# ITK - LABORATORIUM

*Temat ćwiczenia:*

***TSP - Problem komiwojażera***

| Imię i nazwisko | Rok ak. | Gr. | Sem. | Komputer | Data | Podpis prowadzącego |
|---|---|---|---|---|---|---|
| ..... | 200__/__ | ZC6 | IX | LAB..... | ..... | |

## Zadanie

- ⇨ Zapoznać się z problemem komiwojażera
- ⇨ Zapoznać się z programem VISUAL GENETIC – TSP
- ⇨ Dla trasy human100.mst (w katalogu tours) znaleźć optymalne parametry dla reprezentacji ścieżkowej (jak najlepsze rozwiązanie w jak najkrótszym czasie).
- ⇨ Porównać rozwiązania z innymi sekcjami.
- ⇨ Praca własna z programem.

## Miejsce na notatki:

## Sprawozdanie ma zawierać:

1. Cel ćwiczenia.
2. Krótki opis problemu
3. Porównanie wyników uzyskanych dla różnych parametrów
4. Wnioski z ćwiczenia.

# MATERIAŁY POMOCNICZE

## 1. Problem komiwojażera - przykład rozwiązania za pomocą AG

*http://panda.bg.univ.gda.pl/~sielim/genetic/gen_komi.htm*

### 1.1 Problem komiwojażera - o co chodzi

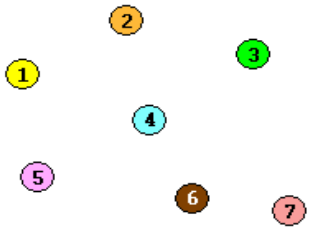
W skrócie: mamy płaską mapę a na mapie wiele miast. Chcemy odwiedzić wszystkie te miasta, a jednocześnie ponieść jak najmniejsze koszty podróży, czyli wybrać najkrótszą drogę. Możemy dowolnie wybierać kolejność odwiedzania miast, między którymi poruszamy się po liniach prostych (czyli możliwie najkrócej), ale na końcu chcielibyśmy wrócić do punktu wyjścia. Możemy więc zacząć od dowolnego miasta. Dla kilku miast problem jest prosty, natomiast bardzo się komplikuje przy większej ich ilości. Ściśle rzecz biorąc problem ten jest NP-zupełny, co w ogólności oznacza, że komplikacja problemu szybko wzrasta.

Ogólnie uważa się, że dla odpowiednio dużej ilości miast znalezienie optymalnej drogi jest praktycznie niemożliwe, nawet dla bardzo szybkich komputerów. Dodatkowo dodanie do mapy jednego miasta komplikuje problem wielokrotnie, proporcjonalnie do liczby wszystkich miast. Liczba kombinacji zadania komiwojażera wyraża się wzorem

$$\frac{(n-1)!}{2}$$ , gdzie **n** – liczba miast

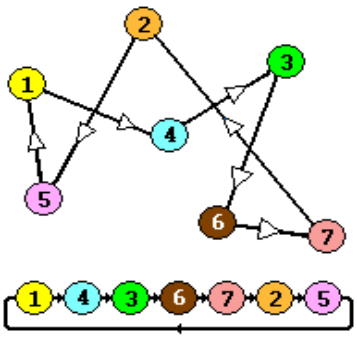
Np. zmiana z 50 do 51 miast skutkuje 50-krotnym wzrostem czasu obliczeń (zamiast 1 godziny – 2 dni)

Nie liczymy na to, że algorytm genetyczny będzie nam znajdował rozwiązanie optymalne - chcemy za pomocą algorytmu genetycznego znaleźć rozwiązanie suboptymalne, czyli możliwie jak najlepsze. Na początek załóżmy, że mamy 7 miast. Przykładowa mapa może wyglądać tak jak na rysunku poniżej.




### 1.2 Fenotyp

Fenotypem jest mapa z zaznaczonymi drogami, które dają przykładową trasę komiwojażera. Możemy ją narysować w postaci grafu z jednym, dużym cyklem spinającym wszystkie miasta. Cykl ten może być dowolny (w szczególności tak pogmatwany, że w ogóle nie będzie wyglądał jak cykl, ale jak poplątana sieć). Fenotyp, czyli pewne rozwiązanie naszego problemu jest dozwolony, jeśli spina wszystkie miasta jednym cyklem (ale każde miasto tylko raz). Może wyglądać to np. jak na rysunku poniżej. Pod spodem pokazany jest cykl, w jaki układają się miasta.



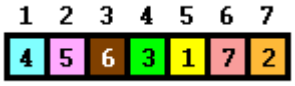
## 1.3 Funkcja oceny

Funkcją oceny jest po prostu droga, czyli długość całego cyklu w grafie. Im mniejsza, tym lepiej.

## 1.4 Genotyp, czyli budowa chromosomu

### *1.4.1 Sposób pierwszy - oparty o klasyczne rozmieszczenie genów*

1. Numerujemy wszystkie miasta, czyli wierzchołki naszego grafu.
2. Tworzymy chromosom o długości takiej, jaka jest liczba miast (tu: 7) - każde miasto ma swój gen.
3. W genie danego miasta zapisujemy numer innego miasta, do którego chcemy przejść z danego. Czyli jeśli w chromosomie gen numer X ma wartość Y oznacza to, że z miasta numer X idziemy do miasta Y. W genie Y mamy wartość Z i stąd wiemy, że z miasta numer Y mamy iść do miasta numer Z. I tak dla wszystkich miast. W ten sposób kodujemy całą mapę z przykładową drogą komiwojażera. Inaczej mówiąc w chromosomie mamy zapisaną informację o następnikach wszystkich miast.Chromosom opisujący podaną w przykładzie trasę będzie wyglądał tak:



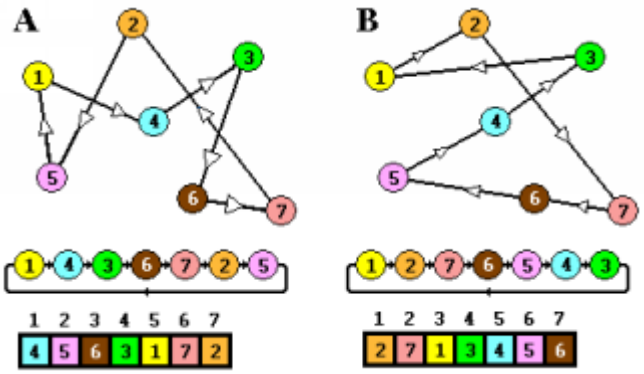

### *1.4.2 Sposób drugi - kodowanie permutacyje*

Jest to kodowanie bardziej intuicyjne dla problemu komiwojażera, który jest typowym problemem kombinatorycznym. Postępujemy następująco:

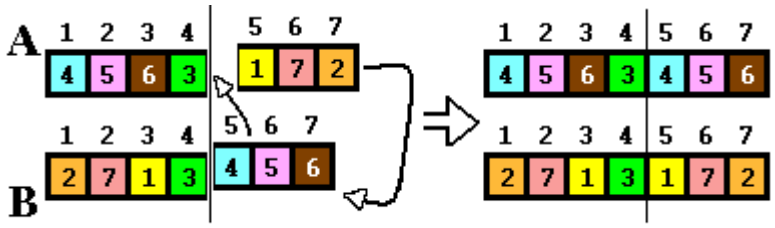
1. Numerujemy wszystkie miasta.
2. Tworzymy chromosom o długości takiej, jaka jest liczba miast (czyli 7).
3. W kolejnych genach zapisujemy kolejne miasta na drodze. Czyli, jeśli w genie numer 1 siedzi miasto X a w genie numer 2 miasto Y to oznacza to, że idziemy z miasta X do Y. W ten sposób geny w chromosomie układają się dokładnie tak, jak miasta w cyklu. W tym wypadku nie ma stałego przypisania miasta do genu, więc każda przykładowa trasa może być zapisana na wiele sposobów. Dla podanych siedmiu miast połączonych drogą dwa przykładowe chromosomy odwzorowujące ten sam cykl mogą wyglądać tak:


## 1.5 Operatory genetyczne

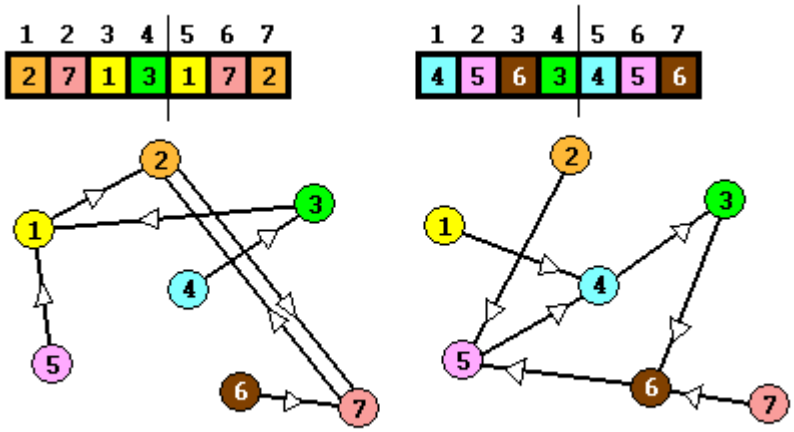
Problem komiwojażera wraz z zaproponowanymi dwoma sposobami kodowania to typowy przykład problemu, do którego trzeba indywidualnie zaprojektować operację krzyżowania. Istnieją bowiem poważne ograniczenia przy nadawaniu wartości genów. Przyjrzyjmy się pierwszemu sposobowi kodowania. Weźmy dwa przykładowe chromosomy i odpowiadające im cykle na dwóch mapach:



Teraz dokonamy krzyżowania tych dwóch chromosomów postępując zgodnie z regułami klasycznymi: wybieramy jeden punkt krzyżowania i wymieniamy kawałki chromosomów. Powstają dwa nowe:




A tak wyglądają zakodowane w tych chromosomach "mapy":




Jak widać, rezultat nie jest zbyt ciekawy. Po takim krzyżowaniu uzyskaliśmy dwa niedozwolone rozwiązania - drogi na mapach nie tworzą cykli tylko chaotyczne powiązania między miastami. Możemy zaradzić temu problemowi dokonując korekt w chromosomach. Wyszukujemy wszystkie miasta, których w chromosomie brakuje i te, które występują wielokrotnie. Losowo zamieniamy powtarzające się na brakujące. Jest z tym trochę roboty i w dodatku pozostaje inny problem: po takim losowym łączeniu może się okazać, że miasta dalej nie tworzą jednego cyklu, ale np. są spięte w kilku rozłącznych cyklach. Musimy więc jeszcze porozcinać cykle i połączyć je ze sobą.

## 2. Co zrobić żeby się komiwojażer nie przepracował...

*http://www.republika.pl/k0pper/geny.htm#komiwojazer*

Ten tekst będzie opowiadał o starym jak świat problemie NP-trudnym znanym pod nazwą: 'problem komiwojażera'. Na początku może kilka słów na czym ten problem polega: Komiwojażer musi odwiedzić wszystkie miasta z zadanego regionu i wrócić do miasta początkowego (jest to problem szukania cyklu). Wszystkie miasta są ze sobą połączone (mamy do czynienia z grafem pełnym, czyli kliką). Mając do dyspozycji macierz odległości pomiędzy poszczególnymi miastami należy znaleźć cykl o najmniejszym koszcie. I jeszcze jedno: każde miasto nie może być odwiedzone więcej niż jeden raz. A więc na wejściu mamy informację o odległościach pomiędzy miastami, a na wyjściu należy wygenerować najlepszą kolejność odwiedzanych miast.

Czasami dane o miastach zapisane są w postaci listy miast wraz z ich współrzędnymi. W tym wypadku do stworzenia macierzy odległości należy wykorzystać wzór na odległość Euklidesową pomiędzy dwoma punktami. Na płaszczyźnie jest to pierwiastek z sumy kwadratów różnic odległości dla poszczególnych współrzędnych:

$$Odleglosc(a,b)=sqrt((x_a-x_b)*(x_a-x_b)+(y_a-y_b)*(y_a-y_b))\ ,$$

Sprawdzenie wszystkich możliwych tras jest zadaniem bardzo czasochłonnym. Dlatego do znalezienia najlepszej wykorzystane zostaną algorytmy genetyczne.

Populacja początkowa będzie składała się z pewnej liczby tras wygenerowanych zupełnie losowo. Należy w tym miejscu przez chwilę zastanowić się nad reprezentacją pojedynczego rozwiązania. Ze względu na fakt, że rozwiązaniem problemu komiwojażera jest lista kolejno odwiedzanych miast, natychmiastowym pomysłem na reprezentację rozwiązania jest właśnie lista kolejnych miast. Jest to reprezentacja bardzo prosta i bardzo szybka jeśli chodzi o wyliczenie funkcji oceny dla każdego osobnika. Ma ona jednak bardzo dużą wadę. Mianowicie skrzyżowanie dwóch tras może dać osobnika nieprawidłowego.

Np. skrzyżowanie trasy **1-2-3-4-5** z trasą **4-5-1-3-2** w punkcie między drugim, a trzecim miastem da następujących potomków: **1-2-1-3-2** oraz **4-5-3-4-5**. Żadne z dzieci nie jest poprawne (zawierają kilkukrotne wystąpienie tych samych miast, a jednocześnie pewne miasta w ogóle nie występują). Dlatego w przypadku takiej reprezentacji należy zadbać albo o naprawę pokrzyżowanych osobników, albo o zmianę standardowego operatora krzyżowania jakimś bardziej wymyślnym.

Innym sposobem reprezentacji trasy jest lista pokazująca kolejność pobierania miast do tworzonej trasy, np: punktem odniesienia dla tej reprezentacji jest lista kolejnych miast: **1-2-3-4-5**. Pojedynczy osobnik np. **4-4-1-2-1** pokazuje, w jakiej kolejności wybierane są kolejno odwiedzane miasta. Na początku jest czwórka więc pierwszym odwiedzanym miastem będzie miasto umieszczone na czwartej pozycji w liście odniesienia, czyli czwórka. Czwórkę tą usuwa się z listy odniesienia (pozostają miasta 1-2-3-5), natomiast lista odwiedzanych miast wygląda następująco:

4 Kolejnym elementem osobnika jest ponownie czwórka. W tej chwili na czwartym miejscu listy odniesienia jest piątka, więc kolejnym odwiedzanym miastem będzie miasto nr 5, a lista odniesienia będzie wyglądała następująco: 1-2-3.

W kolejnych krokach będzie to wyglądało następująco:

| Analizowany element osobnika | Lista odniesienia | Tworzona lista odwiedzanych miast |
|---|---|---|
| 1 | **1**-2-3 | 4-5-1 |
| 2 | 2-**3** | 4-5-1-3 |
| 1 | **2** | 4-5-1-3-2 |

Reprezentacja ta wprowadza spore zamieszanie przy przechodzeniu na reprezentację wykorzystywaną przy funkcji oceny, jednak kłopoty te rekompensuje przy krzyżowaniu i mutacji. Cechą charakterystyczną tej reprezentacji jest fakt, że na i-tej pozycji jest liczba z przedziału od 1 do n-i+1 (gdzie n to liczba wszystkich miast). Ze względu na to wymiana materiału genetycznego między dwoma osobnikami za pomocą standardowego krzyżowania x-punktowego zawsze da dopuszczalne potomstwo.

Dla problemu komiwojażera (i innych jemu podobnych) wymyślono kilka rodzajów krzyżowań, które zawsze dają rozwiązania dopuszczalne, jak np:

- **Krzyżowanie z częściowym odwzorowaniem (PMX)**

Wybiera się losowo pewną podtrasę w obu rodzicach i przekazuje się ją do potomka (podtrasa pierwszego rodzica trafia do drugiego potomka i na odwrót):

Rodzic pierwszy: **1-2-3-4-5**, rodzic drugi: **2-4-3-5-1**, pozostawiana podtrasa: miasta od 2 do 4:
Pierwsze dziecko: **x-4-3-5-x**, drugie dziecko **x-2-3-4-x**.

Teraz uzupełnia się te trasy tak żeby nie powstał konflikt (dwa takie same miasta w trasie):
Pierwsze dziecko: **1-4-3-5-x**, drugie dziecko **x-2-3-4-1** (do dzieci nie można dodać miasta 5 (do pierwszego) ani 2 (do drugiego) ponieważ pojawiłyby się dwa takie same miasta w trasie).

Następnie tworzone są odwzorowania (na podstawie wymienianych podtras): 2<->4, 3<->3, 4<->5 i za ich pomocą uzupełniane są dzieci. W pierwszym brakuje piątki, więc na to miejsce wstawiamy miasto nr 2 (5<->4 i 4<->2), natomiast w drugim brakuje dwójki, więc wstawimy 5 (2<->4 i 4<->5).

Wygenerowane w ten sposób dzieci będą wyglądały następująco:
Pierwsze: **1-4-3-5-2**, drugie **5-2-3-4-1**.

- **Krzyżowanie z porządkowaniem (OX)**

Potomków tworzy się na podstawie podtras pobranych z rodziców (podtrasa pierwszego dziecka pobierana jest z drugiego rodzica natomiast podtrasa drugiego dziecka z pierwszego). Następnie uzupełnia się trasy miastami pobranymi z drugiego rodzica z zachowaniem porządku z pominięciem miast już wykorzystanych:

Rodzic pierwszy: **3-2-1-4-5**, rodzic drugi: **2-4-3-5-**1, pozostawiana podtrasa: miasta od 2 do 4:
Pierwsze dziecko: **x-4-3-5-x**, drugie dziecko **x-2-1-4-x**
Uuzupełniamy pierwsze dziecko: **x-4-3-5-*2*** (ominęliśmy 5 i 3 ponieważ te miasta już występują) -> ***1*-4-3-5-2**; i drugie dziecko: **x-2-1-4-*3*** (ominęliśmy 1, 2 i 4 ponieważ te miasta już występują) -> ***5*-2-1-4-3**.

W przypadku drugiej reprezentacji ('mniej naturalnej') taka zabawa jest niepotrzebna ponieważ standardowe operatory krzyżowania x-punktowego zawsze dadzą prawidłowych potomków. Jeśli chodzi o mutację to sytuacja jest bardzo podobna: w zależności od rodzaju reprezentacji można zastosować standardowe operatory mutacji, bądź jakieś bardziej wyrafinowane.

W przypadku reprezentacji 'naturalnej' najprostszym rodzajem mutacji jest wymiana ze sobą dwóch miast w rozwiązaniu. Np:
**1-2-3-4-5** -> wymiana miast 2 i 4 -> **1-4-3-2-5**

Można również zmieniać kolejność przechodzenia miast. Np:
**1-2-3-4-5** -> Zmiana kolejności w obrębie miast 1 i 4 -> **4-3-2-1-**5

Można również przesunąć jakieś miasto (lub grupę miast) w ramach rozwiązania. Np:
**1-2-3-4-5** -> Wstawienie miasta 1 pomiędzy 4 i 5 -> **2-3-4-1-5**

Dla drugiej reprezentacji taka zabawa jest niepotrzebna, ponieważ w tym przypadku mutacja sprowadza się do zamiany wartości z pozycji i losową wartością z przedziału od 1 do n-i+1 (n - liczba wszystkich miast).