Zainteresowania Naukowe – Karol Węgrzycki

W dotychczasowej pracy naukowej zajmowałem się złożonością obliczeniową i konstruowaniem efektywnych algorytmów problemów związanych z pakowaniem. Zajmuję się zarówno dokładnymi algorytmami jak i algorytmami aproksymacyjnymi.

Dokładne problemy pakowania

W problemie Subset Sum mamy dany zbiór S liczb całkowitych oraz liczbę całkowitą t, a naszym zadaniem jest stwierdzić czy istnieje podzbiór $X \subseteq S$ o sumie równej dokładnie t. Problem Subset-Sum jest problemem NP-trudnym a nawet przy pewnych naturalnych założeniach nie da się rozwiązać go w czasie $2^{o(n)}$. Najszybszy znany algorytm dla problemu Subset Sum działa w czasie $\mathcal{O}^\star(2^{n/2})$ i w pamięci $\mathcal{O}^\star(2^{n/4})$ [14]. Wraz z Jesperem Nederlofem dla problemu Subset-Sum udało nam się poprawić tą złożoność pamięciową i zaproponowaliśmy algorytm działający w czasie $\mathcal{O}^\star(2^{n/2})$ i pamięci $\mathcal{O}^\star(2^{0.2499999n})$ [7].

Rozważałem także fundamentalny problem Bin Packingu który jest generalizacją problemu plecakowego. Mamy w nim dany zbiór n przedmiotów o różnych wagach oraz m pojemników o ograniczonej pojemności a naszym zadaniem jest znaleźć dopasowanie przedmiotów do pojemników w taki sposób żeby nie przekroczyć pojemności żadnego pojemnika.

Najszybszy obecnie znany algorytm dla problemu Bin Packing działa w czasie $\mathcal{O}^\star(2^n)$, nawet gdy liczba pojemników m jest stała [1]. W pracy [9] wraz z Jesperem Nederlofem, Jakubem Pawlewiczem i Celine Swennehuis zaproponowaliśmy algorytm działający w czasie $\mathcal{O}((2-\varepsilon_m)^n)$, który rozwiązuje problem Bin Packing dla m pojemników oraz $\varepsilon_m > 0$ jest dodatnią stałą zależną jedynie od m.

Kolejnym problemem, jaki badałem, jest problem Equal-Subset-Sum w którym dla danego zbioru musimy znaleźć dwa rozłączne, niepuste podzbiory o takiej samej sumie. Problem Equal-Subset-Sum jest NP-trudny i podobnie jak problem Subset Sum, nie da się otrzymać dokładnego algorytmu działającego w czasie podwykładniczym (przy pewnych naturalnych założeniach). Najszybszy znany algorytm działa w czasie $\mathcal{O}^\star(3^{n/2})$ i używa techniki Meet-in-the-Middle.

Wraz z Jesperem Nederlofem, Jakubem Pawlewiczem i Marcinem Muchą, pokazaliśmy algorytm dokładny dla problemu Equal-Subset-Sum działający w czasie $\mathcal{O}^\star(1.7088^n)$ [5], czym pobiliśmy naturalną granice wynikającą z użycia techniki Meet-in-the-Middle.

Aproksymacyjne problemy pakowania

Rozważałem także aproksymacyjne algorytmy problemów pakowania. Zwykle problemy plecakowe można przybliżać w bardzo efektywnym czasie. Dla problemu Partition, najszybszy znany algorytm aproksymacyjny działa w czasie $\widetilde{\mathcal{O}}(n+1/\varepsilon^2)$. Dla problemu plecakowego wiadomo, że nie istnieje algorytm aproksymacyjny działający w czasie $\widetilde{\mathcal{O}}(n+1/\varepsilon^{1.99})$ [3].

W pracy [6] wraz z Marcinem Muchą i Michałem Włodarczykiem pokazaliśmy, że problem Partition można aproksymować w czasie $\mathcal{O}^\star(n+1/\varepsilon^{5/3})$. Użyliśmy technik bazujących na kombinatoryce addytywnej.

Ograniczenia dolne dla problemów pakowania

Kolejnym projektem było wyznaczenie dolnych ograniczeń na czas rozwiązywania problemów pakowania. Wraz z Markiem Cyganem, Marcinem Muchą i Michałem Włodarczykiem [3] wyjaśniliśmy, dlaczego najlepszy znany algorytm dla problemu plecakowego oparty na programowaniu dynamicznym nie został istotnie ulepszony od ponad 50 lat. Udało nam się scharakteryzować nową klasę trudności problemów wielomianowych i pokazaliśmy, że prawdopodobnie pseudowielomianowy algorytm dla problemu plecakowego jest optymalny. Nasze wyniki znalazły także zastosowanie w dowodzeniu trudności aproksymacji.

Ostatnio w pracy [13] wraz z Adamem Polakiem oraz Larsem Rohwedderem zaproponowaliśmy pseudowielomianowe algorytmy dla problemu plecakowego, które działają szybciej niż klasyczny algorytm gdy przedmioty mają bardzo małe wagi.

Problemy odległości w grafach

Interesuje się także problemem znajdywania najkrótszych odległości pomiędzy każdą parą krawędzi w ważonych grafach skierowanych czyli problemem All-Pairs Shortest Path (APSP). Problem APSP można rozwiązać w czasie $\mathcal{O}(n^{3-o(1)})$ i uważa się, że nie da się go rozwiązać istotnie szybciej [16]. Najszybszy algorytm aproksymacyjny działa w czasie $\mathcal{O}(\frac{n^\omega}{\varepsilon}\log W)$ [17], gdzie $\omega \leq 2.38$ to wykładnik najszybszego obecnie znanego algorytmu dla mnożenia macierzy. Moim celem było poprawienie tego algorytmu lub wytłumaczenie dlaczego jest to niemożliwe. Dodatkowo pracowaniem nad problemem znalezienia największej odległości w grafie. Podobnie jak dla problemu APSP, najszybszy znany algorytm aproksymacyjny działa w czasie $\mathcal{O}(\frac{n^\omega}{\varepsilon}\log W)$.

Zaproponowanie algorytmu $\mathcal{O}(\frac{n^{\omega-\delta}}{\varepsilon^{\mathcal{O}(1)}})$ dla dowolnej $\delta > 0$ automatycznie dałby znacząco szybszy algorytm dla mnożenia macierzy. Co więcej, algorytm $\mathcal{O}(\frac{n^\omega}{\varepsilon^{o(1)}})$ zaprzeczyłby hipotezie APSP. W pracy [2] wraz z Karlem Bringmannem i Marvinem Künnemannem poprawiliśmy czynnik $\log W$ i udowodniliśmy, że średnice, medianę i APSP w grafach nieskierowanych można obliczyć w czasie $\widetilde{\mathcal{O}}(n^\omega/\varepsilon)$.

Dodatkowo dla APSP w skierowanych grafach pokazaliśmy, że problem silnie wielomianowej aproksymacji dla APSP jest równoważny problemu dokładnego policzenia $(\min,\max)$-produktu macierzy. Pozwala to automatycznie uzyskać silnie wielomianową aproksymacje dla problemu APSP w czasie $\widetilde{\mathcal{O}}(n^{\frac{\omega+3}{2}}/\varepsilon) = \widetilde{\mathcal{O}}(n^{2.69}/\varepsilon)$. Równoważność ta ilustruje, że poprawa naszego algorytmu automatycznie oznaczałaby poprawę dla wielu innych ważnych problemów.

Pozostałe wyniki

Badałem także problem znajdywania cyklu Hamiltona [8] oraz problem komiwojażera na płaszczyźnie [4]. Dodatkowo pracowałem nad nowymi technikami derandomizacji [10], aproksymacyjnymi algorytmami znajdowania najbliższych sąsiadów [11] oraz zaproponowałem modele rozchodzenia się plotek w sieciach społecznościowych [15, 12].

Literatura

[1] A. Björklund, T. Husfeldt, and M. Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.

[2] K. Bringmann, M. Künnemann, and K. Węgrzycki. Approximating APSP without scaling: equivalence of approximate min-plus and exact min-max. *STOC 2019*.

[3] M. Cygan, M. Mucha, K. Węgrzycki, and M. Wlodarczyk. On problems equivalent to (min, +)-convolution. *ACM Trans. Algorithms*, 2019.

[4] S. Kisfaludi-Bak, J. Nederlof, and K. Wegrzycki. A Gap-ETH-Tight Approximation Scheme for Euclidean TSP. *Accepted to FOCS 2021*. URL `https://arxiv.org/abs/2011.03778`.

[5] M. Mucha, J. Nederlof, J. Pawlewicz, and K. Węgrzycki. Equal-subset-sum faster than the meet-in-the-middle. *ESA 2019*, .

[6] M. Mucha, K. Węgrzycki, and M. Wlodarczyk. A subquadratic approximation scheme for partition. *SODA 2019*, .

[7] J. Nederlof and K. Wegrzycki. Improving Schroeppel and Shamir's Algorithm for Subset Sum via Orthogonal Vectors. *STOC 2021*.

[8] J. Nederlof, M. Pilipczuk, C. M. F. Swennenhuis, and K. Wegrzycki. Hamiltonian cycle parameterized by treedepth in single exponential time and polynomial space. In *WG 2021*.

[9] J. Nederlof, J. Pawlewicz, C. M. F. Swennenhuis, and K. Wegrzycki. A Faster Exponential Time Algorithm for Bin Packing With a Constant Number of Bins via Additive Combinatorics. In *SODA 2021*, 2021.

[10] J. Nederlof, M. Pilipczuk, C. M. F. Swennenhuis, and K. Wegrzycki. Isolation schemes for problems on decomposable graphs. *CoRR*, abs/2105.01465, 2021. URL `https://arxiv.org/abs/2105.01465`.

[11] A. Pacuk, P. Sankowski, K. Wegrzycki, and P. Wygocki. Locality-sensitive hashing without false negatives for l_p. In *COCOON 2016*, 2016.

[12] A. Pacuk, P. Sankowski, K. Wegrzycki, and P. Wygocki. There is something beyond the twitter network. In *Hypertext 2016*, 2016.

[13] A. Polak, L. Rohwedder, and K. Wegrzycki. Knapsack and subset sum with small items. In *ICALP 2021*, 2021.

[14] R. Schroeppel and A. Shamir. A T=$O(2^{n/2})$, S=$O(2^{n/4})$ Algorithm for Certain NP-Complete Problems. *SIAM J. Comput.*, 10(3):456–464, 1981.

[15] K. Wegrzycki, P. Sankowski, A. Pacuk, and P. Wygocki. Why do cascade sizes follow a power-law? In *WWW 2017*, pages 569–576, 2017.

[16] V. V. Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). *IPEC 2015*.

[17] U. Zwick. All pairs shortest paths in weighted directed graphs-exact and almost exact algorithms. *FOCS 1998*.