

Algorytmy metaheurystyczne – podsumowanie

dr hab. inż. Maciej Komosiński, prof. PP

Instytut Informatyki
Politechnika Poznańska
www.cs.put.poznan.pl/mkomosinski

Problem optymalizacji kombinatorycznej

- Problem optymalizacji kombinatorycznej jest problemem minimalizacji bądź **maksymalizacji** i charakteryzuje się zbiorem **instancji**
- Instancja problemu jest parą (S, f)
 - S oznacza zbiór wszystkich możliwych rozwiązań
 - f jest funkcją celu – odwzorowaniem definiowanym jako

$$f : S \rightarrow \mathbb{R}$$

Minimalizacja (koszt, *cost*) lub maksymalizacja (zysk, *gain*)

- W przypadku minimalizacji należy znaleźć takie rozwiązanie $i_{opt} \in S$ które spełnia

$$f(i_{opt}) \leq f(i) \text{ dla wszystkich } i \in S$$

- W przypadku maksymalizacji należy znaleźć takie rozwiązanie $i_{opt} \in S$ które spełnia

$$f(i_{opt}) \geq f(i) \text{ dla wszystkich } i \in S$$

- Takie rozwiązanie i_{opt} jest globalnie optymalne (globalnie minimalne lub maksymalne)

Problemy NP-trudne

- Czas znalezienia optimum rośnie bardzo szybko (wykładniczo) wraz ze wzrostem rozmiaru problemu
- W praktyce oznacza to, że nie można znaleźć rozwiązania optymalnego w realnym czasie dla zadań o rzeczywistych rozmiarach
- Wszystkie problemy NP-trudne są w pewnym sensie równoważne (niektóre można transformować do innych).

Klasyfikacja algorytmów optymalizacyjnych

- dokładne
 - przeszukiwanie wyczerpujące (*exhaustive*, *brute force*)
 - programowanie matematyczne (MP)
 - podziału i ograniczeń (B&B)
 - programowanie dynamiczne (DP)
- heurystyczne
 - specjalizowane – dla konkretnego problemu
 - uniwersalne – metaheurystyki
 - przeszukiwanie losowe (*random search*)
 - przeszukiwanie lokalne – algorytmy: zachłanny i stromy, symulowane wyżarzanie (SA), przeszukiwanie tabu (TS)
 - przeszukiwanie populacyjne – algorytmy: ewolucyjny, grupy cząstek (PSO), pszczoły (ABC), grawitacyjny (GSA), ...
 - mrówkowe (ACO)
 - hybrydowe
 - ...
- aproksymacyjne: wiemy o ile ich wynik jest gorszy od optimum

Co jest potrzebne do użycia/stworzenia algorytmu optymalizacji?

- 1 Reprezentacja rozwiązania (struktura danych)
- 2 Metoda zmieniania rozwiązanie lub generująca kolejne
- 3 Funkcja oceniająca rozwiązanie

Reprezentacja pojedynczego rozwiązania

- Ciąg binarny, np. SAT
- Reprezentacja zmiennoprzecinkowa, np. programowanie nieliniowe
- Permutacja, np. TSP, QAP
- Macierz
- Drzewo
- ...

Exhaustive search, brute force search

- Wymaga wygenerowania i sprawdzenia każdego rozwiązania dopuszczalnego – wady oczywiste
- Zaleta – proste
- Jedyne wymaganie – systematyczny sposób przeszukiwania S
- Jak to zrobić, zależy od reprezentacji

Exhaustive search – SAT

0000 – 0

0001 – 1

0010 – 2

0011 – 3

...

Zadanie: napisz program wypisujący wszystkie rozwiązania dla zadanego n .

Exhaustive search – TSP

- Permutacje. Niedopuszczalne jest np.
1-2-3-1-4
- Liczba permutacji – $n!$
- Rekurencyjne generowanie permutacji
 - ustalenie pierwszej pozycji
 - generowanie $(n - 1)!$ permutacji dla ustalonej pierwszej pozycji...
- Wersja bez rekurencji?

Zadanie: napisz program wypisujący wszystkie rozwiązania dla zadanego n .

Przeszukiwanie losowe

```
procedure PRZESZUKIWANIE_LOSOWE;  
begin  
  xbest:=GENERUJ_LOSOWO;  
  repeat  
    x:=GENERUJ_LOSOWO;  
    if  $f(x) \leq f(x_{best})$  then  
      xbest:=x;  
  until WARUNEK_STOPU;  
end;
```

Zadanie: dla SAT i TSP napisz programy wypisujące dla zadanego n jedno losowe rozwiązanie. Możesz korzystać tylko z funkcji $\text{random}(x)$.

Przeszukiwanie losowe

procedure PRZESZUKIWANIE_LOSOWE;

begin

 xbest:=GENERUJ_LOSOWO;

repeat

 x:=GENERUJ_LOSOWO; ← implementacja?

if $f(x) \leq f(x_{\text{best}})$ **then**

 xbest:=x;

until WARUNEK_STOPU;

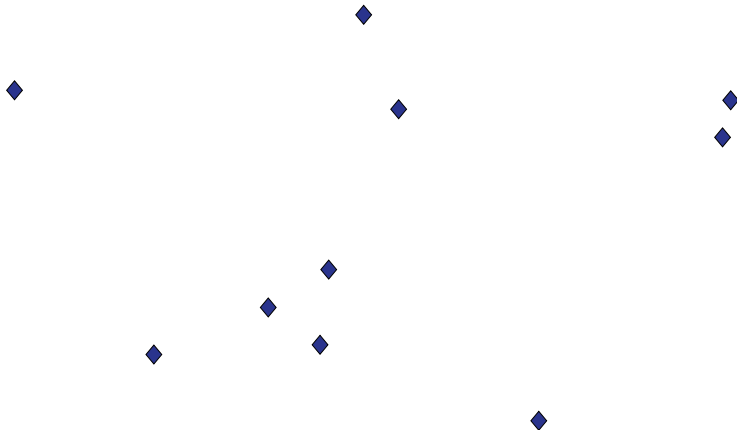
end;

Zadanie: dla SAT i TSP napisz programy wypisujące dla zadanego n jedno losowe rozwiązanie. Możesz korzystać tylko z funkcji $\text{random}(x)$.

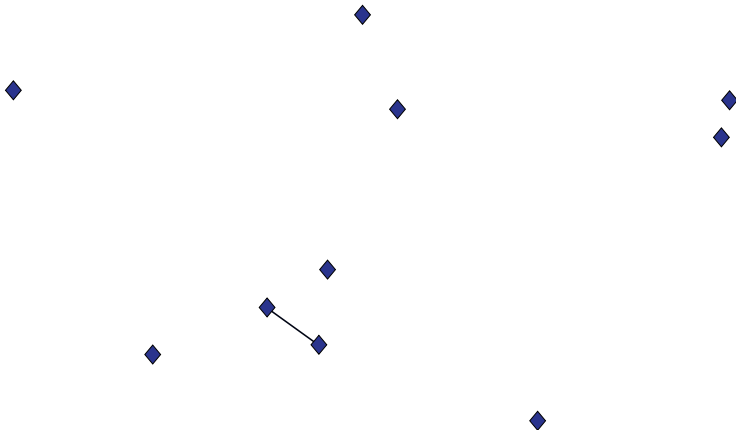
Metody przybliżone – heurystyki

- Cel – uzyskanie dobrych rozwiązań w rozsądnym czasie
- Podział:
 - specjalizowane (dedykowane)
 - uniwersalne – metaheurystyki

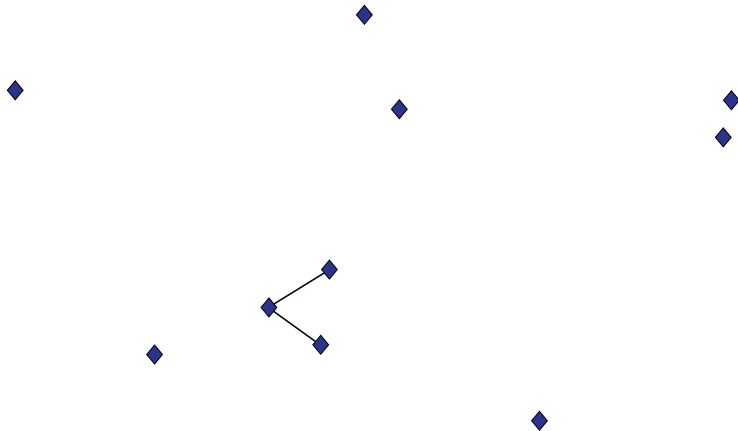
Przykład prostej heurystyki dla problemu komiwojażera



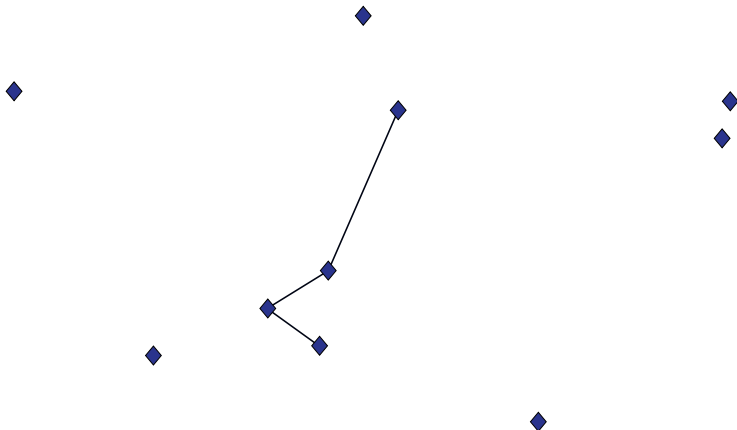
Przykład prostej heurystyki dla problemu komiwojażera



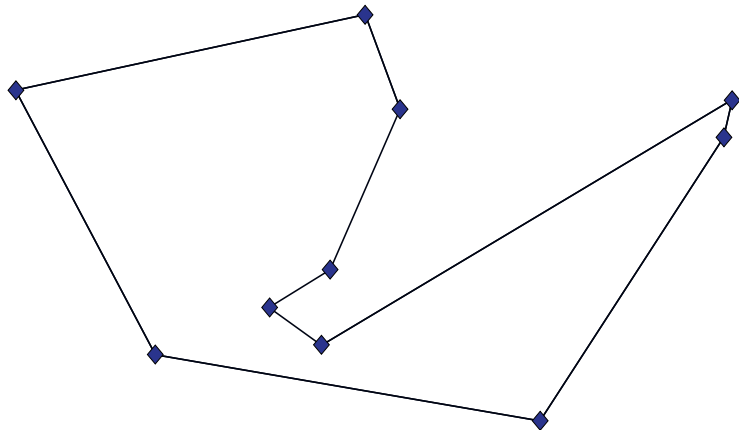
Przykład prostej heurystyki dla problemu komiwojażera



Przykład prostej heurystyki dla problemu komiwojażera



Przykład prostej heurystyki dla problemu komiwojażera



Cechy specjalizowanych heurystyk

- Z reguły bardzo krótki czas pracy.
- Wyniki zwykle dobre lub bardzo dobre, ale niekiedy dość odległe od optimum.
- Każda stworzona specjalnie dla konkretnego problemu (wymagana wiedza ekspercka).

Metaheurystyki

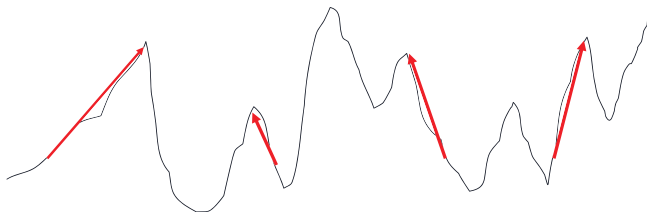
- Często bazują na analogiach do procesów ze świata rzeczywistego (fizyki, chemii, biologii), które można interpretować w kategoriach optymalizacji.
- Umożliwiają uzyskanie wyników bliskich optimum.
- Nie wymagają wiedzy eksperckiej o konkretnym problemie optymalizacji, choć takowa jest zaletą.
- Dla każdej metaheurystyki przypomnij sobie odpowiednią analogię turysty w górach.

Lokalna optymalizacja, wspinaczka

- Metoda stroma i zachłanna
- Pojęcie rozwiązania sąsiedniego
- Wymagania wobec sąsiedztwa
 - Niewielki (sensowny) rozmiar
 - Możliwość dojścia do dowolnego rozwiązania z każdego rozwiązania początkowego
 - Rozwiązania sąsiednie powinny być podobne

Wielokrotna lokalna optymalizacja

- Po osiągnięciu optimum lokalnego generujemy kolejne rozwiązanie startowe i ponownie uruchamiamy lokalną optymalizację



Symulowane wyżarzanie

- Analogia – powolne ochładzanie ciała stałego
- Związek z optymalizacją
 - Rozwiązanie – struktura ciała stałego
 - Funkcja celu – energia danej struktury
 - Sąsiedztwo – losowe perturbacje pod wpływem temperatury
- Możliwość akceptowania rozwiązań gorszych niż aktualne

Przeszukiwanie tabu

- Zasada zakazu powrotu do miejsc ostatnio odwiedzonych.
- Wybór najlepszego spośród nie zakazanych ruchów nawet, jeżeli prowadzi do gorszego rozwiązania.

Algorytmy Ewolucyjne (w tym genetyczne)

- Analogia – ewolucja populacji biologicznej
- Związek z optymalizacją
 - Rozwiązanie – osobnik
 - Funkcja celu – przystosowanie osobnika
 - Selekcja rozwiązań – walka o byt
 - Operator rekombinacji – rozmnażanie

Operator rekombinacji (krzyżowania)

- Konstruuje nowe rozwiązanie poprzez łączenie cech dwóch (lub więcej) rekombinowanych rozwiązań
- Może zawierać elementy losowe
- Założenie: dobre rozwiązania są do siebie w pewien sposób podobne, zatem zapewne mają pewne wspólne cechy, które korelują z jakością rozwiązań

Przykład operatora rekombinacji dla problemu planowania Systemu Dystrybucji

- Rozważaj po kolei każdy region
 - Jeżeli region był w obu rekombinowanych rozwiązaniach przydzielony do tego samego centrum dystrybucji, to w rozwiązaniu potomnym przydziel go do tego samego centrum
 - W przeciwnym wypadku przydziel go do centrum dystrybucji pochodzącego od jednego (wybranego losowo) rekombinowanego rozwiązania

Kolonie mrówek

- Analogia – mrówki poszukujące pożywienia
- Proste zasady jakimi kieruje się każda z mrówek prowadzą do bardzo sprawnego działania całej kolonii
- Związek z optymalizacją
 - Rozwiązanie – droga mrówki
 - Funkcja celu – długość trasy
 - Sąsiedztwo – ruchy mrówek
 - Ślad feromonowy – współdzielona wiedza

Algorytmy hybrydowe i hiperheurystyki

- Hybrydowe algorytmy ewolucyjne
 - Metody wykorzystujące zarówno operator rekombinacji (krzyżowania), jak i operatory sąsiedztwa (lokalną optymalizację)
 - Rozwiązanie uzyskane przy użyciu rekombinacji i mutacji jest często ogólnie dobre, ale wymaga systematycznych, drobnych poprawek by stać się lokalnym optimum
- Inne hybrydy
- Hiper-heurystyki, hiper-hiper-heurystyki, ...; NFL

Porównywanie algorytmów optymalizacji

- Czas vs. jakość
- Eksploracja (dywersyfikacja) vs. eksploatacja (intensyfikacja)
 - charakterystyka poznanych algorytmów: losowy, dokładny, metaheurystyki i ich parametry

Optymalizacja wielokryterialna

Uwzględnienie wielu kryteriów

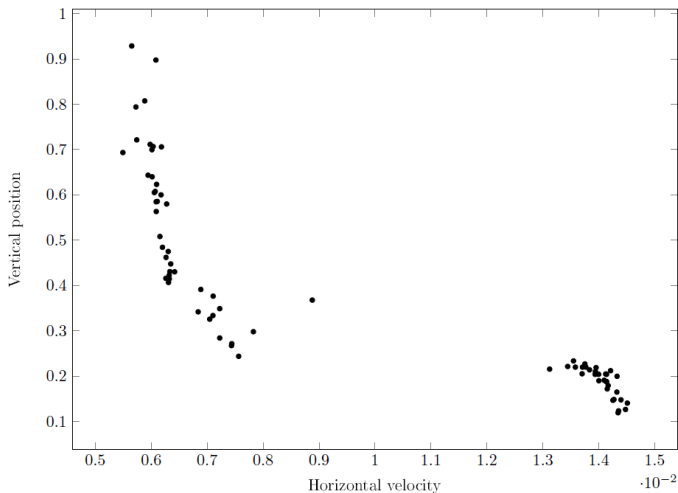
- Przykłady: TSP, plan zajęć

Optymalizacja wielokryterialna

Uwzględnienie wielu kryteriów

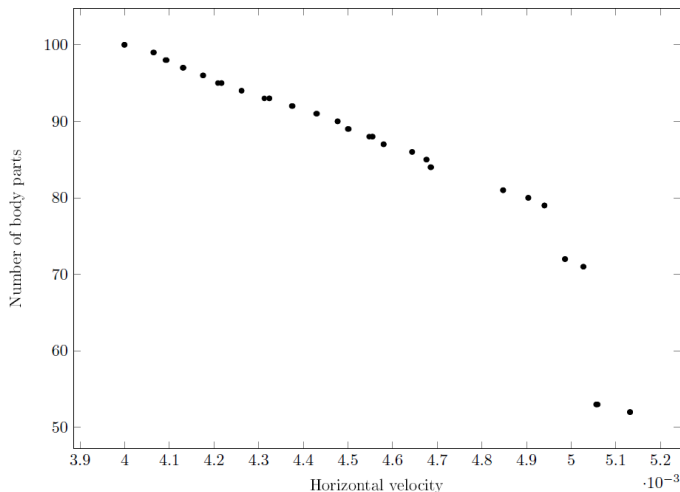
- Przykłady: TSP, plan zajęć
- Zamieniamy na jedno kryterium
 - Porządek leksykograficzny
 - Suma ważona
 - Relacja dominacji i użycie wybranej liczby charakteryzującej
- Uwzględniamy wiele kryteriów – relacja dominacji
 - Pareto Local Search – zaczynamy od wielu rozwiązań losowych, (*) wyłaniamy niezdominowane, dla każdego niezdominowanego tworzymy sąsiedztwo oceniając wielokryterialnie sąsiadów, wracamy do (*)
 - Dedykowane mechanizmy i elementy algorytmów – np. selekcja w AE (NSGA, SPEA i inne)

Optymalizacja wielokryterialna, populacja, przykład 2D (a)



(a) maximization of horizontal velocity and vertical position

Optymalizacja wielokryterialna, populacja, przykład 2D (b)



(b) maximization of horizontal velocity and number of creature's body parts

Optymalizacja wielokryterialna, populacja, przykład 3D

