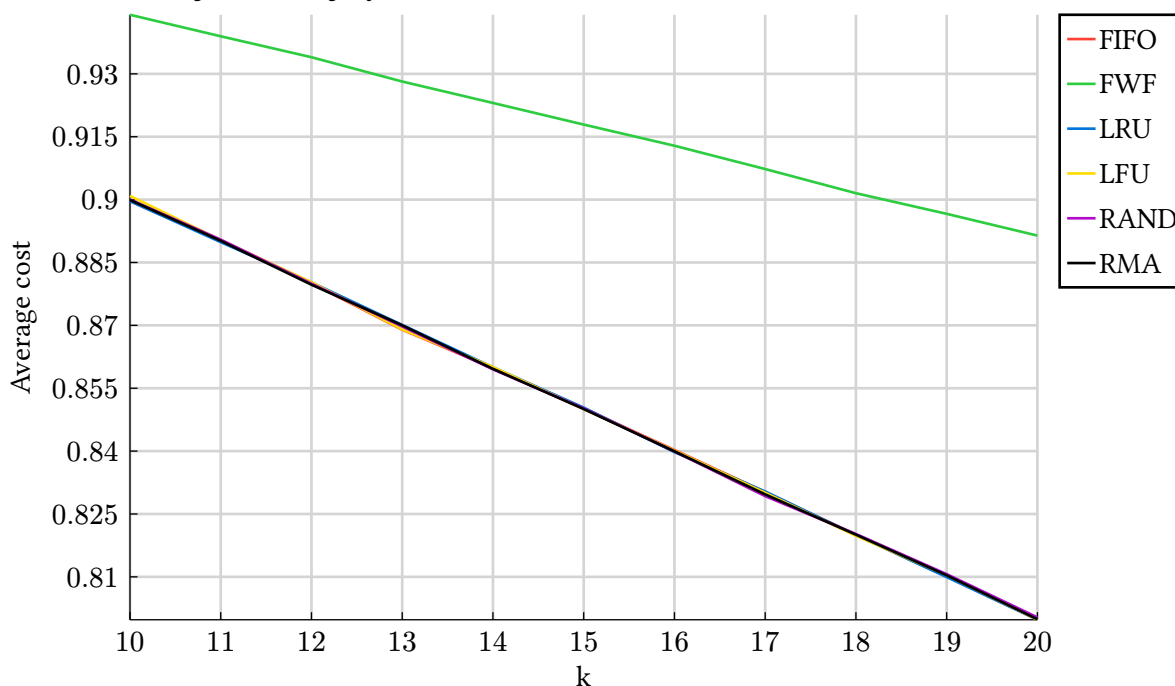
- FIRST IN FIRST OUT (FIFO),
- FLUSH WHEN FULL (FWF),
- LEAST RECENTLY USED (LRU),
- LEAST FREQUENTLY USED (LFU) – licznik użycia strony przechowujemy nawet jeśli strony nie ma w cache'u,
- RANDOM (RAND) – losujemy stronę do wyrzucenia z rozkładem jednostajnym w całym cache'u,
- RANDOMIZED MARKUP ALGORITHM (RMA) – stosujemy algorytm oznaczający i wyrzucamy stronę nieoznaczoną, losowaną z rozkładem jednostajnym.

Przeprowadź eksperymenty dla n ze zbioru $\{20, 30, 40, 50, 60, 70, 80, 90, 100\}$ i k ze zbioru $\{\frac{n}{10}, \dots, \frac{n}{5}\}$ (np. dla $n = 40$ mamy $k \in \{4, 5, 6, 7, 8\}$). Przygotuj krótkie sprawozdanie ilustrujące uzyskane wyniki.

2. Wyniki

Eksperyment polegał na losowaniu 1000000 stron a następnie wyciągania ich z cache'a który był obsługiwany przez różne algorytmy opisywane w rozdziale 1. Testy były wykonywane na różnego rodzaju rozkładach prawdopodobieństwa stosowanych do losowania wyciąganych stron, na różnych wielkościach cache'a oraz na różnych zbiorach stron.

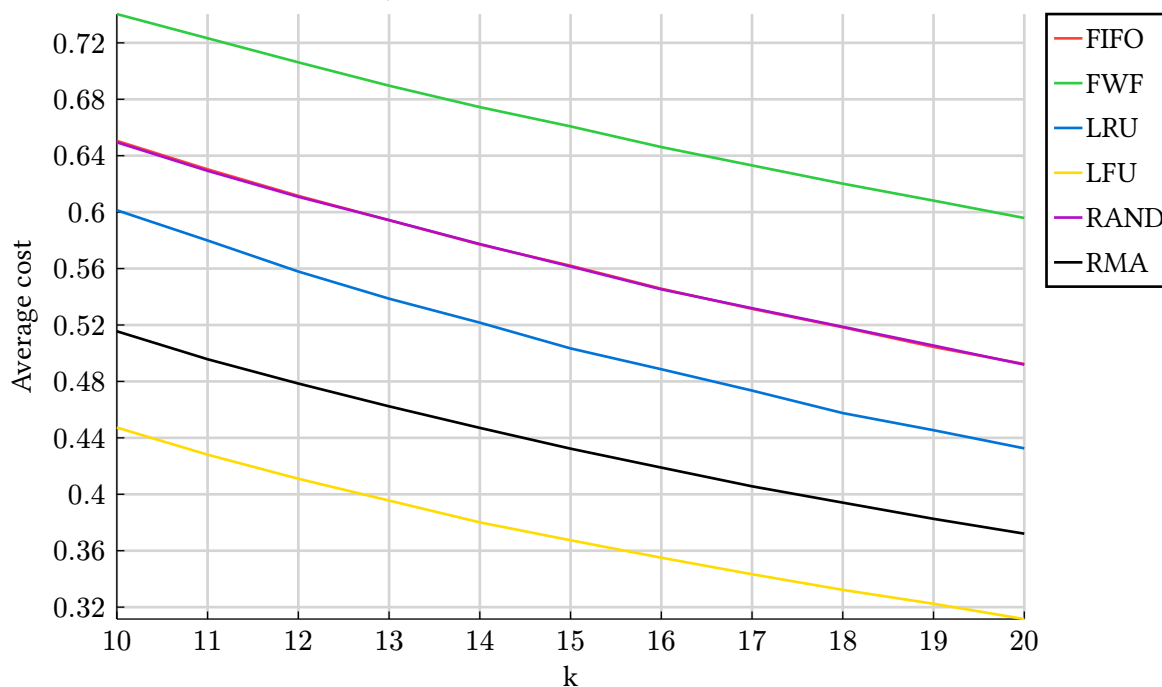
2.1. Rozkład jednostajny



Rysunek 1: Average cost of getting page with:
n = 100 and Uniform distribution

Rysunek 1 przedstawia wykres średnich kosztów dla $n = 100$ oraz $k \in \{10, \dots, 20\}$ dla różnych metod obsługi cache'a. Algorytm FWF (ang. Flush When Full) widocznie odstaje od innych algorytmów, które wydają się być dokładnie tak samo wydajne w stronach losowanych w rozkładzie jednostajnym.

2.2. Rozkład harmoniczny

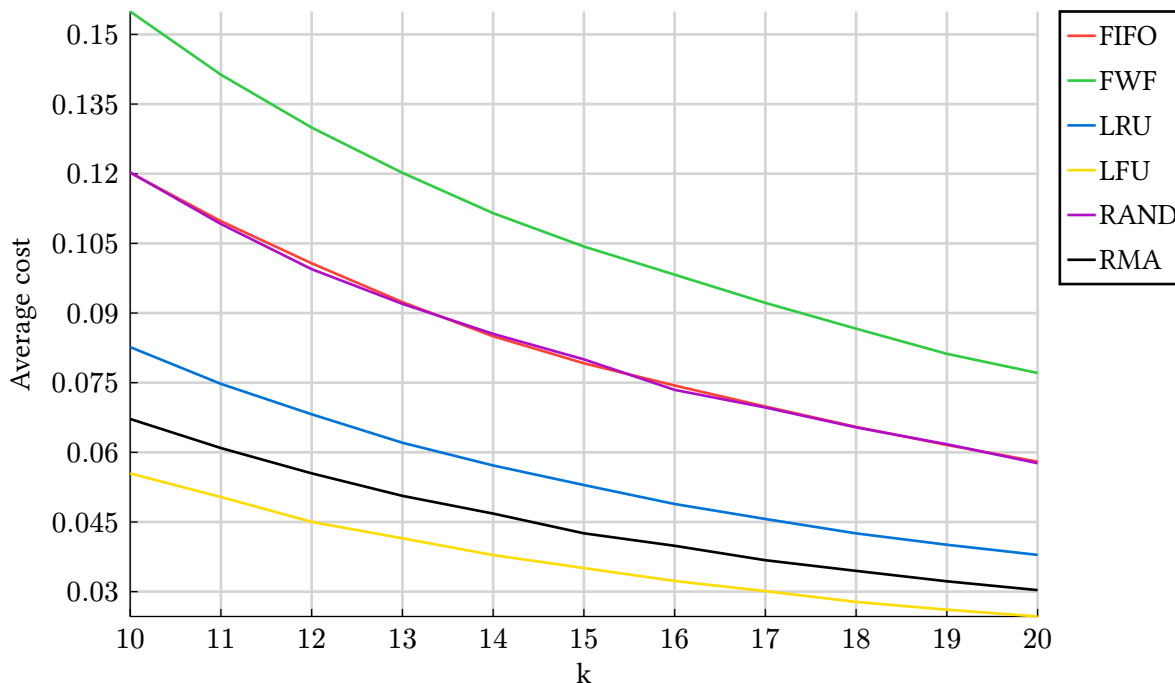


Rysunek 2: Average cost of getting page with:
n = 100 and Harmonic distribution

Na Rysunek 2 widać porównanie opisywanych metod obsługi chache'a względem ich średniego kosztu przy rozkładzie harmonicznym. Algorytm FWF podobnie jak w poprzednim rozkładzie wypada

najgorzej. Algorytmy typu FIFO (ang. First In, First Out) oraz RAND (ang. Randomized) mają bardzo zbliżony średni koszt. Najlepsze okazują się algorytmy – w kolejności – LRU (ang. Least Recently Used), RMA (ang. Randomized Markup Algorithm) oraz LFU (ang. Least Frequently Used), gdzie ten ostatni ma średni koszt prawie dwa razy mniejszy od najgorszego.

2.3. Rozkład dwuharmoniczny



Rysunek 3: Average cost of getting page with:
n = 100 and Double Harmonic distribution

Rysunek 3 ma dokładnie tę samą kolejność algorytmów co Rysunek 2. Jedyna różnica pomiędzy tymi wykresami jest taka że wszystkie algorytmy mają dużo mniejszy średni koszt i maleje on w sposób bardziej geometryczny niż wykres z rozkładem harmonicznym.

2.4. Rozkład geometryczny