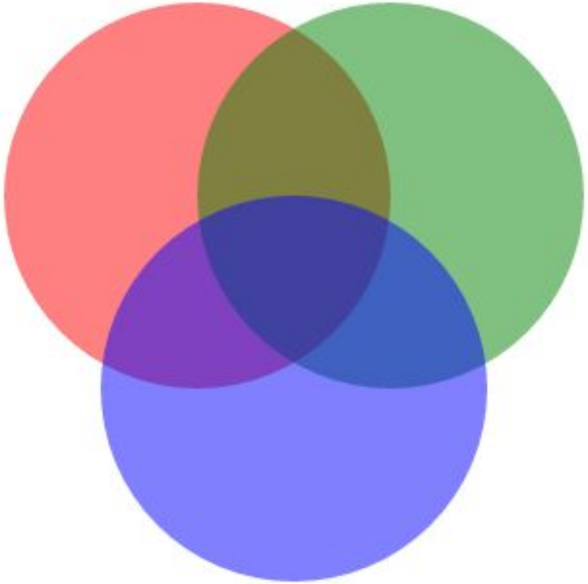
Co nowego w algorytmach pakujących i klastrujących?

Michał Włodarczyk

Różne paradygmaty algorytmiki

Algorytmy aproksymacyjne

Algorytmy parametryzowane



Złożoność drobnoziarnista

Złożoność drobnoziarnista

Fine-grained complexity

- ➔ Co teoria złożoności mówi o problemach w P?
- ➔ Czy algorytm $O(n^{10})$ to koniec poszukiwań?
- ➔ Jak zbudować teorię dostrzegającą drobniejszą skalę?

<u>Złożoność drobnoziarnista:</u>

Jakie jest optymalne c, takie że problem **A** posiada algorytm $O(n^c)$?

n - rozmiar instancji

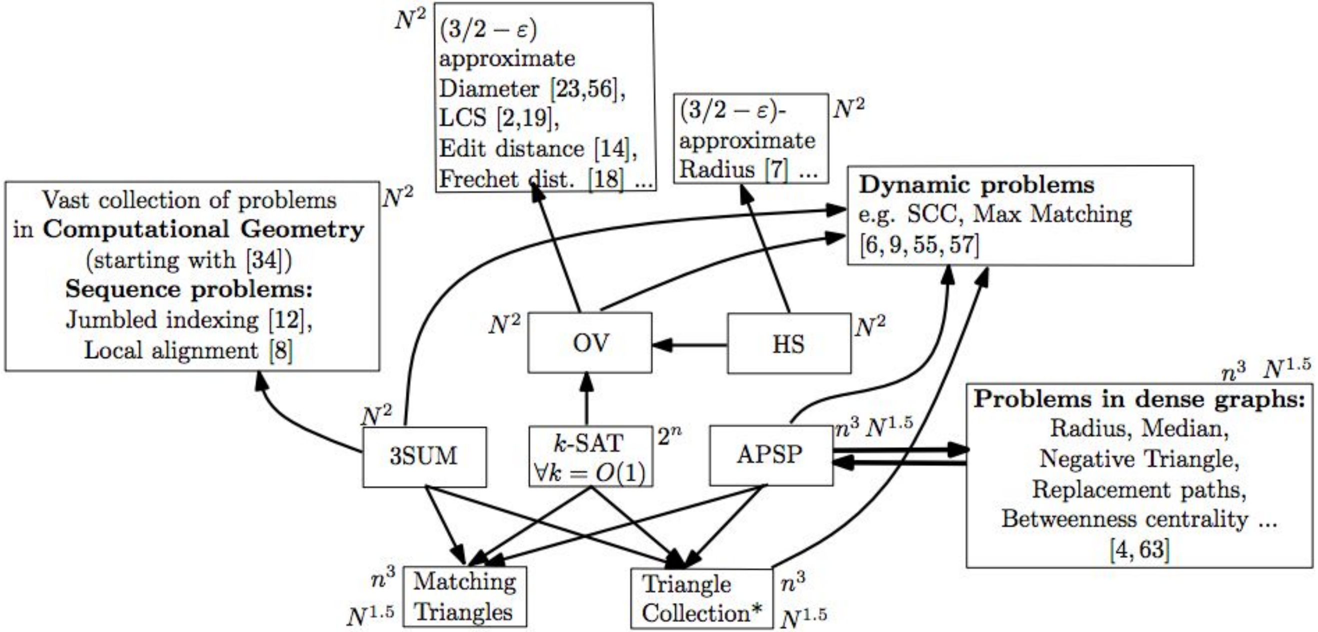
Złożoność liniowa:	$O(n)$
Złożoność kwadratowa:	$O(n^2)$
Złożoność kubiczna:	$O(n^3)$
Złożoność wielomianowa:	$O(n^c)$

Złożoność drobnoziarnista

Fine-grained complexity

<u>Złożoność drobnoziarnista:</u>

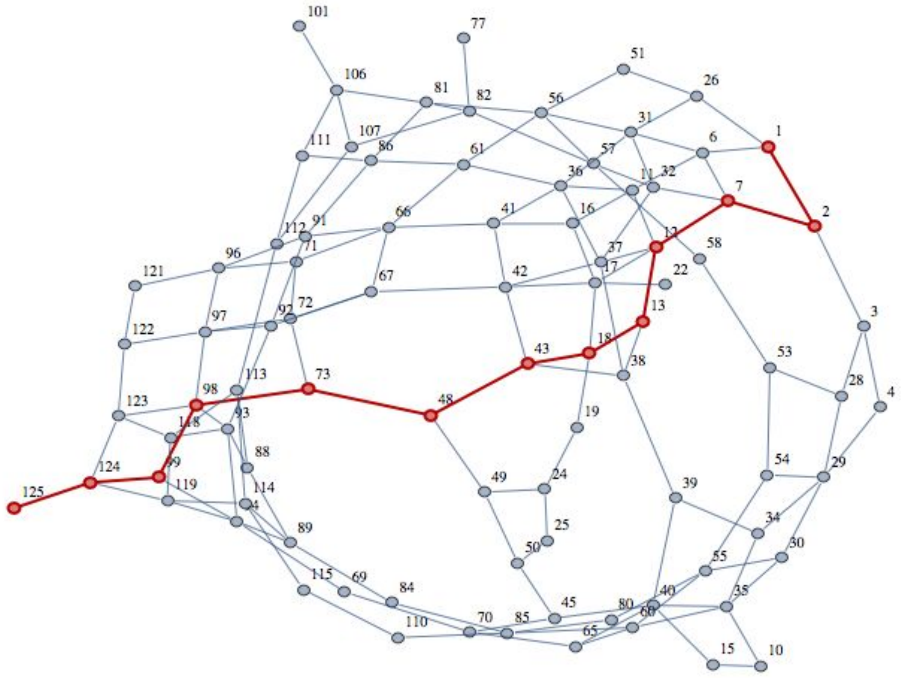
Jakie jest optymalne c, takie że problem **A** posiada algorytm O(n^c)?



Grafika: Williams '15

Wszystkie najkrótsze ścieżki

All Pairs Shortest Paths (APSP)



<u>Złożoność drobnoziarnista:</u>
Jakie jest optymalne c, takie że problem **A** posiada algorytm $O(n^c)$?

<u>All Pairs Shortest Paths:</u> dany jest ważony graf nieskierowany G(V, E),
n wierzchołków, m krawędzi

<u>Cel:</u> dla każdej pary s,t $\in$ V(G) obliczyć najkrótszą ścieżkę od s do t

Wszystkie najkrótsze ścieżki

All Pairs Shortest Paths (APSP)

- Algorytm $O(n^3)$ Warshall ‘62 Floyd ‘62
- Algorytm $O(n^2 \cdot \log(n) + n \cdot m)$ Johnson ‘77 (ale $m \sim n^2$)
- Drobna poprawa: $n^3 / 2^{\Omega(\sqrt{\log(n)})}$ Williams ‘14
- Czy można obliczyć APSP w $O(n^{2.99})$?

Podkubiczna równoważność: Williams Williams ‘15

- APSP w grafach skierowanych i nieskierowanych
- Najtańszy cykl
- Trójkąt o ujemnej wadze
- Promień grafu
- Mnożenie macierzy w półpierścieniu (min,+)

Złożoność drobnoziarnista:
Jakie jest optymalne c, takie że problem **A** posiada algorytm $O(n^c)$?

All Pairs Shortest Paths: dany jest ważony graf nieskierowany G(V, E),
n wierzchołków, m krawędzi

Cel: dla każdej pary $s,t \in V(G)$ obliczyć najkrótszą ścieżkę od s do t

Hipoteza APSP: żaden z tych problemów nie posiada algorytmu $O(n^{3-\varepsilon})$

Problem sumy podzbioru

Subset Sum

- NP-trudny (słabo)
- Pseudo-wielomianowy algorytm: $O(t \cdot n)$ Bellman '57
- Można lepiej: $O(t \cdot \log^c t)$ przy pomocy color coding oraz szybkiej konwolucji ciągów Bringmann '17
- Lepiej już się nie da: $O(t^{1-\varepsilon} \cdot 2^{o(n)})$ przeczy hipotezie SETH Abboud et al. '19

Konwolucja ciągów:

- Równoważna mnożeniu wielomianów stopnia n
- Pozwala policzyć algebraiczną sumę zbiorów A+B
- Naiwny algorytm $O(n^2)$
- Można lepiej: $O(n \cdot \log n)$ przy pomocy szybkiej transformaty Fouriera (FFT)

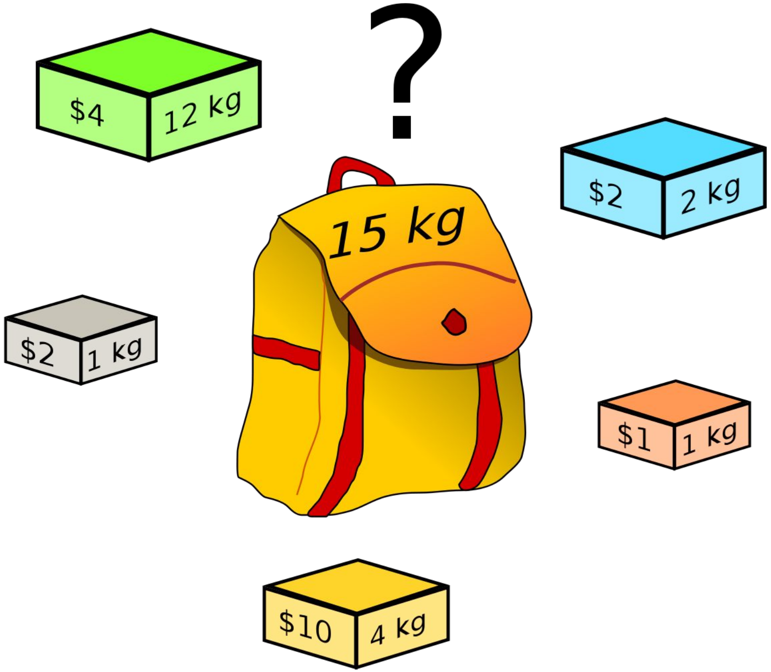
Subset Sum: dany zbiór n liczb $\{a_1\, a_2 \ldots a_n\}$ oraz liczba t
Pytanie: czy istnieje podzbiór $\{a_1\, a_2 \ldots a_n\}$ sumujacy się do t?

Konwolucja: dane 2 ciągi $(a_1\, a_2 \ldots a_n)$ $(b_1\, b_2 \ldots b_n)$
Zadanie: obliczyć $c_k = \Sigma_1^k\, a_i b_{k-i+1}$ dla $k = 1..n$

$c_1 = a_1 b_1$
$c_2 = a_1 b_2 + a_2 b_1$
$c_3 = a_1 b_3 + a_2 b_2 + a_3 b_1$
$c_4 = a_1 b_4 + a_2 b_3 + a_3 b_2 + a_4 b_1$
…

Problem plecakowy

Knapsack



<u>Knapsack:</u> dany zbiór n par

$\{(a_1 w_1)\ (a_2 w_2) \dots (a_n w_n)\}$ oraz liczba t

<u>Zadanie:</u> znaleźć podzbiór S spełniający

$\Sigma_S\ a_i \leq t$ maksymalizujący $\Sigma_S\ w_i$

Problem plecakowy
Knapsack

- NP-trudny (słabo)
- Pseudo-wielomianowy algorytm: $O(t \cdot n)$ Bellman '57
- Czy można lepiej?
- Potrzebny odpowiednik szybkiej konwolucji w półpierścieniu (min,+)

Czy (min,+)-konwolucja posiada algorytm $O(n^{2-\varepsilon})$?
- FFT wymaga struktury pierścienia
- Algorytm $O(n^{1.86})$ dla szczególnego przypadku
  Chan Lewenstein '15

<u>Knapsack:</u> dany zbiór n par $\{(a_1\, w_1)\ (a_2\, w_2) \dots (a_n\, w_n)\}$ oraz liczba t
<u>Zadanie:</u> znaleźć podzbiór S spełniający $\Sigma_S\, a_i \le t$ maksymalizujący $\Sigma_S\, w_i$

<u>(+,*)-konwolucja:</u> dane 2 ciągi
$(a_1\, a_2 \dots a_n)\ (b_1\, b_2 \dots b_n)$
<u>Zadanie:</u> obliczyć $c_k = \Sigma_1^k\, a_i \cdot b_{k-i+1}$
dla $k = 1 \dots n$

<u>(min,+)-konwolucja:</u> dane 2 ciągi
$(a_1\, a_2 \dots a_n)\ (b_1\, b_2 \dots b_n)$
<u>Zadanie:</u> obliczyć $c_k = \min_1^k (a_i + b_{k-i+1})$
dla $k = 1 \dots n$

Trudność (min,+)-konwolucji

Podkwadratowa równoważność: Cygan Mucha Węgrzycki W '17

- ➔ (min,+)-konwolucja
- ➔ Knapsack, Unbounded Knapsack
- ➔ Subaddytywność ciągów
- ➔ Problem maksymalnej sumy spójnego podciągu
- ➔ Bisekcja i problem najtańszego podgrafu na drzewach

Hipoteza (min,+)-conv: żaden z tych problemów nie posiada algorytmu $O(n^{2-\varepsilon})$

Hipoteza (min,+)-conv $\Longrightarrow$ Hipoteza APSP Bremner et al. '06

Knapsack: dany zbiór n par
$\{(a_1\ w_1)\ (a_2\ w_2)\ldots\ (a_n\ w_n)\}$ oraz liczba t
Zadanie: znaleźć podzbiór S spełniający
$\Sigma_S\ a_i \leq t$ maksymalizujący $\Sigma_S\ w_i$

(+,*)-konwolucja: dane 2 ciągi
$(a_1\ a_2 \ldots\ a_n)\ (b_1\ b_2 \ldots\ b_n)$
Zadanie: obliczyć $c_k = \Sigma_1^k\ a_i \cdot b_{k-i+1}$
dla $k = 1 \ldots n$

(min,+)-konwolucja: dane 2 ciągi
$(a_1\ a_2 \ldots\ a_n)\ (b_1\ b_2 \ldots\ b_n)$
Zadanie: obliczyć $c_k = \min_1^k\ (a_i + b_{k-i+1})$
dla $k = 1 \ldots n$

Algorytmy aproksymacyjne

- Szukamy rozwiązań przybliżonych dla problemów NP-trudnych
- Vertex Cover posiada 2-aproksymację
- Knapsack posiada $(1 + \varepsilon)$-aproksymację (wielomianowy schemat aproksymacji)

Knapsack: dany zbiór n par $\{(a_1\ w_1)\ (a_2\ w_2)\dots(a_n\ w_n)\}$ oraz liczba t
Zadanie: znaleźć podzbiór S spełniający $\Sigma_S\ a_i \leq t$ maksymalizujący $\Sigma_S\ w_i$

Algorytm ***c***-aproksymacyjny: działa w czasie wielomianowym i znajduje rozwiązanie nie gorsze niż ***c*** razy optimum

Złożoność aproksymacji

Jaka jest złożoność $(1+\varepsilon)$-aproksymacji?

- ➔ Knapsack: $O(n^2 / \varepsilon)$ Bellman '57
- ➔ Można lepiej: $O(n + \varepsilon^{-2.4})$ Chan '18

- ➔ Subset Sum: $O(n + \varepsilon^{-2})$ Kellerer et al. '97
- ➔ Czy można lepiej?
- ➔ Nie, przy założeniu hipotezy (min,+)-conv Bringmann '18
- ➔ Co jeśli chcemy podzielić po równo?

- ➔ Partition: $O(n + \varepsilon^{-2})$ Gens Levner '80
- ➔ Można lepiej: $O(n + \varepsilon^{-1.66})$ Mucha Węgrzycki W '19
- ➔ Składniki:
  - ◆ Subset Sum w $O(t \cdot \log^c t)$ Bringmann '17
  - ◆ Kombinatoryka addytywna Galil Margalit '91

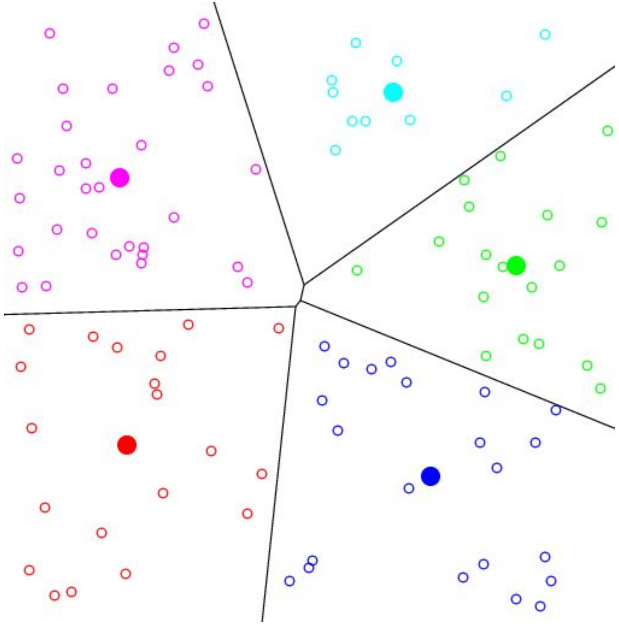
<u>Knapsack:</u> dany zbiór n par $\{(a_1\, w_1)\, (a_2\, w_2) \dots (a_n\, w_n)\}$ oraz liczba t
<u>Zadanie:</u> znaleźć podzbiór S spełniający $\Sigma_S\, a_i \leq t$ maksymalizujący $\Sigma_S\, w_i$

<u>Subset Sum:</u> dany zbiór n liczb $\{a_1\, a_2 \dots a_n\}$ oraz liczba t
<u>Zadanie:</u> znaleźć podzbiór $\{a_1\, a_2 \dots a_n\}$ o maksymalnej sumie nieprzekraczającej t

<u>Partition:</u> dany zbiór n liczb $\{a_1\, a_2 \dots a_n\}$
<u>Zadanie:</u> znaleźć podzbiór $\{a_1\, a_2 \dots a_n\}$ o maksymalnej sumie nieprzekraczającej $\Sigma\, a_i / 2$

Problem k-median

Wielomianowa aproksymacja



Algorytm **c**-aproksymacyjny: działa w czasie wielomianowym i znajduje rozwiązanie nie gorsze niż **c** razy optimum

k-median: dane n punktów $X = \{x_1 x_2 \dots x_n\}$ w metryce **d**

Zadanie: znaleźć zbiór k centrów $S \subseteq X$ i funkcję $\phi : X \longrightarrow S$ aby zminimalizować

$\Sigma\, \mathbf{d}(x_i , \phi(x_i))$

Problem k-median

Wielomianowa aproksymacja

k-median:

- ➔ NP-trudny
- ➔ Aproksymacja: 2.67 Byrka et al. '16
- ➔ Ograniczenie dolne: $1 + 2/e \approx 1.73$ (o ile P ≠ NP) Guha Khuller '99

Capacitated k-median:

- ➔ Aproksymacja: O(log k) Charikar et al. '98
- ➔ Czy da się lepiej?

<u>Algorytm **c**-aproksymacyjny</u>: działa w czasie wielomianowym i znajduje rozwiązanie nie gorsze niż **c** razy optimum

<u>k-median:</u> dane n punktów $X = \{x_1 x_2 \dots x_n\}$ w metryce **d**
<u>Zadanie:</u> znaleźć zbiór k centrów $S \subseteq X$ i funkcję $\phi : X \longrightarrow S$ aby zminimalizować $\Sigma\, \mathbf{d}(x_i, \phi(x_i))$

<u>Capacitated k-median:</u> jak wyżej, ale centra mają swoje pojemności: $|\, \phi^{-1}(x_i) \,| \le p_i$

Problem k-median

Parametryzowana aproksymacja

k-median:

- ➔ Prosty algorytm $O(n^k)$
- ➔ W[2]-trudny - brak algorytmu $f(k) \cdot n^c$
- ➔ Aproksymacja $1 + 2/e + \varepsilon$ w czasie $k^{O(k)} \cdot n^c$
  Cohen-Addad et al. '19
- ➔ Ograniczenie dolne: $1 + 2/e$
  (przy założeniu hipotezy GapETH) jw.

Capacitated k-median:

- ➔ Aproksymacja: $7 + \varepsilon$
  Adamczyk Byrka Marcinkowski Meesum W '19
- ➔ Można lepiej: $3 + \varepsilon$
  Cohen-Addad Li '19

<u>Algorytm parametryzowany</u>: dla instancji rozmiaru n i parametru k działa w czasie $f(k) \cdot n^c$, np. $2^k \cdot n^2$, $k! \cdot n^5$

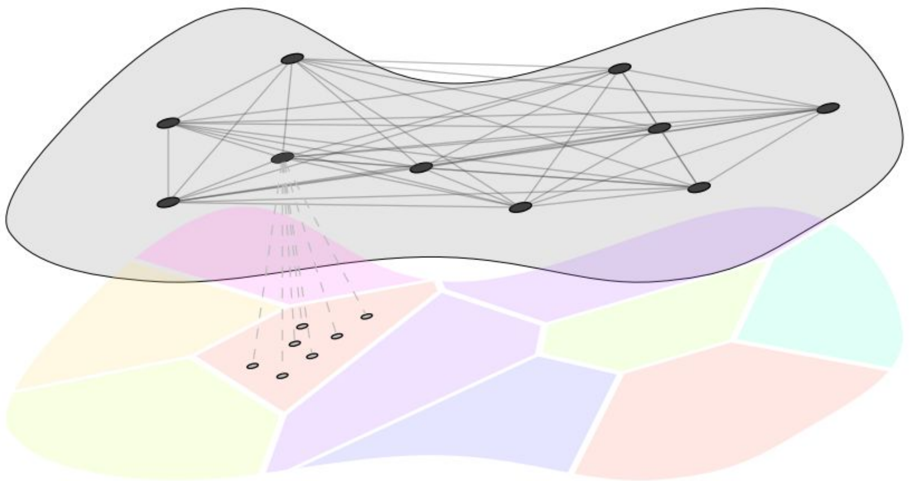
<u>k-median:</u> dane n punktów $X = \{x_1 x_2 \dots x_n\}$ w metryce **d**
<u>Zadanie:</u> znaleźć zbiór k centrów $S \subseteq X$ i funkcję $\phi : X \longrightarrow S$ aby zminimalizować $\Sigma\, \mathbf{d}(x_i, \phi(x_i))$

<u>Capacitated k-median:</u> jak wyżej, ale centra mają swoje pojemności: $|\, \phi^{-1}(x_i) \,| \leq p_i$

Capacitated k-median

Parametryzowana aproksymacja

- ➔ Zanurzenie w metrykę z *koroną* rozmiaru ~k
- ➔ Zanurzenie korony w drzewo: O(log k)-aproksymacja
- ➔ Liczba konfiguracji korony: $k^{O(k)} \longrightarrow$ stała aproksymacja



Algorytm parametryzowany: dla instancji rozmiaru n i parametru k działa w czasie f(k)·poly(n), np. $2^k \cdot n^2$, $k! \cdot n^5$

k-median: dane n punktów $X = \{x_1 x_2 \dots x_n\}$ w metryce **d**
Zadanie: znaleźć zbiór k centrów $S \subseteq X$ i funkcję $\phi : X \longrightarrow S$ aby zminimalizować $\Sigma\, \mathbf{d}(x_i , \phi(x_i))$

Capacitated k-median: jak wyżej, ale centra mają swoje pojemności: $|\, \phi^{-1} (x_i)\, | \leq p_i$

Dziękuję za uwagę