

# Algorytm ewolucyjny dla problemu n-queens

dr inż. Joanna Kołodziejczyk

## 1 Zadanie

Celem zadania jest wykonanie implementacji algorytmu ewolucyjnego do znalezienia rozwiązania dla n-hetmanów.

## 2 Algorytm ewolucyjny pseudokod

```
 $P \leftarrow P_0$   
 $evaluate(P)$   
 $gen \leftarrow 0$   
 $best \leftarrow min(evaluate(P))$   
while  $\{gen < gen_{max} \wedge evaluate(P(best)) > ff_{max}\}$  do  
   $P_n \leftarrow selection(P)$   
   $crossover(P_n)$   
   $mutation(P_n)$   
   $evaluate(P_n)$   
   $best \leftarrow min(evaluate(P_n))$   
   $P \leftarrow P_n$   
   $gen \leftarrow gen + 1$   
end while  
return  $P(best), evaluate(P(best))$ 
```

## 3 Opis algorytmu

- Każdy osobnik reprezentuje tablicę z n-hetmanami. Osobnik jest wektorem o długości  $n$ . Interpretacja osobnika jest następująca: indeks w wektorze, to współrzędna  $x$  a wartość w wektorze to współrzędna  $y$ . Wartości w wektorze nie mogą się powtarzać (dzięki temu zapewnia się separację wierszy i kolumn). np. dla  $n = 6$   $osobnik = [6, 5, 2, 1, 3, 4]$  to ustawienie hetmanów w pozycjach  $\{(1, 6); (2, 5); (3, 2); (4, 1); (5, 3); (6, 4)\}$  (zakładając, że wektor jest indeksowany od 1).
- $P$  populacja (zbiór potencjalnych ustawień hetmanów na szachownicy) - tablica o zadanej liczności  $pop$  osobników. Przykład populacji dla  $pop = 5$  i  $n = 6$  przedstawia tabela 1.

Osobnik	kod osobnika
1	[6, 5, 2, 1, 3, 4]
2	[2, 1, 4, 5, 3, 6]
3	[6, 1, 3, 5, 2, 4]
4	[5, 2, 1, 3, 6, 4]
5	[5, 2, 4, 6, 1, 3]

Tablica 1: Populacja  $P$  o rozmiarze  $pop = 5$  dla  $n = 6$

- $P_0$  to początkowa populacja, która jest tablicą o wielkości  $pop$  z  $n$ -hetmanami, których pszycje są losowe - (losowanie wykonać poprzez losowanie permutacji z  $n$ ).
- $evaluate(P)$  oblicza funkcję przystosowania dla problemu  $n$ -hetmanów, która zwraca liczbę ataków. Wykonaj dla każdego osobnika w populacji, by dowiedzieć się jak skutecznym jest rozwiązaniem.
- $best$  - indeks najlepszego osobnika w populacji  $P$ .
- $P_n$  jest nową populacją po selekcji
- $selection(P)$  selekcja turniejowa:
 

```

while  $i < pop$  do
   $i_1 \leftarrow random(pop)$ 
   $i_2 \leftarrow random(pop)$ 
  if  $i_1 \neq i_2$  then
    if  $evaluate(P(i_1)) \leq evaluate(P(i_2))$  then
       $P_n(i) \leftarrow P(i_1)$ 
    else
       $P_n(i) \leftarrow P(i_2)$ 
    end if
   $i \leftarrow i + 1$ 
end if
end while

```
- $ff_{max}$  jest wartością oczekiwaną funkcji przystosowania/użyteczności dla rozwiązania. Dla problemu  $n$ -hetmanów jest to najmniejsza liczba ataków, tj. 0.
- $gen_{max}$  - maksymalna liczba kroków algorytmu ewolucyjnego (generacja) 1000.
- funkcja  $crossover(P_n)$  zmienia osobniki wymieniając informacje zawarte w dwóch losowo wybranych osobnikach.
 

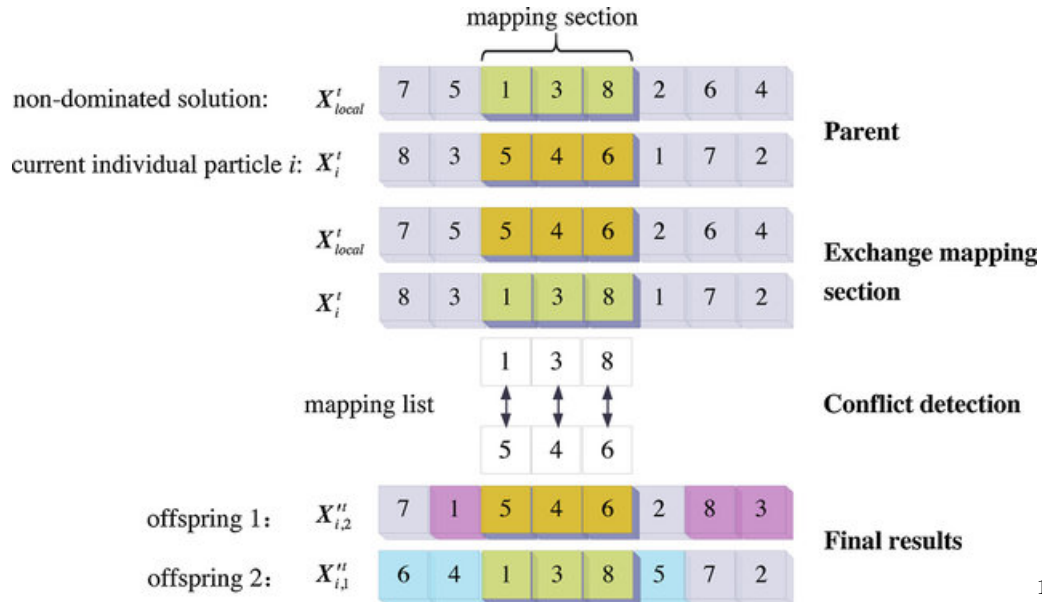
```

while  $i < pop - 2$  do
  if  $random() \leq p_c$  then
     $cross(P(i), P(i+1))$ 
   $i \leftarrow i + 2$ 
end if
end while

```

$p_c$  - prawdopodobieństwo krzyżowania — parametr kontrolujący liczbę krzyżowań.  
 $random()$  - zwraca wartości rzeczywiste od 0 do 1.

Funkcja *cross* realizuje krzyżowanie metodą PMX (partially mapped crossover) przedstawiony na schemacie. „Mapping section” jest losowane dla każdej pary, czyli może mieć inny rozmiar i położenie.



- funkcja  $mutation(P_n)$  zmienia losowo osobniki wg schematu.

```

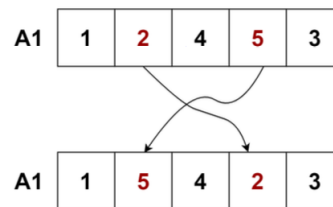
while  $i < pop$  do
  if  $random() \leq p_m$  then
    mutate( $P(i)$ )
   $i \leftarrow i + 1$ 
end if
end while

```

$p_m$  - prawdopodobieństwo mutacji — parametr kontrolujący liczbę mutacji.

$random()$  - zwraca wartości rzeczywiste od 0 do 1.

Funkcja *mutate* ma wymieniać dwóch losowo wybranych hetmanów.



<sup>1</sup>Author: Chao Guan, „Multi-objective particle swarm optimization for multi-workshop facility layout problem”

### 3.1 Wykonanie

Przed rozpoczęciem działania algorytmu trzeba ustalić niektóre parametry:

- $n$  - rozmiar szachownicy i liczba hetmanów
- $pop$  - liczba osobników w populacji: np. 10 lub 100
- $gen_{max}$  - maksymalna liczba generacji np. 1000 lub 10000
- $p_c$  - dyskryminator krzyżowania np. 0.7 lub 0.9
- $p_m$  - dyskryminator mutacji np. 0.2 lub 0.5

Rozwiązaniem ma być najlepszy osobnik - pokazujący ustawienia hetmanów na szachownicy oraz liczba bít. Należy wykonać wykres zmienności wartości funkcji przystosowania najlepszego osobnika w generacjach oraz średniej wartości funkcji przystosowania z danej populacji też w generacjach (oś X - generacje, oś Y - wartość funkcji przystosowania).

## 4 Przekazanie zadań

Kod z rozwiązaniem proszę podpiąć w Teams. Proszę w **nazwach plików źródłowych zawierać swoje nazwisko** celem łatwiejszej identyfikacji.