Wrocław, 6 marca 2022

Algorytmy Metaheurystyczne

Laboratorium 1

Algorytmy heurystyczne

prowadzący: *dr inż. A. Gnatowski, dr inż. R. Idzikowski, dr inż. J. Rudy*

1 Heurystyki

Heurystyki (heurezy) należą do grupy algorytmów, które nie gwarantują uzyskania rozwiązania optymalnego (w przeciwieństwie do algorytmów dokładnych), ani nie gwarantują jak duża będzie odchyłka względem optimum (w przeciwieństwie do algorytmów przybliżonych). W niektórych przypadkach nie gwarantują nawet uzyskania rozwiązania dopuszczalnego. Z drugiej strony heurystyki są zwykle prostsze w użyciu i działają w krótszym czasie. Z tego względu niejednokrotnie heurystyki wykorzystywane są jako składniki większych i bardziej wyrafinowanych algorytmów. Poniżej znajduje się opis kilku heurystyk dla problemu komiwojażera.

2 Metoda k-random

Metoda ta polega na wygenerowaniu losowego dopuszczalnego rozwiązania. Wadami takiego podejścia jest niedeterministyczność tej metody oraz ryzyko otrzymania bardzo słabego (a nawet najgorszego możliwego) rozwiązania. Aby zredukować to ryzyko generuje się k losowych rozwiązań i jako wynik zwraca się najlepszego napotkanego kandydata (w szczególności k może być równe 1). Jest równoznaczne losowemu próbkowaniu przestrzeni rozwiązań, a im większa próba tym większa szansa na uzyskanie dobrego rozwiązania. Teoretycznie dla $k \to \infty$ prawdopodobieństwo znalezienia optimum jest równe 1 ("prawie na pewno"), jednak w praktyce zbyt duże k przestaje być opłacalne. Parametr ten pozwala sterować balansem pomiędzy czasem działania a (statystyczną) jakością wyników.

3 Metoda najbliższego sąsiada

Zaczniemy od przypomnienia co to jest metoda zachłanna. Jest to klasyczna heurystyka, bazująca na założeniu, że decyzje optymalne lokalnie są również optymalne globalnie. Metoda polega na konstruowaniu rozwiązania krok po kroku. Najczęściej rozpoczyna się od rozwiązania pustego i poprzez ciąg decyzji buduje się kolejne elementy rozwiązania częściowego aż do uzyskania rozwiązania pełnego. Istotą jest to że każda decyzja (lokalna) podejmowana jest w sposób optymalny. Szybkość metody wynika z faktu, że decyzje podejmowane w każdym kroku są proste (tj. są to zwykle problemy obliczeniowe o bardzo niewielkiej przestrzeni rozwiązań, możliwej do efektywnego rozwiązania metodą przeglądu zupełnego). Dla niektórych problemów algorytmy zachłanne są optymalne, przykładem jest problem wydawania reszty (przy standardowych nominałach monet). Dla wielu problemów (np. szachy) metody zachłanne nie są jednak optymalne.

Dla problemu komiwojażera typową implementacją metody zachłannej jest algorytm najbliższego sąsiada. Algorytm zaczyna się od wybrania (w dowolny sposób) miasta początkowego. Następnie w pętli wybierane są kolejne miasta, przy czym zawsze wybierane jest miasto, które jest najbliższym sąsiadem miasta odwiedzonego jako ostatnie. Algorytm kończy się po odwiedzeniu wszystkich miast.

4 Rozszerzona metoda najbliższego sąsiada

Zauważmy, że w opisanym powyżej algorytmie najbliższego sąsiada po odwiedzeniu danego miasta wszystkie krawędzie do niego wchodzące nie będą mogły być już użyte w kolejnych krokach. Skutkiem ubocznym jest fakt, że decyzje algorytmu (a tym samym jakość wyniku) mogą zależeć od wyboru miasta początkowego, gdyż wybór ten "usuwa" część krawędzi, które mogły w kolejnych krokach okazać się (lokalnie) najkrótszymi. Rozszerzenie algorytmu najbliższego sąsiada ma na celu jego uniezależnienie od wyboru wierzchołka początkowego.

5 Algorytm 2-OPT

2-OPT jest heurystyką dedykowaną dla problemu komiwojażera (choć po niewielkich zmianach znajduje też zastosowanie dla zbliżonych problemów). Jest ona bardziej złożona i należy do grupy algorytmów iteracyjnych, ściślej algorytmów popraw. Innymi słowy w każdej iteracji algorytm stara się poprawić istniejące rozwiązanie i kończy się gdy nie udało się tego zrobić. Ponadto 2-OPT jest algorytmem lokalnego poszukiwania co oznacza że wykorzystuje pojęcie otoczeniem.

Intuicyjnie *otoczeniem* (sąsiedztwem) $N(\pi)$ rozwiązania π nazywamy pewien zbiór rozwiązań „bliskich" π, które nazywamy sąsiadami π. Sama bliskość jest pojęciem względnym – otoczenie dla tego samego π zwykle można zdefiniować na różne sposoby (różne typy otoczeń). Rozmiar otoczenia to liczba sąsiadów. Uzupełnieniem jest pojęcie *ruchu*, które można opisać jako funkcję, której argument jest π, a wartością powstały sąsiad. Otoczenie π możemy więc alternatywnie zdefiniować jako zbiór rozwiązań możliwych do uzyskania za pomocą danego ruchu. Zwykle oczekuje się by otoczenie spełniało szereg własności:

1. $N(\pi)$ nie powinno zawierać π.
2. $N(\pi)$ nie powinno być zbyt duże ani puste.
3. Każdy element $N(\pi)$ powinien niewiele różnić się od π (ruchy powinny być elementarne).
4. Ruchy powinny być możliwie proste (niższy koszt konstrukcji i oceny sąsiada).
5. Dla dowolnych π_1, π_2 powinien istnieć ciąg ruchów, którym można przejść od π_1 do π_2 (każde rozwiązanie powinno być osiągalne z każdego)[1].

Przykładowym otoczeniem dla problemu komiwojażera (lub innych problemów reprezentujących rozwiązania za pomocą permutacji) jest otoczenie *invert*. Ruch dany jest funkcją $\mathrm{invert}(\pi, i, j)$, która polega na odwróceniu kolejności wierzchołków od i-tego do j-tego. Dla przykładowego rozwiązania:

$$\pi = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) \tag{1}$$

wykonanie ruchu $\mathrm{invert}(\pi, 4, 7)$ da rozwiązanie pośrednie postaci:

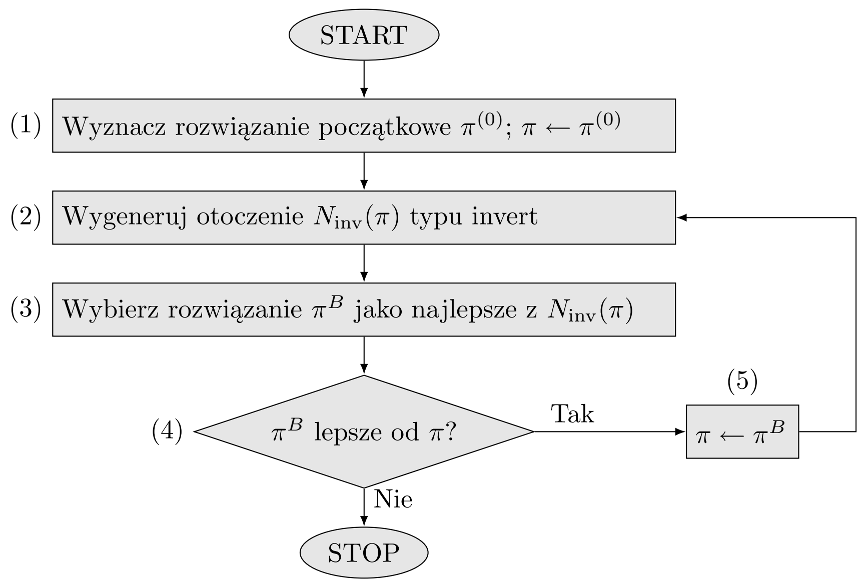
$$\pi = (1, 2, 3, 7, 6, 5, 4, 8, 9, 10). \tag{2}$$

Algorytm 2-OPT opiera się na obserwacji, że ruchy typu invert prowadzą często do poprawy rozwiązania przez poprawienie („rozplątanie") charakterystycznych nieoptymalnych pętli, łatwych do zwizualizowania w euklidesowym wariancie problemu. Algorytm 2-OPT składa się z następujących kroków:

- **Krok 1** Wybierz rozwiązanie początkowe (w dowolny sposób), które staje się rozwiązaniem aktualnym.
- **Krok 2** Wyznacz wartość funkcji celu wszystkich sąsiadów rozwiązania aktualnego (dla otoczenia invert).
- **Krok 3** Jako kandydata do poprawy wybierz najlepszego z ocenionych sąsiadów.
- **Krok 4** Jeśli kandydat nie jest lepszy od aktualnego rozwiązania, to zakończ algorytm.
- **Krok 5** Zastąp aktualne rozwiązanie kandydatem i przejdź do kroku 2.

Schemat algorytmu 2-OPT znajduje się na Rysunku 1.

[1] Czasami wymaga się mniej restrykcyjnej wersji: by z każdego rozwiązania dopuszczalnego było osiągalne co najmniej jedno optimum (własność połączenia).



Rysunek 1: Uproszczony schemat algorytmu 2-OPT