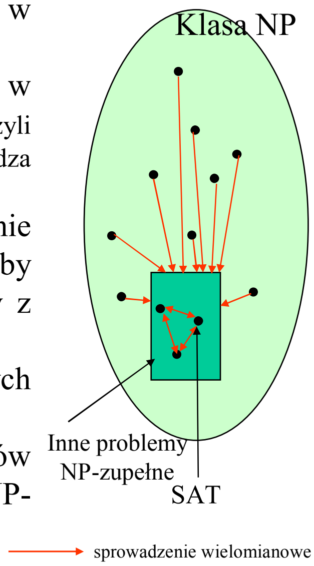
Problemy optymalizacyjne - zastosowania

`http://zajecia.jakubw.pl/nai`

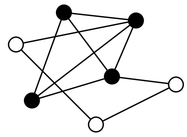
Problemy NP-zupełne (przypomnienie)

- **Klasa *P*** - problemy rozwiązywalne w czasie wielomianowym.
- **Klasa *NP*** - problemy rozwiązywalne w wielomianowym czasie na NDTM (czyli takie, których poprawność rozwiązania sprawdza się wielomianowo)
- SAT jest "uniwersalny", jego rozwiązanie w czasie wielomianowym pozwalałoby na rozwiązanie <u>wszystkich</u> problemów z klasy NP w czasie wielomianowym.
- Tego rodzaju problemów (nazywanych **NP-zupełnymi**), jest więcej!
- Nie znamy szybkich algorytmów rozwiązywania problemów NP-zupełnych.



KLIKI W GRAFIE

Niech G = (V, E) - dany graf.
Kliką nazywamy zbiór wierzchołków grafu G połączonych "każdy z każdym".



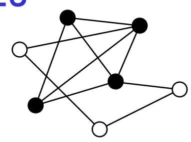
Czy w danym grafie istnieje klika rzędu k?

Problem istnienia kliki jest NP-zupełny

Sprowadzimy 3-SAT do problemu kliki.
Każdy literał a_i kodujemy jako jeden wierzchołek w grafie. Wierzchołki łączymy krawędzią, jeśli odpowiednie dwa literały należą do różnych klauzul i nie są wzajemnie sprzeczne (tzn. nie łączymy zmiennej i jej zaprzeczenia).

Niech k - liczba klauzul. Wtedy klika rzędu k w tak skonstruowanym grafie odpowiada wartościowaniu spełniającemu formułę.

SZUKANIE KLIKI MAKSYMALNEJ



Metody zachłanne:
jeden krok = dodanie do podzbioru kolejnego wierzchołka
jakość dodawanego wierzchołka = jego rząd

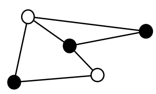
Metody oparte na sąsiedztwie:
rozwiązania sąsiednie różnią się jednym wierzchołkiem

Uwaga: musimy też oceniać rozwiązania nie będące klikami. Przykład (graf n-wierzchołkowy):
k - liczba wierzch. w podzbiorze
p - liczba krawędzi w podzbiorze

$$jakosc\_rozw. = k + n\frac{2p}{k(k-1)}$$

POKRYCIE WIERZCHOŁKOWE GRAFU

Dany jest graf G = (V, E). Znaleźć najmniejszy podzbiór wierzchołków taki, by każda krawędź kończyła się jednym z nich.



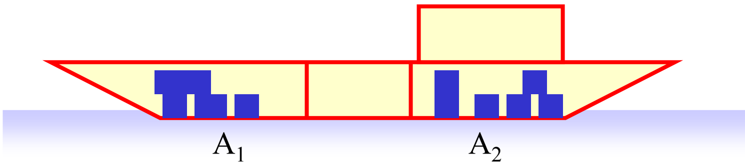
Metody zachłanne:
jeden krok = dodanie jednego wierzchołka
ocena wierzchołka = liczba nowo pokrytych krawędzi

Metody oparte na sąsiedztwie:
rozwiązania sąsiednie różnią się tylko jednym wierzchołkiem

Ocena uwzględnia też liczbę globalnie pokrytych krawędzi

Problem NP-zupełny.

PODZIAŁ ZBIORU



Mamy dany zbiór *n* wartości rzeczywistych $\{a_1, \ldots , a_n\}$.
Czy da się podzielić zbiór na dwa rozłączne podzbiory A_1 i A_2 takie, żeby suma wartości z A_1 równała się sumie wartości z A_2?

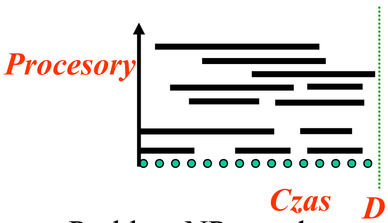
Metody zachłanne:
jeden krok = jedna para elementów
optymalizacja ze wzgl. na lokalnie najmniejszą różnicę wag

Metody oparte na sąsiedztwie:
rozwiązania sąsiednie różnią się o co najwyżej jeden lub dwa elementy

Problem NP-zupełny.

PLANOWANIE ZADAŃ

Dany jest zbiór zadań do wykonania (w dowolnej kolejności) o ustalonych długościach, oraz liczba *m* procesorów. Czy da się rozdzielić i rozplanować zadania tak, żeby się zmieścić w pewnym limicie czasu *D*?



Problem NP-zupełny.

Metody zachłanne:
jeden krok = dołożenie jednego zadania
przydzielamy zadanie pierwszemu procesorowi, który jest wolny (poczynając od najdłuższych zadań)

Metody oparte na sąsiedztwie:
rozwiązania sąsiednie różnią się przyporządkowaniem jednego zadania

Inna wersja - problem pakowania (bin packing): w zadaniu j.w. użyć jak najmniej procesorów. http://www.jakubw.pl/sci/binpack/

POKRYWANIE MACIERZY

Dana jest macierz zerojedynkowa $A=\{a_{ij}\}$ o rozmiarze $n*m$. Znaleźć najmniejszy podzbiór kolumn B taki, że w każdym wierszu co najmniej jedna jedynka należy do zbioru B.

Inaczej: dana jest lista bibliotek i lista książek, które dana biblioteka wypożycza. Znaleźć minimalny podzbiór bibliotek, oferujących łącznie ten sam komplet książek, co wszystkie.

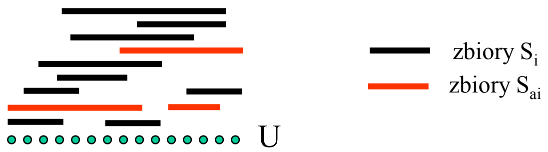
Dane jest zapotrzebowanie na pewne surowce (w sensie ich rodzajów, nie ilości) oraz lista dostawców, z których każdy ma w ofercie część surowców. Podpisać minimalną liczbę kontraktów zapewniającą otrzymywanie kompletu surowców.

Problem NP-zupełny.

Pokrycie zbioru - równoważne poprzedniemu

Dany jest zbiór U i rodzina jego podzbiorów $\{S_1, \dots, S_n\}$, dająca w sumie U. Znajź najmniejszą podrodzinę $\{S_{a1}, \dots, S_{ak}\}$ taką, że:

$$S_{a1} \vee \dots \vee S_{ak} = S_1 \vee \dots \vee S_n = U$$



Metody zachłanne:
jeden krok = jeden nowy element podrodziny
optymalizacja: liczba nowo pokrytych elementów U

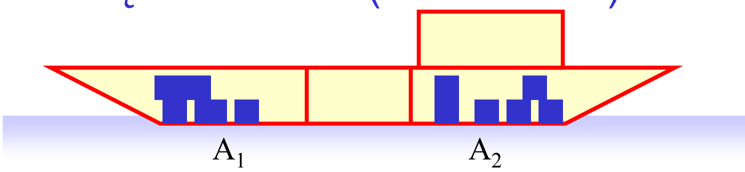
Metody oparte na sąsiedztwie:
rozwiązania sąsiednie to podrodziny różniące się jednym elementem.
Jakość: wielkość podrodziny, ale też stopień pokrycia U w przypadku rozwiązań niekompletnych.

SĄSIEDZTWO - ZASADY OGÓLNE

Metody typu przeszukiwanie tabu czy algorytm wspinaczki, bazują na pojęciu **sąsiedztwa** rozwiązań (punktów przestrzeni stanów). Może być ono zdefiniowane przez nas w zasadzie dowolnie, jednak należy trzymać się kilku zasad ogólnych:

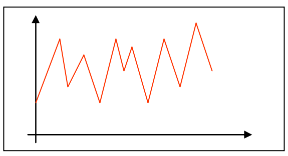
- **Sąsiedztwo nie powinno być zbyt liczne.**
  W praktyce: najwyżej kwadratowo liczne. Częścią wielu algorytmów jest pętla przeglądająca wszystkich sąsiadów, lub funkcja losująca jednego z nich.
- **Relacja sąsiedztwa powinna być spójna.**
  Tzn. do każdego rozwiązania powinniśmy dojść idąc od sąsiada do sąsiada. Inaczej część z nich będzie nieosiągalna.
- **Przejście do sąsiada powinno niewiele zmieniać funkcję celu.**
  Milczące założenie: w pobliżu dobrych rozwiązań warto szukać jeszcze lepszych - bez tego heurystyki są bezradne.

SĄSIEDZTWO (PRZYKŁAD)



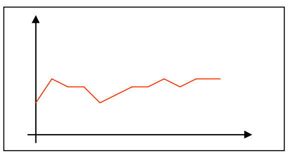
Mamy dany zbiór n wartości rzeczywistych $\{a_1, \ldots, a_n\}$. Podzielić zbiór na dwa rozłączne podzbiory A_1 i A_2 takie, żeby suma wartości z A_1 była bliska sumie wartości z A_2.

Funkcja celu: moduł różnicy sum zbiorów (minimalizujemy).



Sąsiednie rozwiązanie: przekładamy jeden pakunek.

Trudne do optymalizacji.



Sąsiednie rozwiązanie: zamieniamy parę pakunków.

Relacja niespójna.
Znacznie więcej sąsiadów.

Najlepiej połączyć te dwie definicje.