

# Algorytmy genetyczne. Lab 12

Marcin Fabrykowski

10 czerwca 2012

## 1 Zdefiniowanie problemu

Naszym celem było przygotowanie strategii gry w kamienie. Gra polega na zdejmowaniu naprzemiennie jednego bądź dwóch kamieni. Grę zaczyna się z 50 kamieniami na stole.

## 2 Kodowanie

Osobnika reprezentującego strategię gry początkowo kodowałem binarnie na 50 bitach. Była to relacja: numer bitu  $\rightarrow$  liczba kamieni na stole. Wartość 0 oznaczała zabranie jednego kamienia, natomiast wartość 1 dwóch kamieni.

Niezależnie od parametrów algorytmu genetycznego (szczegóły w kolejnych sekcjach), nie udawało się doprowadzić wynikowej strategii do idealnej, tj. takiej która od samego początku gry dawałaby zwycięskie ruchy. Stabilizacja zaczynała się od ok 46 kamieni na stole.

Dlatego zastosowałem kodowanie 55 bitowe, reprezentujące grę zaczynającą się od 55 kamieni. Problem niestabilności na początku gry pozostał, jednak został on przesunięty poza obszar naszego zainteresowania, co w efekcie dobrania odpowiednich parametrów, daje nam optymalną strategię gry w kamienie.

### 2.1 Inicjalizacja

Osobniki w populacji początkowej inicjalizowane są losowym ciągiem bitów. Reprezentują one losowe strategie.

## 3 Ocena dostosowania osobnika

Aby ocenić dostosowanie osobnika, przeprowadzamy stukrotną rozgrywkę między badanym osobnikiem a setką najlepszych osobników w populacji. Za każdą wygraną rozgrywkę przyznajemy 10 pkt dostosowania.

Bo przeprowadzeniu pojedynków, sprawdzamy czy nasz osobnik nie próbuje zabrać dwóch kamieni, jeżeli pozostał tylko jeden. W takim przypadku ustanawiamy karę w wysokości 500 pkt dostosowania.

## 4 Rozgrywka

Rozgrywka przeprowadzana jest do czasu, aż na stole nie pozostanie żadnej kamień. Liczba kamieni zdejmowanych przez aktualnego gracza określana jest na podstawie wartości na  $n + 1$  bicie osobnika (przesunięcie wyniku z indeksowania osobnika: bit zerowy odpowiada jednemu pozostałemu kamieniowi, bit pierwszy - dwóm). Zwracana jest prawda, jeżeli gracz pierwszy wygrał, bądź fałsz w przeciwnym wypadku.

## 5 Parametry algorytmu genetycznego

### 5.1 Założenia dot. oceny strategii

Jako ocenę otrzymanej strategii, wykorzystam fakt minimalizacji problemu gry w kamienie do zasady "przegranego pola", tj. pola z którego gracz nie ma możliwości wygrać, jeżeli przeciwnik wykonuje odpowiednie ruchy.

Takimi polami są pola  $1, 4, 7, 10, 13, \dots, 3 * n + 1$ . Miara "dobroci" strategii będzie procent takich pól do których zostanie zepchnięty przeciwnik do wszystkich możliwych. Generowanie będzie powtarzane pięciokrotnie, a wynikiem ostatecznym będzie średnia.

### 5.2 Porównanie wybranych zestawów parametrów

#### 5.2.1 Kodowanie 50 bitów

- Populacja: 100
  - Pokoleń: 50
  - Krzyżowanie: 0.9
  - Mutacja: 0.01

	OnePoint	TwoPoints	PartialMatch
Roulette	47.5% 0.242s	58.75% 0.244s	41.25% 0.25s
Rank	42.25% 0.192s	47.4% 0.198s	66.25% 0.202s

#### 5.2.2 Kodowanie 55 bitów

- Populacja: 100
  - Pokoleń: 50
  - Krzyżowanie: 0.9
  - Mutacja: 0.01

	OnePoint	TwoPoints	PartialMatch
Roulette	58.606% 0.267s	46.25% 0.27s	33.75% 0.27s
Rank	53.75% 0.226s	48.75% 0.216s	50% 0.23s

- Populacja: 200
  - Pokoleń: 200
  - Krzyżowanie: 0.9
  - Mutacja: 0.01

	OnePoint	TwoPoints	PartialMatch
Roulette	78.75% 2.11s	83.75% 2.11s	40% 2.17s
Rank	97.5% 1.9s	98.75% 1.73s	97.5% 1.78

### 5.3 Ocena wyników

Zauważamy, że dla źle dobranych parametrów algorytmu, metoda krzyżowania ani selekcji nie odgrywa większej roli. Natomiast przy lepiej dobranych parametrach, metoda rankingowa daje lepsze wyniki niż ruletka. Metody krzyżowania nie wprowadzają zbyt dużych różnic. Małą przewagę "dobroci" strategii jak i czasu wykonywania ma metoda dwupunktowa, dlatego zdecydowałem się wybrać tę właśnie metodę.