Algorytmy i struktury danych

Algorytmy przybliżone i dokładne

Problem komiwojażera

Ewa Figielska 2022

Problem komiwojażera

- Problem komiwojażera (travelling salesman problem – TSP)

Komiwojażer ma odwiedzić n miast.

Przez każde z miast ma przejechać dokładnie jeden raz.

Dane są odległości między każdą parą miast.

Celem jest znalezienie najkrótszej drogi łączącej wszystkie miasta.

Problem optymalizacyjny

- Problem komiwojażera jest **problemem optymalizacyjnym.**
- Dla problemu optymalizacyjnego poszukujemy **rozwiązania**, które spełnia **ograniczenia** problemu i minimalizuje (bądź maksymalizuje) funkcję celu.

- W problemie komiwojażera:
  - Rozwiązanie ma postać ciągu liczb reprezentujących miasta; kolejność liczb w ciągu określa kolejność odwiedzania miast.
  - Rozwiązanie nie może zawierać powtarzających się wartości; czasem nałożone jest wymaganie na wartość elementu początkowego i/lub końcowego.
  - Funkcją celu jest długość trasy, która jest minimalizowana.

Algorytm dokładny

- **Algorytm dokładny** znajduje najlepsze możliwe rozwiązanie problemu optymalizacyjnego.
- Najlepsze możliwe rozwiązanie problemu nazywane jest **rozwiązaniem optymalnym.**

- Algorytm dokładny (optymalny) dla problemu komiwojażera:
  1. Wygeneruj wszystkie możliwe kolejności odwiedzania n miast;
  2. Dla każdej kolejności wyznacz długość trasy przejazdu;
  3. Wybierz najkrótszą trasę.

- Złożoność algorytmu dokładnego $\boldsymbol{O(n!)}$.

- Taki algorytm można zostosować tylko dla problemów o niewielkich rozmiarach.

- W praktyce spotykane są zwykle złożone problemy o dużych rozmiarach, dla których nie istnieją algorytmy dokładne o złożoności wielomianowej. Aby rozwiązać takie problemy, projektowane są **algorytmy przybliżone**.

Algorytm przybliżony - heurystyka

Algorytmy przybliżone nazywane są **heurystykami.**

- **Heurystyka** nie gwarantuje znalezienia rozwiązania optymalnego.
- Heurystyka znajduje **rozwiązanie przybliżone** (gorsze od optymalnego) w "rozsądnym" czasie (heurystyka powinna być efektywna - mieć złożoność wielomianową).
- Heurystyka często daje rozwiąznia dość bliskie optymalnym, ale mogą się również pojawiać wyniki bardzo odległe od optimum.

Algorytm zachłanny

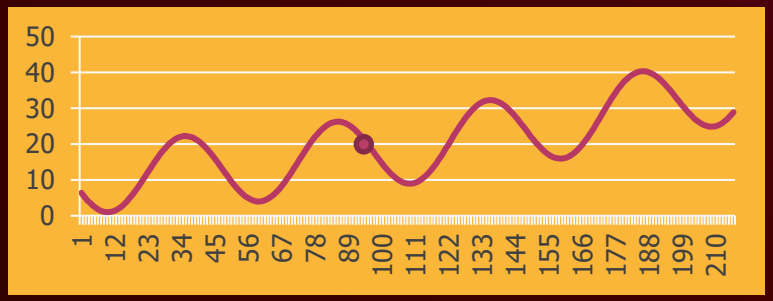
- Przykładem heurystyki jest **algorytm zachłanny** (greedy algorithm).
- Ogólnie, algorytm zachłanny polega na sukcesywnej budowie rozwiązania poprzez podejmowanie w każdym kroku najlepszej w danym momencie decyzji. Rozwiązanie konstruowane jest od "zera".

- Algorytm zachłanny dla problemu komiwojażera – algorytm najbliższego sąsiada.
  1. Ustaw jedno z miast (wskazane lub wybrane losowo) jako bieżące miasto;
  2. Dodaj do trasy przejazdu to z nieodwiedzonych miast, do którego odległość z miasta bieżącego jest najmniejsza, po czym ustaw nowo dodane miasto jako bieżące;
  3. Powtarzaj krok 2 aż do chwili, gdy wszystkie miasta zostaną odwiedzone.
- Złożoność obliczeniowa algorytmu zachłannego $\boldsymbol{O(n^2)}$.

- Powyższy algorytm nie jest uniwersalny - pasuje tylko dla problemu komiwojażera.

Poprawa rozwiązania

- Gdy znane jest jakieś rozwiązanie przybliżone problemu (otrzymane np. za pomocą algorytmu zachłannego), to można je spróbować poprawić stosując inny algorytm heurystyczny.
- Na przykład można by zastosować taki algorytm:
  1. Przeszukaj sąsiedztwo bieżącego rozwiązania.
  2. Jeżeli w sąsiedztwie znajduje się rozwiązanie lepsze od bieżącego, to zaakceptuj je jako bieżące i wróć do kroku 1.
  3. Jeżeli w sąsiedzwie nie ma rozwiązania lepszego niż bieżące, to zakończ proces przeszukiwań.



Startując z zaznaczonego punktu i przeszukując jego sąsiedztwo, algorytm może utknąć w lokalnym optimum

Sąsiad danego rozwiązania jest tworzony przez małą zmianę tego rozwiązania (niewiele się od niego różni).

- Aby móc uciec z lokalnego optimum, lepszy byłby algorytm taki:
  1. Przeszukaj sąsiedztwo bieżącego rozwiązania.
  2. Jeżeli w sąsiedztwie znalezione zostało rozwiązanie lepsze od bieżącego, to zaakceptuj je jako bieżące i idz do kroku 4.
  3. Jeżeli w sąsiedzwie nie zostało znalezione rozwiązanie lepsze niż bieżące, to zaakceptuj gorsze rozwiązanie pod pewnymi warunkami.
  4. Jeżeli kryterium stopu jest spełnione to zakończ, w przeciwnym przypadku idź do kroku 1.

Metaheurystyka

- Algorytmem, który przez wykonanie ruchu do rozwiązania gorszego niż bieżące, pozwala na ucieczkę z lokalnego optimum, jest na przykład **algorytm symulowanego wyżarzania** (SW, ang. *simulated annealing*, SA).
- Algorym SW jest przedstawicielem **algorytmów metaheurystycznych**.

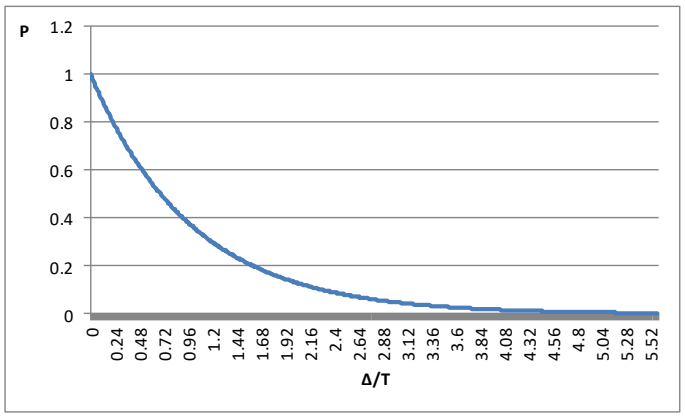
- Metaheurystyki są to uniwersalne heurystyki, bazujące często na analogiach do procesów ze świata rzeczywistego (fizyki, chemii, biologii), które można interpretować w kategoriach optymalizacji.
  - Np wyżarzanie:
    - wzrost temperatury gorącej kąpieli do takiej wartości, w której ciało stałe topnieje
    - powolne zmniejszanie temperatury do chwili, w której cząsteczki ułożą się wzajemnie i osiągną (ang. ground state) temperaturę zerową
    - przeciwieństwo hartowania

Algorytm SW – schemat

- Algorytm symulowanego wyżarzania jest procedurą iteracyjną.
- Startuje on z pewnego rozwiązania początkowego i w każdej kolejnej iteracji tworzy nowe rozwiązanie przez wprowadzenie małej zmiany w bieżącym rozwiązaniu (czyli tworzy sąsiada bieżącego rozwiązania).
- Jeżeli nowe rozwiązanie jest lepsze od bieżącego, to jest ono akceptowane i przyjmowane jako bieżące rozwiązanie w następnej iteracji.
- Jeżeli nowe rowiązanie jest gorsze niż poprzednie , to jest ono akceptowane z prawdopodobieństwem $P = e^{-\Delta/T}$, które jest malejącą funkcją Δ/T, gdzie
  - T jest parametrem nazywanym temperaturą,
  - Δ jest różnicą miedzy funkcjami oceny (np. długością trasy dla TSP) dla nowego i poprzedniego rozwiązania.

Algorytm SW - idea

- Algorytm startuje z wysoką wartością temperatury.
- Czyli na początku procesu optymalizacji prawdopodobieństwo akceptacji gorszegp rozwiązania jest duże -> algorytm przegląda duże obszary przestrzeni rozwiązań i identyfikuje te obszary, w których występują dobrej jakości rozwiązania.
- Wraz z postępem procesu optymalizacji, temperatura jest stopniowo obniżana -> maleje prawdopodobieństwo akceptacji gorszego rozwiązania, a algorytm skupia się na obiecujących obszarach.
- Jeżeli nowe rozwiązanie jest dużo gorsze od niż bieżące, to wartość Δ jest duża -> takie rozwiązanie jest akkceptowane z małym prawdopodobieństwem.
- Kiedy nowe rozwiązanie jest w niewielkim stopniu gorsze od bieżącego, to wartość Δ jest mała -> nowe rowiązanie jest akceptowane z dużym prawdopodobieństwem.



$$P = e^{-\Delta/T}$$

Prawdopodobieństwo akceptacji gorszego rozwiązania

Ocena jakości rozwiązań algorytmów heurystycznych

- **Względne odchylenie od optimum.**

$$\delta = \frac{S - S_{opt}}{S_{opt}} 100\%$$

gdzie

S – wartość funkcji celu dla rozwiązania uzyskanego za pomocą badanego algorytmu heurystycznego

S_{opt} – wartość funkcji celu dla rozwiązania optymalnego

- **Względna poprawa rozwiązania (porównanie z wynikami otrzymanymi za pomocą innej heurystyki).**

$$\sigma = \frac{S_h - S}{S_h} 100\%$$

gdzie

S - wartość funkcji celu dla rozwiązania otrzymanego za pomocą badanego algorytmu heurystycznego

S_h – wartość funkcji celu dla rozwiązania otrzymanego za pomocą innego (gorszego) algorytmu heurystycznego

Uwaga: Wzory są ważne dla problemu **minimalizacji**

Przykład

Tabela odległości między miastami:
(odległość od miasta i do miasta j)

i \ j	1	2	3	4	5
1	0	36	96	36	15
2	40	0	86	5	26
3	7	69	0	83	24
4	18	68	8	0	73
5	30	79	69	30	0

- Liczba miast, $n = 5$
- Zakładamy, że komiwojażer startuje z miasta 1 i nie wraca do miasta macierzystego.

- Algorym dokładny (AD) - rozważane są wszystkie możliwe kolejności miast, liczba możliwości = 4!
  - Rozwiązanie (optymalne) - trasa: 1,2,4,3,5
  - Długość trasy d_{AD}= 73

Kolejność	Długość trasy
1,2,3,4,5	36+86+83+73 = 278
1,3,2,4,5	96+69+5+73 = 243
…	…
1,2,4,3,5	36+5+8+24= 73
…	…

- Algorytm zachłanny (AZ) - algorytm najbliższego sąsiada
  - Rozwiązanie (przybliżone) trasa: 1,5,4,3,2
  - Długość trasy d_{AZ} = 122

Dodane miasto	Aktualna długość trasy
1	0
5	0+15 = 15
4	15+30 = 45
3	45+8 = 53
2	53 + 69 = 122
Długość trasy:	122

Przykład. Ocena jakości rozwiązań algorytmów heurystycznych

- **Względne odchylenie od optimum**

  (Porównanie rozwiązania otrzymanego przez algorytm zachłanny z rozwiązaniem optymalnym)

  Dla algorytmu zachłannego**:**

$$\boldsymbol{\delta_{AZ} = \frac{d_{AZ} - d_{AD}}{d_{AD}} 100\% = \frac{122-73}{73} 100\% = 67.12\%}$$

  Względne odchylenie od optimum rozwiązania otrzymanego za pomocą algorytmu zachłannego wynosi 67.12%.

Przykład. Ocena jakości rozwiązań algorytmów heurystycznych

- Załóżmy, że podjęta została próba poprawienia rozwiązania otrzymanego algorytmem zachłannym przez zastosowanie algorytmu SW , w wyniku której otrzymano trasę przejazdu komiwojażera: 1,5,2,4,3 o długości $d_{ASW} = 107.$

- **Względna poprawa rozwiązania**

  (Porównanie rozwiązania otrzymanego algorytmem symulowanego wyżarzania z rozwiązaniem wyznaczonym przez algorytm zachłanny)

  Dla algorytmu SW:

$$\boldsymbol{\sigma_{ASW} = \frac{d_{AZ} - d_{ASW}}{d_{AZ}} 100\% = \frac{122 - 107}{122} 100\% = 12.30\%}$$

  Względna poprawa rozwiązania (wyznaczonego przez algorytm zachłanny) uzyskana przez zastosowanie algorytmu symulowanego wyżarzania wynosi 12.3%.