PROGRAMOWANIE DYNAMICZNE Z GWARANCJAMI

Autoreferat rozprawy doktorskiej

Autor:
Karol Węgrzycki
Instytut Informatyki
Uniwersytet Warszawski

Promotor:
dr. hab Marcin Mucha
Instytut Informatyki
Uniwersytet Warszawski

1 Wstęp

Rozwiązanie konkretnego algorytmicznego problemu jest pracochłonnym zadaniem. Znalezienie poprawnego i efektywnego algorytmu często wymaga poświęcenia ogromnej ilości zasobów. W związku z tym, po zakończeniu tego procesu chcielibyśmy mieć gwarancje, że nasze rozwiązanie nie może zostać znacząco usprawnione.

W rozprawie doktorskiej staramy się rozwiązać ten problem.

2 Programowanie dynamiczne w problemach aproksymacyjnych

Problem plecakowy jest jednym z najbardziej fundamentalnych problemów optymalizacyjnych [32]. Zdefiniowany jest następująco:

Definicja 2.1 (Problem Plecakowy)**.** *Dany jest zbiór n przedmiotów $E_n = \{1, \ldots, n\}$ oraz pojemność plecaka $t \in \mathbb{N}$. Przedmiot j ma nieujemną wagę $w_j \in \mathbb{N}$ oraz nieujemną wartość $v_j \in \mathbb{N}$. Znajdź podzbiór przedmiotów $E \subseteq E_n$ o jak największej wartości $v(E) = \sum_{i \in E} v_i$ taki, że ich sumaryczna waga $w(E) = \sum_{i \in E} w_i$ nie przekracza pojemności t.*

Problem plecakowy jest NP-zupełny [27]. Wariant problemu plecakowego w którym zadaniem jest znalezienie *multizbioru* elementów nazywa się Unbounded Knapsack. Niech $\Sigma(S)$ oznacza sumę elementów zbioru S. Problem Subset Sum jest zdefiniowany następująco:

Definicja 2.2 (Subset Sum)**.** *Dany jest zbiór $S \subset \mathbb{N}$ składający się z n liczb oraz liczba całkowita t. Znajdź podzbiór $S' \subseteq S$ o największej możliwej $\Sigma(S') \leq t$.*

Problem Subset Sum jest specjalnym przypadkiem problemu plecakowego w którym wagi przedmiotów są równe ich wartością. Pomimo tego uproszczenia problem Subset Sum nadal pozostaje NP-trudny. W szczególności, nawet jeśli ograniczmy się do przypadku gdy $t = \Sigma(S)/2$, problem dalej pozostaje NP-trudny. Ten szczególny przypadek problemu Subset Sum nosi nazwę Partition:

Definicja 2.3 (Partition)**.** *Dany jest zbiór $S \subset \mathbb{N}$ składający się z n liczb. Znajdź podzbiór $S' \subseteq S$, taki że $\Sigma(S') \leq \Sigma(S)/2$ o największej możliwej $\Sigma(S')$.*

Problem Partition znajduje zastosowanie w szeregowaniu [26], kryptografii [33], a nawet w teorii gier [21, 34].

Najczęstszym sposobem radzenia sobie z NP-trudnością jest aproksymacja. Pierwszą aproksymację dla problemu plecakowego (i w szczególności dla Subset Sum i Partition) zaproponowali Ibarra i Kim [22] w 1975. Ich algorytm działa w czasie $\mathcal{O}(n/\varepsilon^2)$. Obecnie, po wielu poprawkach [28] najszybsze algorytmy to: dla problemu Partition $\mathcal{O}(\min\{n/\varepsilon, n+1/\varepsilon^2\})$ [19], dla problemu Subset Sum $\mathcal{O}(\min\{n/\varepsilon, n+1/\varepsilon^2 \log(1/\varepsilon)\})$ [29], oraz dla problemu plecakowego $\widetilde{\mathcal{O}}(n+1/\varepsilon^{9/4})$ [25].

Warto zauważyć, że wszystkie dotychczasowe algorytmy dla tych problemów działają w czasie $\Omega((n+1/\varepsilon)^2)$.

2.1 Nasze wyniki

Pokazujemy, ze dla problemu Partition można przyśpieszyć kwadratową aproksymację i tym samym poprawić algorytm $\widetilde{\mathcal{O}}(n+1/\varepsilon^2)$ Gensa i Levnera [19].

Twierdzenie 2.4. *Problem Partition można aproksymować w czasie* $\widetilde{\mathcal{O}}(n+1/\varepsilon^{\frac{5}{3}})$.

Dodatkowo wykluczamy podobną poprawę dla problemu plecakowego, poprzez udowodnienie dla niego ograniczeń dolnych.

3 Problemy równoważne z Min-Plus Convolution

W tej części rozprawy, badamy ograniczenia dolne przy innych założeniach niż P≠NP. Niedawno badacze zaproponowali kilka użytecznych założeń dla problemów w P. Jedną z pierwszych takich hipotez jest założenie, że dla problemu 3SUM nie istnieje algorytm działający w czasie $\mathcal{O}(n^{2-\varepsilon})$. Z tego założenia wynikają ograniczenia dolne dla problemów geometrycznych [18] oraz problemów dynamicznych [38]. Kolejne założenie mówi, że dla problemu All-Pairs Shortest Path (APSP) nie istnieje algorytm działający w czasie $\mathcal{O}(n^{3-\varepsilon})$. Z tego założenia wynika trudność znalezienia promienia, mediany i wielu innych charakterystyk grafów [50]. Jeszcze innym założeniem jest silna hipoteza wykładniczego czasu (ang. Strong Exponential Time Hypothesis) (SETH) [23, 24]. Założenie to było badane pod kątem algorytmów parametryzowanych [8, 31] a ostatnio zostało użyte do pokazania trudności problemów w P [49], na przykład dla problemu policzenia odległości edycyjnej [4] oraz najdłuższego wspólnego podciągu [1, 6].

W rozprawie proponujemy kolejne założenie trudności.

Definicja 3.1 (Min-Plus Convolution)**.** *W problemie Min-Plus Convolution dane są ciągi* $(a[i])_{i=0}^{n-1}$,$(b[i])_{i=0}^{n-1}$. *Zwróć ciąg* $(c[i])_{i=0}^{n-1}$, *takie że* $c[k] = \min_{i+j=k}(a[i]+b[j])$.

Podobnie do założenia o trudności problemu APSP [51], zezwalamy na czynniki $\mathrm{polylog}(W)$ w definicji algorytmów podkwadratowych.

Założenie 3.2. *Nie istnieje algorytm działający w czasie* $\mathcal{O}(n^{2-\varepsilon}\mathrm{polylog}(W))$ *dla problemu Min-Plus Convolution dla żadnego stałego* $\varepsilon > 0$.

Badamy trzy główne kategorie problemów. Pierwsza kategoria to klasyczny problem plecakowy i jego wariant Unbounded Knapsack. Pokazujemy, że te problemy są równoważne problemowi Min-Plus Convolution. Wynik ten jest zaskakujący, ponieważ dla problemu Subset Sum, który jest szczególnym przypadkiem problemu plecakowego istnieją algorytmy działające w czasie $\widetilde{\mathcal{O}}(n+t)$ [5]. Drugą kategorią są problemy luźno związane z problemem Min-Plus Convolution takie jak decyzyjna wersja problemu Min-Plus Convolution oraz problemy związane z subaddytywnością. Ostatnią kategorie stanowią problemy, dla których relacja z Min-Plus Convolution była znana wcześniej. Dla nich upraszczamy obecnie znane redukcje.

3.1 Nasze wyniki

Głównym wkładem tej części rozprawy jest udowodnienie, że następujące problemy są równoważne.

Twierdzenie 3.3. *Poniższe stwierdzenia są równoważne:*

1. *Problem Max-Plus Convolution można rozwiązać w* $\mathcal{O}(n^{2-\varepsilon})$ *gdy* $\varepsilon > 0$.
2. *Problem Max-Plus Convolution UpperBound można rozwiązać w* $\mathcal{O}(n^{2-\varepsilon})$ *gdy* $\varepsilon > 0$.
3. *Problem SuperAdditivity Testing można rozwiązać w* $\mathcal{O}(n^{2-\varepsilon})$ *gdy* $\varepsilon > 0$.
4. *Problem Unbounded Knapsack można rozwiązać w* $\mathcal{O}((n+t)^{2-\varepsilon})$ *gdy* $\varepsilon > 0$.
5. *Problem plecakowy można rozwiązać w* $\mathcal{O}((n+t)^{2-\varepsilon})$ *gdy* $\varepsilon > 0$.

(zezwalamy na zrandomizowane algorytmy).

Dodatkowo, przyśpieszamy obecnie znane schematy aproksymacyjne dla problemu Min-Plus Convolution. Używając podobnych technik pokazujemy szybsze algorytmy aproksymacyjne dla problemów 3SUM i Tree Sparsity.

4 Równoważność aproksymacji $(\min, +)$ z dokładnym $(\min, \max)$

Skalowanie jest jedną z najbardziej podstawowych technik w algorytmice. Żeby zaprezentować tą technikę wyobraźmy sobie jakiś problem ważony, z wagami w przedziale $\{1, \ldots, W\}$. Skalowanie to po prostu uproszczenie sobie zadania w następujący sposób: rozważmy po kolei każdy z $\log W$ bitów wag a następnie postarajmy się rozwiązać problem w wersji nieważonej. Na końcu pozostaje jedynie sprytnie połączyć rozwiązania dla każdej ze skal.

Skalowanie okazało się bardzo użyteczną techniką dla problemów grafowych [13, 16, 17, 20, 37]. Na przykład, algorytm skalujący pozwolił rozwiązać problem maksymalnych ważonych skojarzeń w czasie $\mathcal{O}(m\sqrt{n}\log(nW))$ [17] (ostatnio, udało się rozwiązać ten problem w czasie $\mathcal{O}(m\sqrt{n}\log W)$ [13]).

Czasami jednak, algorytmy skalujące mogą okazać się dużo wolniejsze niż alternatywne podejścia, ponieważ wymagają narzutu w postaci czynnika $\log W$ w czasie działania. W praktyce, wagi dane są jako liczby zmiennoprzecinkowe więc czynnik $\log W$ może być nawet rzędu n. Sprawia to, że niekiedy nawet naiwne algorytmy działają szybciej niż zaawansowane algorytmy skalujące.

Z tego powodu, kładzie się silny nacisk na badanie algorytmów *silnie wielomianowych* [36, 42, 44, 46]. Mówimy, że algorytm działa w czasie silnie wielomianowym, jeśli ilość operacji arytmetycznych nie zależy od W (maksymalnej wartości arytmetycznej). Na przykład, obecnie znany najszybszy silnie wielomianowy algorytm dla problemu maksymalnego ważonego skojarzenia działa w czasie $\widetilde{\mathcal{O}}(nm)$ [14, 45]. (Przyjmujemy, że $\widetilde{\mathcal{O}}(T) = \mathcal{O}(T \operatorname{polylog} T)$, w szczególności $\widetilde{\mathcal{O}}(n^c)$ nie ukrywa czynników $\log W$.)

W tej rozprawie staramy się pokazać silnie wielomianowe algorytmy, które są tak samo szybkie jak najlepsze znane słabo wielomianowe algorytmy (ale bez czynnika $\log W$).

Udaje się nam osiągnąć ten cel dla dużej klasy algorytmów aproksymacyjnych. Proponujemy nowe, algorytmiczne techniki, które pozwalają nam na sprowadzenie problemów aproksymacyjnych w półpierścieniu $(\min, +)$ do dokładnych problemów w półpierścieniu $(\min, \max)$. W szczególności zajmiemy się następującymi problemami:

- **Problem policzenia najkrótszych ścieżek:** W problemie *All-Pairs Shortest Path* (APSP) dostajemy na wejściu skierowany, ważony graf. Nasze zadanie to zwrócić odległość pomiędzy każdą parą wierzchołków.

- **Produkty macierzowe:** Dane macierze $A, B \in \mathbb{R}_+^{n\times n}$, ich produkt nad półpierścieniem $(\oplus, \otimes)$ to macierz $C \in \mathbb{R}_+^{n\times n}$ gdzie $C[i,j] = \bigoplus_{1\le k\le n}(A[i,k] \otimes B[k,j])$. Naiwnie, produkt ten można obliczyć używając $\mathcal{O}(n^3)$ operacji w półpierścieniu. Dla konkretnego półpierścienia $(+, \cdot)$ ten problem to po prostu standardowe mnożenie macierzy, które można wykonać w czasie $\mathcal{O}(n^\omega) \le \mathcal{O}(n^{2.373})$ [30]. Problem *Min-Plus Product* to problem obliczenia produktu macierzy nad półpierścieniem $(\min, +)$.

- **Charakterystyki grafów:** Rozważamy charakterystyki grafów takie jak: średnica, promień, najlżejszy trójkąt i cykl o najlżejszej wadze.

Charakterystyki grafów i Min-Plus Product można zredukować do problemu APSP, więc dla wszystkich z wymienionych problemów istnieje algorytm $\widetilde{\mathcal{O}}(n^3)$ [15, 47]. Używając algorytmu Williamsa [48] można je przyśpieszyć do $n^3/2^{\Omega(\sqrt{\log n})}$. Ponadto, algorytm $\mathcal{O}(n^{3-\delta})$ dla chociażby jednego z wymienionych problemów automatycznie dałby algorytm $\mathcal{O}(n^{3-\delta'})$ dla wszystkich [2, 51]. Uważa się jednak, że żaden z nich nie może być rozwiązany w czasie prawdziwie podkubicznym [2, 51].

Zwick zaproponował schemat $(1+\varepsilon)$-aproksymacji dla problemu APSP działający w czasie $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon}\log W)$ [54]. Ten schemat aproksymacji daje analogiczne gwarancje dla problemu Min-Plus Product oraz wspomnianych charakterystyk grafów [2, 39]. Algorytm Zwicka jest prawie optymalny [1], poza czynnikami $\log W$ w czasie działania. To sugeruje następujące pytanie:

> *Czy problem APSP, obliczenie Min-Plus Product i obliczenie wspomnianych charakterystyk grafów mają silnie wielomianowe algorytmy aproksymacyjne działające w czasie $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon})$? Albo przynajmniej w czasie $\widetilde{\mathcal{O}}(n^{3-\delta}/\varepsilon)$ dla jakiejś $\delta > 0$?*

4.1 Nasze wyniki

W rozprawie odpowiadamy na powyższe pytanie twierdząco. Dla charakterystyk grafów pokazujemy silnie wielomianowy algorytm działający w czasie $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon})$.

Twierdzenie 4.1. *Średnica, promień, mediana, najlżejszy trójkąt, najlżejszy cykl w grafie skierowanym można aproksymować w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon})$.*

Gdy w problemie APSP ograniczymy się do grafów nieskierowanych jesteśmy w stanie dostać silnie wielomianowy algorytm $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon})$.

Twierdzenie 4.2. *$(1+\varepsilon)$-aproksymację APSP w nieskierowanych grafach można obliczyć w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(\frac{n^\omega}{\varepsilon})$.*

Dla APSP w dowolnych, ważonych grafach skierowanych pokazujemy prawdziwie podkubiczny, silnie wielomianowy schemat aproksymacji.

Twierdzenie 4.3. *$(1+\varepsilon)$-aproksymację dla problemu APSP w ważonych, skierowanych grafach można obliczyć w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(n^{\frac{\omega+3}{2}}/\varepsilon)$.*

Nasz algorytm aproksymacyjny dla skierowanego APSP to tak naprawdę redukcja z aproksymacji APSP do problemu obliczenia dokładnego Min-Max Product, tj. policzenia produktu macierzy w półpierścieniu $(\min, \max)$. Ten problem z kolei jest równoważny problemowi All-Pairs Bottleneck Path. Zarówno problem All-Pairs Bottleneck Path jak i Min-Max Product można rozwiązać w czasie $\widetilde{\mathcal{O}}(n^{\frac{\omega+3}{2}})$ [12].

Dodatkowo, pokazujemy redukcje w drugą stronę co daje równoważność pomiędzy aproksymacją problemu APSP i dokładnym algorytmem dla Min-Max Product. W szczególności poprawa czasu działania naszego algorytmu automatycznie dałaby szybszy algorytm dla problemu All-Pairs Bottleneck Path.

[1] Pełna wersja rozprawy zawiera dokładne wyjaśnienie tego stwierdzenia

Twierdzenie 4.4. *Dla dowolnego $c \geq 2$, poniższe zdania są równoważne:*

- *$(1+\varepsilon)$-aproksymację dla skierowanego APSP można znaleźć w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(n^c/\mathrm{poly}(\varepsilon))$,*
- *$(1+\varepsilon)$-aproksymację Min-Plus Product można znaleźć w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(n^c/\mathrm{poly}(\varepsilon))$,*
- *Problem Min-Max Product można rozwiązać w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(n^c)$,*
- *Problem All-Pairs Bottleneck Path można rozwiązać w silnie wielomianowym czasie $\widetilde{\mathcal{O}}(n^c)$.*

Nasze techniki automatycznie przekładają się na inne problemy w półpierścieniu $(\min, +)$. W szczególności pokazujemy silnie wielomianową aproksymację w czasie $\widetilde{\mathcal{O}}(\frac{n^{3/2}}{\sqrt{\varepsilon}})$ dla problemu Min-Plus Convolution i Tree Sparsity. Pokazujemy także analogiczną do Twierdzenia 4.4 równoważność pomiędzy aproksymacją Min-Plus Convolution a dokładnym problemem Min-Max Convolution.

5 Poprawa czasu działania dla problemów na grafach

Rozwiązanie APSP dokładnie, w czasie $\mathcal{O}(n^{3-\delta})$ dla pewnego $\delta > 0$ w ważonych skierowanych grafach jest ważnym problem otwartym [51]. W tej części rozprawy staramy się poprawić algorytmy oparte o technikę programowania dynamicznego dla wybranych problemów na grafach *nieważonych*.

Dla nieskierowanych, nieważonych grafów istnieje algorytm działający w czasie $\widetilde{\mathcal{O}}(n^\omega)$, gdzie $\omega < 2.373$ jest wykładnikiem najszybszego znanego algorytmu dla problemu mnożenia macierzy [43]. Dla skierowanych grafów Zwick [54] pokazał algorytm działający w czasie $\mathcal{O}(n^{2.575})$ oparty o mnożenie macierzy prostokątnych. Dodatkowo Zwick [54] zaproponował algorytm $\mathcal{O}(W^{0.68}n^{2.575})$ gdy wiemy, że wagi należą do zbioru $\{-W, \ldots, W\}$.

5.1 Nasze wyniki

W tej części rozprawy, rozważamy graf skierowany G o średnicy $D < \infty$. Wprowadzamy nową technikę, która pozwala na zliczanie cykli i nieprostych ścieżek w grafie G w czasie mnożenia macierzy. Nasze techniki bazują na szybkim algorytmie znajdującym dekompozycje Frobeniusa macierzy i mnożeniu macierzy Hankela przez wektor. Pozwala nam to rozwiązać efektywnie następujące problemy:

- All Nodes Shortest Cycles – dla każdego wierzchołka, zwróć długość najkrótszego cyklu, który go zawiera. Pokazujemy algorytm działający w czasie $\mathcal{O}(n^\omega)$. Poprawia to poprzedni algorytm działający w czasie $\mathcal{O}(n^{(\omega+3)/2})$ [52].
- Niech $S(c)$ oznacza zbiór wierzchołków leżących na cyklach o długości $\leq c$. Cygan, Gabow i Sankowski [9] pokazali, że dla ustalonego $c \in \{1, \ldots, n\}$, zbiór $S(c)$ można policzyć w czasie $\widetilde{\mathcal{O}}(n^\omega)$. Pokazujemy algorytm, który znajduje wszystkie $S(1), S(2), \ldots, S(D)$ w tym samym czasie.
- Prezentujemy algorytm $\widetilde{\mathcal{O}}(n^\omega)$ dla zliczania zamkniętych nieprostych ścieżek dla wszystkich długości w zbiorze $\{1, \ldots, D\}$. Poprawiamy tym samym algorytm [3], który zliczał wyłącznie zamknięte nieproste ścieżki o jednej, ustalonej długości.
- Usprawniamy algorytm zaproponowany przez Yustera i Zwicka [53]. Udowadniamy, że można przeprocesować graf skierowany w czasie $\widetilde{\mathcal{O}}(n^\omega)$ w taki sposób, aby odpowiedzieć na pytania o odległości pomiędzy parami wierzchołków w czasie liniowym. Nasz algorytm,

zużywając te same zasoby co algorytm Yustera i Zwicka [53] jest w stanie zwrócić w zapytaniu D liczb: $w^k_{u,v}$ dla $k \in \{1, \ldots, D\}$ (gdzie $w^k_{u,v}$ oznacza ilość różnych ścieżek nieprostych z wierzchołka u do v o długości dokładnie k).

- All Pairs All Walks – W tym problemie zadanie polega na zwróceniu tablicy liczb $w^k_{u,v}$ dla wszystkich $u, v \in G$ i $k \in \{1, \ldots, D\}$. Pokazujemy algorytm działający w czasie $\mathcal{O}(n^3)$. Poprawia to algorytm $\widetilde{\mathcal{O}}(Dn^\omega)$ dla dostatecznie dużych D.

Warto zwrócić uwagę, że w przeciwieństwie do obecnie znanych algorytmów (np. [9, 52]), nasze techniki pozwalają jedynie na zliczanie cykli i ścieżek nieprostych w grafach o długości ograniczonej przez średnice grafu $D < \infty$.

6 Artykuły składające się na tę rozprawę

Prace składające się na główną część rozprawy zostały zaprezentowane na następujących konferencjach:

- *Improved Distance Queries and Cycle Counting by Frobenius Normal Form,* praca wspólna z Piotrem Sankowskim, opublikowana na konferencji **STACS 2017** [40].
- *On Problems Equivalent to (min, +)-Convolution*, praca wspólna z Marekiem Cyganem, Marcinem Muchą, Michałem Włodarczykiem, opublikowana na konferencji **ICALP 2017** [10].
- *A Subquadratic Approximation Scheme for Partition*, praca wspólna z Marcinem Muchą, Michałem Włodarczykiem, opublikowana na konferencji **SODA 2019** [35].
- *Approximating APSP without Scaling: Equivalence of Approximate Min-Plus and Exact Min-Max*, praca wspólna z Karlem Bringmannem i Marvinem Künnemannem, opublikowana na konferencji **STOC 2019** [7].

Pełne wersje wybranych prac zostały opublikowana w następujących czasopismach:

- *Improved Distance Queries and Cycle Counting by Frobenius Normal Form,* praca wspólna z Piotrem Sankowskim, zaproszona do Special Issue on Theoretical Aspects of Computer Science (STACS 2017), opublikowana w czasopiśmie **Theory of Computing Systems** [41].
- *On Problems Equivalent to (min, +)-Convolution*, praca wspólna z Marekiem Cyganem, Marcinem Muchą, Michałem Włodarczykiem, opublikowana w czasopiśmie **ACM Transactions on Algorithms** [11].

Bibliografia

[1] Amir Abboud, Arturs Backurs i Virginia Vassilevska Williams. „Tight Hardness Results for LCS and Other Sequence Similarity Measures". W: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015.* Red. Venkatesan Guruswami. IEEE Computer Society, 2015, s. 59–78.

[2] Amir Abboud, Fabrizio Grandoni i Virginia Vassilevska Williams. „Subcubic Equivalences Between Graph Centrality Problems, APSP and Diameter". W: *Proc. 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'15).* 2015, s. 1681–1697.

[3] Noga Alon, Raphael Yuster i Uri Zwick. „Finding and Counting Given Length Cycles". W: *Algorithmica* 17.3 (1997), s. 209–223.

[4] Arturs Backurs i Piotr Indyk. „Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false)". W: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*. Red. Rocco A. Servedio i Ronitt Rubinfeld. ACM, 2015, s. 51–58.

[5] Karl Bringmann. „A near-linear pseudopolynomial time algorithm for subset sum". W: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2017, s. 1073–1084.

[6] Karl Bringmann i Marvin Künnemann. „Quadratic Conditional Lower Bounds for String Problems and Dynamic Time Warping". W: *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*. Red. Venkatesan Guruswami. IEEE Computer Society, 2015, s. 79–97.

[7] Karl Bringmann, Marvin Künnemann i Karol Wegrzycki. „Approximating APSP without scaling: equivalence of approximate min-plus and exact min-max". W: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. Red. Moses Charikar i Edith Cohen. ACM, 2019, s. 943–954.

[8] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk i Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[9] Marek Cygan, Harold N. Gabow i Piotr Sankowski. „Algorithmic Applications of Baur-Strassen's Theorem: Shortest Cycles, Diameter, and Matchings". W: *J. ACM* 62.4 (2015), s. 28.

[10] Marek Cygan, Marcin Mucha, Karol Wegrzycki i Michal Wlodarczyk. „On Problems Equivalent to (min, +)-Convolution". W: *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*. Red. Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn i Anca Muscholl. T. 80. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 22:1–22:15.

[11] Marek Cygan, Marcin Mucha, Karol Wegrzycki i Michal Wlodarczyk. „On Problems Equivalent to (min, +)-Convolution". W: *ACM Trans. Algorithms* 15.1 (2019), 14:1–14:25.

[12] Ran Duan i Seth Pettie. „Fast algorithms for (max,min)-matrix multiplication and bottleneck shortest paths". W: *Proc. 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'09)*. 2009, s. 384–391.

[13] Ran Duan, Seth Pettie i Hsin-Hao Su. „Scaling Algorithms for Weighted Matching in General Graphs". W: *Proc. 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'17)*. 2017, s. 781–800.

[14] Jack Edmonds i Richard M. Karp. „Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems". W: *J. ACM* 19.2 (1972), s. 248–264.

[15] Robert W. Floyd. „Algorithm 97: Shortest path". W: *Commun. ACM* 5.6 (1962), s. 345.

[16] Harold N. Gabow i Robert Endre Tarjan. „Faster Scaling Algorithms for Network Problems". W: *SIAM J. Comput.* 18.5 (1989), s. 1013–1036.

[17] Harold N. Gabow i Robert Endre Tarjan. „Faster Scaling Algorithms for General Graph-Matching Problems". W: *J. ACM* 38.4 (1991), s. 815–853.

[18] Anka Gajentaan i Mark H. Overmars. „On a Class of $O(n^2)$ Problems in Computational Geometry". W: *Comput. Geom.* 5 (1995), s. 165–185.

[19] Georgii V Gens i Eugenii V Levner. „Fast approximation algorithms for knapsack type problems". W: *Optimization Techniques*. Springer, 1980, s. 185–194.

[20] Andrew V. Goldberg. „Scaling Algorithms for the Shortest Paths Problem". W: *SIAM J. Comput.* 24.3 (1995), s. 494–504.

[21] Brian Hayes. „Computing science: The easiest hard problem". W: *American Scientist* 90.2 (2002), s. 113–117.

[22] Oscar H. Ibarra i Chul E. Kim. „Fast Approximation Algorithms for the Knapsack and Sum of Subset Problems". W: *J. ACM* 22.4 (1975), s. 463–468.

[23] Russell Impagliazzo i Ramamohan Paturi. „On the Complexity of k-SAT". W: *J. Comput. Syst. Sci.* 62.2 (2001), s. 367–375.

[24] Russell Impagliazzo, Ramamohan Paturi i Francis Zane. „Which Problems Have Strongly Exponential Complexity?" W: *J. Comput. Syst. Sci.* 63.4 (2001), s. 512–530.

[25] Ce Jin. „An Improved FPTAS for 0-1 Knapsack". W: *CoRR* abs/1904.09562 (2019). arXiv: `1904.09562`.

[26] Edward G. Coffman Jr. i George S. Lueker. *Probabilistic analysis of packing and partitioning algorithms.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1991.

[27] Richard M. Karp. „Reducibility Among Combinatorial Problems". W: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York.* Red. Raymond E. Miller i James W. Thatcher. The IBM Research Symposia Series. Plenum Press, New York, 1972, s. 85–103.

[28] Hans Kellerer, Ulrich Pferschy i David Pisinger. *Introduction to NP-Completeness of knapsack problems.* Springer, 2004.

[29] Hans Kellerer, Ulrich Pferschy i Maria Grazia Speranza. „An Efficient Approximation Scheme for the Subset-Sum Problem". W: *Algorithms and Computation, 8th International Symposium, ISAAC '97, Singapore, December 17-19, 1997, Proceedings.* Red. Hon Wai Leong, Hiroshi Imai i Sanjay Jain. T. 1350. Lecture Notes in Computer Science. Springer, 1997, s. 394–403.

[30] François Le Gall. „Powers of tensors and fast matrix multiplication". W: *Proceedings of the 39th international symposium on symbolic and algebraic computation.* ACM. 2014, s. 296–303.

[31] Daniel Lokshtanov, Dániel Marx i Saket Saurabh. „Known Algorithms on Graphs of Bounded Treewidth Are Probably Optimal". W: *ACM Trans. Algorithms* 14.2 (2018), 13:1–13:30.

[32] George B Mathews. „On the partition of numbers". W: *Proceedings of the London Mathematical Society* 1.1 (1896), s. 486–490.

[33] Ralph C. Merkle i Martin E. Hellman. „Hiding information and signatures in trapdoor knapsacks". W: *IEEE Trans. Information Theory* 24.5 (1978), s. 525–530.

[34] Stephan Mertens. „The easiest hard problem: Number partitioning". W: *Computational Complexity and Statistical Physics* 125.2 (2006), s. 125–139.

[35] Marcin Mucha, Karol Wegrzycki i Michal Wlodarczyk. „A Subquadratic Approximation Scheme for Partition". W: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019.* Red. Timothy M. Chan. SIAM, 2019, s. 70–88.

[36] James B. Orlin. „A Faster Strongly Polynomial Minimum Cost Flow Algorithm". W: *Operations Research* 41.2 (1993), s. 338–350.

[37] James B. Orlin i Ravindra K. Ahuja. „New scaling algorithms for the assignment and minimum mean cycle problems". W: *Math. Program.* 54 (1992), s. 41–56.

[38] Mihai Patrascu. „Towards polynomial lower bounds for dynamic problems". W: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. Red. Leonard J. Schulman. ACM, 2010, s. 603–610.

[39] Liam Roditty i Virginia Vassilevska Williams. „Minimum Weight Cycles and Triangles: Equivalences and Algorithms". W: *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*. Red. Rafail Ostrovsky. IEEE Computer Society, 2011, s. 180–189.

[40] Piotr Sankowski i Karol Wegrzycki. „Improved Distance Queries and Cycle Counting by Frobenius Normal Form". W: *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*. Red. Heribert Vollmer i Brigitte Vallée. T. 66. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 56:1–56:14.

[41] Piotr Sankowski i Karol Węgrzycki. „Improved Distance Queries and Cycle Counting by Frobenius Normal Form". W: *Theory of Computing Systems* (2018).

[42] Alexander Schrijver. „A Combinatorial Algorithm Minimizing Submodular Functions in Strongly Polynomial Time". W: *J. Comb. Theory, Ser. B* 80.2 (2000), s. 346–355.

[43] Raimund Seidel. „On the All-Pairs-Shortest-Path Problem". W: *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*. Red. S. Rao Kosaraju, Mike Fellows, Avi Wigderson i John A. Ellis. ACM, 1992, s. 745–749.

[44] Éva Tardos. „A strongly polynomial minimum cost circulation algorithm". W: *Combinatorica* 5.3 (1985), s. 247–256.

[45] Mikkel Thorup. „Floats, Integers, and Single Source Shortest Paths". W: *J. Algorithms* 35.2 (2000), s. 189–201.

[46] László A. Végh. „A Strongly Polynomial Algorithm for Generalized Flow Maximization". W: *Math. Oper. Res.* 42.1 (2017), s. 179–211.

[47] Stephen Warshall. „A Theorem on Boolean Matrices". W: *J. ACM* 9.1 (1962), s. 11–12.

[48] R. Ryan Williams. „Faster All-Pairs Shortest Paths via Circuit Complexity". W: *SIAM J. Comput.* 47.5 (2018), s. 1965–1985.

[49] Ryan Williams. „A new algorithm for optimal 2-constraint satisfaction and its implications". W: *Theor. Comput. Sci.* 348.2-3 (2005), s. 357–365.

[50] Virginia Vassilevska Williams. „Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis (Invited Talk)". W: *10th International Symposium on Parameterized and Exact Computation, IPEC 2015, September 16-18, 2015, Patras, Greece*. Red. Thore Husfeldt i Iyad A. Kanj. T. 43. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015, s. 17–29.

[51] Virginia Vassilevska Williams i R. Ryan Williams. „Subcubic Equivalences Between Path, Matrix, and Triangle Problems". W: *J. ACM* 65.5 (2018), 27:1–27:38.

[52] Raphael Yuster. „A shortest cycle for each vertex of a graph". W: *Inf. Process. Lett.* 111.21-22 (2011), s. 1057–1061.

[53] Raphael Yuster i Uri Zwick. „Answering distance queries in directed graphs using fast matrix multiplication". W: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*. IEEE Computer Society, 2005, s. 389–396.

[54] Uri Zwick. „All pairs shortest paths using bridging sets and rectangular matrix multiplication". W: *J. ACM* 49.3 (2002), s. 289–317.