Wydział Elektroniki i Technik Informacyjnych
Politechnika Warszawska

Przeszukiwanie i optymalizacja

Dokumentacja wstępna

Kacper Bugała, Jan Kuc

Warszawa, 2022Z

1. Opis problemu

Tematem projektu jest zaimplementowanie i wykorzystanie algorytmu genetycznego oraz algorytmu ACO do rozwiązania *Sudoku*, a następnie porównanie ich działania przy pomocy zaprojektowanych eksperymentów numerycznych. W grze *Sudoku* dąży się do uzupełnienia planszy 9x9 (podzielonej dodatkowo na 9 mniejszych kwadratów 3x3) liczbami od 1 do 9 tak, aby te same nie powtarzały się w żadnym wierszu, kolumnie ani mniejszym kwadracie. Rozwiązanie problemu jest jednoznaczne, więc do oceny niewygrywającej planszy została algorytmom przypisana funkcja celu, opisana w punkcie 2.1.6.

2. Metody rozwiązania

2.1. Algorytm ewolucyjny

2.1.1. Reprezentacja chromosomu

W tworzonym algorytmie ewolucyjnym do rozwiązywania *Sudoku*, planowane jest reprezentowanie pojedynczego chromosomu jako macierzy NxN, gdzie N oznacza rozmiar planszy gry i w przypadku klasycznej wersji Sudoku N wynosi 9. W każdej z komórek macierzy, może być wpisana wartość od 1 do 9, przy uwzględnieniu odpowiednich ograniczeń, które zostaną narzucone przy generacji populacji początkowej.

2.1.2. Populacja początkowa

Do wygenerowania populacji początkowej, potrzebna będzie macierz wejściowa, opisująca początkowy stan gry, gdyż rozwiązywanie Sudoku rozpoczyna się od planszy z daną liczbą komórek wypełnionych odgórnie określoną liczbą. Macierz wejściowa, oprócz komórek z podanymi wartościami początkowymi, będzie wypełniona zerami.
Następnie, każdy kolejny chromosom generowany do populacji początkowej, będzie budowany według następującego schematu:

- na podstawie podanych wartości początkowych, określić zbiór możliwych wartości, które pozostały do wpisania do planszy (brakujących 1,2,3...)
- losować wartości do komórek, iterując po kolejnych wierszach macierzy, tak, aby w jednym wierszu nie było duplikatów

Jeżeli to podejście nie wystarczy do wygenerowania odpowiedniej populacji początkowej, rozważana jest również inna metoda:

- na podstawie podanych wartości początkowych, określić zbiory możliwych wartości do wpisania dla każdej z pozostałych komórek na planszy, biorąc pod uwagę kolizje w wierszach, kolumnach, ale i blokach
- w tym wypadku nie jest zwracana uwaga na duplikaty np. w wierszach

Możliwe jest też połączenie obu tych metod, lecz niezbędne jest wtedy losowanie dopuszczalnego ustawienia w pętli *while* do skutku, co uznano za podejście chciwe.

2.1.3. Selekcja

Do tworzenia kolejnych generacji chromosomów, wykorzystywana będzie selekcja turniejowa, polegająca na losowaniu par chromosomów i wybieraniu do nowej generacji tego osobnika, który zwraca lepszą wartość funkcji celu.

2.1.4. Krzyżowanie

Jako metodę krzyżowania na aktualnym etapie rozważań wybrano krzyżowanie jedno- lub dwupunktowe, interpretowane jako zamiana określonej liczby całych wierszy między dwoma chromosomami (przy przyjętym prawdopodobieństwie krzyżowania). Takie podejście niweluje szanse na pojawienie się duplikatów w wierszach, a także na ingerencję w umiejscowienie podanych wartości początkowych gry.

2.1.5. Mutacja

Planowana metoda mutacji polega na przejściu przez każdy wiersz chromosomu i zamianie wartości dla (co najmniej) jednej pary komórek (z wyłączeniem komórek z narzuconymi wartościami początkowymi), oczywiście uwzględniając przyjęte prawdopodobieństwo mutacji. Taka mutacja również nie pozwala na pojawienie się duplikatów w rzędach, co nie jest jedynym ograniczeniem Sudoku, lecz takie założenie pozwala na lepsze ukierunkowanie przeszukiwań.

2.1.6. Funkcja celu

Funkcja celu, służąca do oceny jakości chromosomu, będzie znajdować kolizję występujące na planszy i zwracać jej liczbę, więc zadaniem algorytmu będzie minimalizacja liczby kolizji w grze Sudoku, najlepiej, aż do znalezienia rozwiązania, czyli planszy bez duplikatów w wierszach, kolumnach oraz blokach.

2.2. Ant Colony Optimization

2.2.1. Reprezentacja grafu przeszukiwań

Grafową przestrzeń przeszukiwań algorytmu ACO dla zadania rozwiązywnia Sudoku, można sobie wyobrazić w ten sposób, że węzły to kolejne komórki planszy gry, gałęzie z nich wychodzące prowadzą do węzłów oznaczających wpisanie danej liczby do komórki, a od każdego z tych węzłów, gałąź prowadzi do kolejnej komórki planszy.
W ten sposób, każda z mrówek, ma narzucony kierunek przeszukiwań jeśli chodzi o kolejność odwiedzanych komórek.
Mrówka to jedna (z określonej liczby) instancja wykonująca algorytm - przeszukująca ścieżki grafu.

2.2.2. Cykl życia mrówki

Podczas inicjalizacji zadania, wybierana jest liczba mrówek wykonujących algorytm. Dla każdej z nich, losowany jest punkt startowy (z całej planszy gry), a następnie wykonywany jest ruch w narzuconym kierunku (np. poprzez iterowanie po kolejnych wierszach i kolumnach planszy) i wybór jednej z dostępnych wartości do danej komórki, na podstawie kosztu na określonej gałęzi i wartości feromonów "pozostawionych" przez mrówki na gałęziach w poprzednich cyklach.

Podróż mrówek w cyklu kończy się po odwiedzeniu przez każdą z nich wszystkich komórek planszy. W ten sposób, po zakończeniu iteracji, wybierana jest mrówka, która uzyskała najlepszą wartość funkcji celu (w tym przypadku maksymalizacja liczby komórek ze znalezionym optymalnym rozwiązaniem) i na podstawie jej trasy, aktualizowane są globalne wartości feromonów, które mają wpływ na decyzje nowych mrówek, w kolejnych cyklach.

2.2.3. Wykonanie ruchu

Poprawność wykonanego ruchu jest zweryfikowana zgodnie z zasadami gry w *Sudoku*. Po wczytaniu planszy początkowej z wpisanymi liczbami, dla każdej komórki wyliczane są 'naiwnie poprawne' liczby, a dokładniej **niekolidujące** z wpisanymi już liczbami. Jeśli w danej komórce są dostępne jakieś liczby, wybór jednej spośród nich opiera się na losowaniu z wagami, którymi są wartości 'feromonów' dla tych liczb w danym momencie. Po wylosowaniu jednej z dostępnych liczb w danej komórce (o ile taka jest), dostępne liczby dla innych komórek w tym wierszu, kolumnie oraz kwadracie zostaną zaktualizowane (wybrana liczba zostanie z nich wykluczona zgodnie z zasadami). Jeśli dla wybranej komórki nie ma żadnej poprawnej liczby, to pozostanie ona pusta, a komórka zostaje uznana za niepoprawną.

3. Eksperymenty numeryczne

W ramach eksperymentów numerycznych planowane jest przygotowanie kilku zestawów *Sudoku* o różnym poziomie trudności. Analizowane wyniki będą efektem wielokrotnej symulacji każdego z algorytmów, co pozwoli zminimalizować błędy statystyczne (oba algorytmy są niedeterministyczne). Do oceny jakości przeprowadzonych eksperymentów wykorzystane zostaną dane dotyczące złożoności obliczeniowej (czas, liczba iteracji), oraz przyrostu wartości funkcji celu w kolejnych cyklach/generacjach.

4. Wykorzystana technologia

Wykorzystany zostanie język programowania ***Python*** wraz z biblioteką ***numpy***.
Niewykluczone jest także skorzystanie z gotowych generatorów stanów początkowych gry Sudoku.