**dr hab. Bartłomiej Bosek, prof. UJ**
Wydział Matematyki i Informatyki
Uniwersytet Jagielloński, 31-007 Kraków
E-mail: bosek@tcs.uj.edu.pl

Kraków, 20.03.2022

# Recenzja rozprawy doktorskiej Artura Kraski

# pt. *„Cena opóźnień w algorytmach online”*

Rozprawa doktorska Artura Kraski dotyczy algorytmów online z przyzwoleniem na opóźnienia. Klasyczne algorytmy online otrzymują na wejściu w następujących po sobie turach kolejne fragmenty jakieś struktury (np. wierzchołki grafu). Po otrzymaniu kolejnej porcji informacji na wejściu algorytm online musi natychmiast zaprezentować fragment wyniku, który zazwyczaj odnosi się do tego, co się właśnie pojawiło na wejściu (np. kolor nowo-zaprezentowanego wierzchołka). Celem algorytmu online jest zazwyczaj minimalizowanie „kosztu” związanego z przetwarzaniem sekwencyjnie otrzymywanej struktury (np. liczby użytych kolorów). Autor rozprawy rozważa pewną wariację powyższego problemu, polegająca na tym, że algorytm z daną odpowiedzią może zwlekać ileś tur. Z jednej strony to pozwala algorytmowi na podejmowanie bardziej trafnych decyzji. Z drugiej zaś strony, za coraz większą zwłokę algorytm musi zapłacić coraz większą karę, która jest dodatkową wartością dodawaną do sumarycznego kosztu algorytmu. W tak rozszerzonym problemie pytanie, które jest ciekawe rozważania, to czy pozwolenie na opóźnienia w podejmowaniu decyzji zmniejszy sumaryczny koszt, pomimo płacenia dodatkowo naliczonych kar.

Autor w swojej rozprawie doktorskiej koncentruje się na problemach związanych z pojawianiem się obiektów w pewnej skończonej przestrzeni metrycznej. W takim przypadku problemy online z opóźnieniami można generalnie podzielić na dwa rodzaje:

(1) W pierwszym przypadku wykonywane zadania zajmują określoną ilość czasu, która zazwyczaj wynika bezpośrednio z decyzji algorytmu oraz pośrednio z umiejscowienia obiektów w przestrzeni. W tym przypadku dodatkowe opóźnienia po prostu wliczają się naturalnie w sumaryczny czas wykonania zadań. Celem jest takie skonstruowanie algorytmu online, aby omawiany sumaryczny czas jak najmniej odbiegał od przypadku, gdy całe wejście jest znane od razu (przypadku offline). Autor w tym wariancie rozważa *problem podróżującego mechanika* (traveling repair person problem) oraz *problem wyboru przejazdów* (dial-a-ride problem).
(2) W drugim przypadku wykonanie zadania nie zajmuje żadnego czasu. Zaś funkcja kosztu zależy od decyzji algorytmu oraz od odległości pomiędzy obiektami umiejscowionymi w rozważanej przestrzeni. Zwłokę z podjęciem decyzji dodaje się jako dodatkową wartość do funkcji kosztu. W tym wariancie rozważany jest *problem usług online z opóźnieniami*

(online service with delay) oraz *problem skojarzeń z opóźnieniami* (online matching with delays).

Na początku podjęte są trzy następujące problemy, należące do klasy problemów, w których zadania zajmują określoną ilość czasu. Pierwszym omawianym problemem jest *problem podróżującego mechanika*. W tym problemie zadana jest przestrzeń metryczna z wyróżnionym punktem zwanym punktem startowym. W momencie początkowym działania algorytmu w punkcie startowym jest umieszczony serwer, który następnie może przemieszczać się po zadanej przestrzeni ze stałą prędkością. Z biegiem czasu pojawiają się żądania w pewnych punktach przestrzeni metrycznej. Ponadto każde żądanie ma przypisaną pewną nieujemną wagę. Dane żądanie jest traktowane jako obsłużone, jeśli poruszający się serwer pojawi się na pozycji tego żądania. Koszt związany z danym żądaniem to czas, jaki upłynął od momentu początkowego działania algorytmu pomnożony przez wagę obsłużonego żądania. Celem algorytmu jest takie przesuwanie serwerem po zadanej przestrzeni metrycznej, aby zminimalizować sumaryczny koszt obsługi wszystkich żądań.

Autor rozważa zarazem *problem wyboru przejazdów* (dial-a-ride problem), który jest pewnym uogólnieniem powyższego problemu. W tym przypadku, każde żądanie jest zdefiniowane poprzez dwa punkty w przestrzeni metrycznej, które można nazwać odpowiednio źródłem i celem. Aby obsłużyć żądanie, serwer musi najpierw przenieść się do źródła, „odebrać" żądanie, a następnie przenieść się do celu. W momencie pojawienia się w miejscu oznaczonym jako cel żądanie traktowane jest jako zrealizowane. Podobnie jak w przypadku problemu podróżującego mechanika, algorytm ma tak kierować serwerem, aby zminimalizować całkowity ważony koszt usługi.

Trzecim pokrewnym do dwóch pierwszych jest problem szeregowania zadań na niepowiązanych maszynach, gdzie celem jest minimalizacja średniego czasu zakończenia zadania. W tym przypadku zadania pojawiają się w określonych momentach czasowych i mają przypisane nieujemne wagi. Pojedyncza maszyna może jednocześnie wykonywać co najwyżej jedno zadanie. Celem jest przypisanie każdego zadania (w momencie lub po jego przybyciu) do jednej z maszyn, aby zminimalizować ważoną sumę czasów zakończeń.

Na wstępie trzy powyższe problemy zostały uogólnione do pojęcia $\gamma$-resetowalnych problemów przydzielania zadań. Po pierwsze, wszystkie one posiadają tzw. *stan początkowy*. Po drugie, w dużym uproszczeniu pojęcie to oznacza, między innymi, że w problemie 1-resetowalnym ilość czasu potrzebna do powrotu do stanu początkowego jest nie większa niż czas, który minął od ostatniego momentu, gdy algorytm był w stanie początkowym, do danego momentu. Do tych problemów zaliczany jest problem podróżującego mechanika jak i problem wyboru przejazdów. Z kolei w problemach 0-resetowalnych każdy algorytm zawsze jest w stanie początkowym. Do tej klasy należy problem szeregowania zadań na niepowiązanych maszynach.

Autor zaczyna od zaprezentowania algorytmu BASIC, który rozwiązuje 1-resetowalne problemy kolejkowania zadań online z współczynnikiem konkurencyjności 6 — Podrozdział 2.4. Następnie został zaprezentowany dużo bardziej złożony algorytm MIMIC, który rozwiązuje $\gamma$-resetowalne problemy kolejkowania zadań online z współczynnikiem konkurencyjności $3+\gamma$. Dodatkowa randomizacja[1] algorytmu MIMIC daje współczynnik konkurencyjności równy $1+(1+\gamma)/\ln(2+\gamma)$ — Wniosek 2.1. W konsekwencji dla problemu podróżującego mechanika

[1] Randomizacja algorytmu MIMIC polega na wylosowaniu dodatkowego parametru $\omega \in (-1,0]$ od którego ten algorytm zależy. W przypadku deterministycznym $\omega = 0$.

jak i problemu wyboru przejazdów autor uzyskuje algorytm 4-konkurencyjny w przypadku deterministycznym oraz 2.821-konkurencyjny w przypadku randomizowanym. Poprawia to wyniki Hwanga i Jailleta[2] pokazujące stałe konkurencyjności równe odpowiednio 5.14 oraz 3.641. Ponadto dla problemu szeregowania zadań na niepowiązanych maszynach autor uzyskuje algorytm 3-konkurencyjny w przypadku deterministycznym oraz 2.443-konkurencyjny w przypadku randomizowanym. To z kolei poprawia wynik Halla i innych.[3] równy 4 w przypadku deterministycznym oraz wynik Chakrabarti'ego i innych.[4] równy 2.886 w przypadku randomizowanym.

Zaprezentowany wynik jest dalece nietrywialny zaś zastosowane techniki są pomysłowe. Rezultaty dotyczące problemu podróżującego mechanika jak i problemu wyboru przejazdów odnoszą się także w przypadku dowolnej ustalonej liczby serwerów, dowolnych pojemności serwerów, zarówno w przypadku z wywłaszczaniem, jak i bez wywłaszczania. Ponadto warto tu szczególnie zwrócić uwagę na to, że opracowane techniki pozwoliły na poprawienie wyników dla problemu szeregowania zadań na niepowiązanych maszynach, co było poprawą wyniku po 25 latach. Wyniki te zostały zaprezentowane na *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*.[5]

W dalszej części rozprawy doktorskiej autor przechodzi do problemów, w których wykonanie zadania nie zajmuje czasu. Pierwszym z tej grupy problemów to tzw. *problem obsługi zadań z opóźnieniami* (online service with delays). Podobnie jak w problemie podróżującego mechanika zadana jest przestrzeń metryczna z wyróżnionym punktem zwanym punktem startowym. W momencie początkowym działania algorytmu w punkcie startowym jest umieszczony serwer, który następnie może przemieszczać się bo zadanej przestrzeni. W odróżnieniu jednak od problemu podróżującego mechanika serwer zmienia swoje położenie natychmiast płacąc jednak *koszt związany z przemieszczaniem się*, który równy jest długości przemierzonej trasy. Z biegiem czasu pojawiają się żądania w pewnych punktach przestrzeni metrycznej. Dane żądanie jest traktowane jako obsłużone, jeśli poruszający się serwer pojawi się na pozycji tego żądania. Serwer nie ma obowiązku obsługiwania zadań w momencie ich pojawienia się. Może on poczekać, aby zebrało się więcej żądań, w celu podjęcia bardziej optymalnej decyzji. Nie mniej jednak, dla każdego żądania jest naliczany dodatkowy *koszt związany ze zwłoką*, który jest ciągłą niemalejącą funkcją zależną od tego ile czasu upłynęło od momentu pojawienia się żądania, aż do jego obsłużenia. Dodatkowo zakłada się, że jeżeli żądanie zostało obsłużone w momencie pojawienia się, to jego koszt związany ze zwłoką równy jest 0. Celem algorytmu jest takie przesuwanie serwerem po zadanej przestrzeni metrycznej, aby zminimalizować sumę wszystkich kosztów związanych z przemieszczaniem się oraz wszystkich kosztów związanych ze zwłoką.

---

[2] D. Hwang, P. Jaillet. Online scheduling with multi-state machines. *Networks*, 71(3), 209–251, 2018.

[3] L.A. Hall, A.S. Schulz, D.B. Shmoys, J. Wein. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Mathematics of Operations Research*, 22(3), 513-544, 1997.

[4] S. Chakrabarti, C.A. Phillips, A.S. Schulz, D.B. Shmoys, Clifford Stein, and Joel Wein. Improved scheduling algorithms for minsum criteria. In *Proc. 23rd Int. Colloq. on Automata, Languages and Programming (ICALP)*, 646-657, 1996.

[5] M. Bienkowski, A. Kraska, and H. Liu. Traveling repairperson, unrelated machines, and other stories about average completion times. *In Proc. 48th Int. Colloq. on Automata, Languages and Programming (ICALP)*, 28:1-28:20, 2021.

Najlepszy znany rezultat dla powyżej zdefiniowanego problemu należy do Azara i Touitou'a,[6] którzy zaprezentowali randomizowany $O(\log^2 n)$-konkurencyjny algorytm, gdzie $n$ to liczba wszystkich potencjalnych pozycji żądań.[7] Autor rozprawy ogranicza się do problemu, gdy zbiór potencjalnych pozycji pojawienia się żądań tworzy $n$ równo rozmieszczonych punktów na osi liczb rzeczywistych. Bez straty ogólności można założyć, że jest to $n$ kolejnych liczb całkowitych. Dla tak ograniczonego problemu autor prezentuje deterministyczny $O(\log n)$-konkurencyjny algorytm BCKT — Twierdzenie 3.1. Z jednej strony struktura przestrzeni metrycznej rozważana w omawianej pracy doktorskiej jest sporym zawężeniem — nietrywialnym, ale jednak najdalej idącym, jaką w sumie można sobie wymyśleć. Z drugiej strony zaprezentowany algorytm w stosunku do wyniku Azara i Touitou'a jest deterministyczny i ma lepszą stałą konkurencyjności. Omawiany rezultat został zaprezentowany na *25th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2018)*.[8]

Ostatnim problemem podjętym w omawianej rozprawie doktorskiej jest *problem skojarzeń z opóźnieniami* (online matching with delays). Podobnie jak w powyższych problemach jest zadana przestrzeń metryczna rozmiaru $n$. Z biegiem czasu pojawiają się żądania w pewnych punktach tej przestrzeni. Przy czym wszystkich żądań, które się pojawią, jest $2m$. W dowolnym momencie algorytm jest w stanie połączyć dowolną parę żądań, które już się pojawiły, ale jeszcze nie zostały skojarzone. Tak jak i w poprzednim problemie, tak i w tym, obsługa żądań (w tym przypadku ich kojarzenie) nie zajmuje czasu. Zamiast tego algorytm musi zapłacić koszt równy odległości kojarzonych żądań. Ponadto algorytm ma prawo dowolnie długo zwlekać z podjęciem decyzji — płaci za to jednak dodatkowy koszt równy sumie czasów od pojawienia się żądań, aż do ich skojarzenia. Celem algorytmu jest skojarzyć wszystkie żądania minimalizując przy tym sumaryczny koszt wynikający z odległości kojarzonych żądań jak i wynikający z czasów zwlekania z podejmowaniem danych decyzji. Azar i inni.[9] zaprezentowali randomizowany $O(\log n)$-konkurencyjny algorytm rozwiązujący powyższy problem. Z drugiej zaś strony Ashlagi i inni.[10] pokazali dolne oszacowanie na stałą konkurencyjności równe $\Omega(\log n / \log\log n)$ także dla algorytmów randomizowanych. Rozważana jest także wersja dwudzielna problemu skojarzeń z opóźnieniami. Różni się ona jedynie tym, że nadchodzące żądania mają przypisane jeden z dwóch kolorów, zaś algorytm może kojarzyć jedynie żądania o różnych kolorach. W przypadku dwudzielnym stała konkurencyjności dla algorytmów randomizowanych mieści się pomiędzy $\Omega(\sqrt{\log n / \log\log n})$ (Ashlagi i inni.[10]), a $O(\log n)$ (Azar i inni.[9]). Autor omawianego doktoratu prezentuje deterministyczny $O(m)$-konkurencyjny algorytm GREEDY DUAL

[6] Y. Azar and N. Touitou. General framework for metric optimization problems with delay or with deadlines. *In Proc. 60th IEEE Symp. on Foundations of Computer Science (FOCS)*, 60-71, 2019.

[7] Innymi słowy $n$ to rozmiar rozważanej przestrzeni metrycznej.

[8] M. Bienkowski, A. Kraska, P. Schmidt. Online service with delay on a line. *In Proc. 25th Int. Colloq. on Structural Information and Communication Complexity (SIROCCO)*, 237-248, 2018.

[9] Y. Azar, A. Chiplunkar, H. Kaplan. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. *In Proc. 28th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 1051-1061, 2017.

[10] I. Ashlagi, Y. Azar, M. Charikar, A. Chiplunkar, O. Geri, H. Kaplan, R.M. Makhijani, Y. Wang, R. Wattenhofer. Min-cost bipartite perfect matching with delays. *In Proc. 20th Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, 1:1–1:20, 2017.

rozwiązujący omawiany problem w wersji ogólnej jak i w wersji dwudzielnej — Twierdzenie 4.1. Równolegle Azar i Jacob-Fanani[11] pokazali deterministyczny $O(m^{0.59})$-konkurencyjny dla powyższego problemu w obu wersjach — ogólnym i dwudzielnym.

Wynik zaprezentowany w omawianym doktoracie jest eleganckim zastosowaniem schematu prymalno-dualnego w konstruowaniu algorytmu, a nie tylko w jego analizie zachowania. Autor niewątpliwie musiał być dobrze zaznajomiony z użytą techniką i swobodnie posługiwać się nią w rozwiązywanych przez niego problemach. Fakt, że niezależnie pojawił się lepszy wynik nie wpływa na moją ocenę — w ocenie dorobku zawartego w doktoracie w moim odczuciu nie ma to kompletnie znaczenia, czy lepszy wynik pojawił się równolegle, czy też na przykład po 5 latach. Wynik zaprezentowany w omawianym doktoracie został zaprezentowany na *16th International Workshop on Approximation and Online Algorithms (WAOA 2018)*.[12]

Doktorat jest dość dobrze napisany. Jedyną uwagą, którą można tu wskazać, to że tekst stylem bardziej przypominał zbiór publikacji konferencyjnych niż kompletną i spójną monografię. Nie mniej jednak zasadniczo nie wpływa to na merytoryczną ocenę rozprawy.

Niewątpliwie najmocniejszym fragmentem omawianego doktoratu jest Rozdział 2. Użyte metody są najbardziej złożone, a rezultaty w największym stopniu wpływają na postęp w rozważanej tematyce. Może świadczyć o tym choćby to, że zaowocowało to poprawieniem wyniku dotyczącym problemu szeregowania zadań na niepowiązanych maszynach po 25 latach przestoju. W efekcie wyniki zostały zaprezentowane w 2021 roku na konfencji *ICALP*, która jest zaliczana do kategorii A na *Computing Research and Education (CORE) Conference Portal*.[13] Wyniki zaprezentowane w pozostałych rozdziałach są już mniejszej wagi, niemniej jednak wciąż ciekawe i rzucające nowe światło w rozważanej tematyce. Zostały one opublikowane w roku 2018 na konferencjach *SIROCCO* oraz *WAOA* — obie są zaliczane do kategorii B na portalu *CORE*.[13]

Biorąc pod uwagę powyższe argumenty, nie mam najmniejszej wątpliwości, że przedłożona rozprawa doktorska spełnia wymogi ustawowe i wnoszę o dopuszczenie Artura Kraski do dalszych etapów przewodu doktorskiego.

Signed by / Podpisano przez:
Bartłomiej Emil Bosek
Date / Data: 2022-03-20 23:47

Bartłomiej Bosek

[11] Y. Azar and A. Jacob-Fanani. Deterministic min-cost matching with delays. *In Proc. 16th Workshop on Approximation and Online Algorithms (WAOA)*, 21-35, 2018.

[12] M. Bienkowski, A. Kraska, H. Liu, and P. Schmidt. A primal-dual online deterministic algorithm for matching with delays. *In Proc. 16th Workshop on Approximation and Online Algorithms (WAOA)*, 51-68, 2018.

[13] `http://portal.core.edu.au/conf-ranks/`