PROBLEM SUMY PODZBIORU

Osoby realizujące:
Rafał Malin,
Albert Podraza,
Nikodem Tokarski.

TEORIA

W problemie sumy podzbioru jest NP-zupełnym problemem arytmetycznym. Istnieje kilka jego interpretacji lecz najbardziej ogólna to: mamy dany zbiór skończony $S \subset N$ i wartość docelową $t \in N$. Pytamy, czy istnieje podzbiór $S' \subset S$, którego suma elementów wynosi t. Jeśli na przykład S = { 1, 10, 15, 87, 13, 156, 21}, a t = 26, to podzbiór S' = {1, 10, 15} jest rozwiązaniem problemu. Należy pamiętać, że w kodowaniu standardowym, wejściowe liczby całkowite są kodowane jako ciągi binarne. Mając na uwadze to założenie, można pokazać, że szybki algorytm dla tego problemu najprawdopodobniej nie istnienie.

Algorytm przygotowany przez nasz zespół rozpatruje wariant problemu gdzie t = 0.

OPIS PRZYJĘTEJ METODY

1. Opis osobnika

Biorąc pod uwagę fakt, że w algorytmach genetycznych bierze udział bardzo dużo osobników staraliśmy się by reprezentacja pojedynczego chromosomu była jak najprostsza. Chromosom pojedynczego osobnika reprezentuje tablica wypełniona pseudolosowo wartościami 0 lub 1. Mając *n* liczb wprowadzonych do algorytmu, chromosom ma długość *n* bitów, przy czym i-ty bit o wartosci 1 oznacza branie pod uwagę (sumowanie) i-tego elementu wejściowego zbioru liczb.

Dla przykładu: podajemy zbiór {12, 87, 35, 98, 14, 54, 67}.

Chromosom [1001110] reprezentuje podzbiór {12,98,14,54}.

2. Losowanie populacji

W tym miejscu ustalamy jak wielka będzie tworzona populacja. Jeśli populacja będzie zawierała zbyt mało osobników to algorytm może zatrzymać się w jakimś lokalnym minimum (gdy jakieś dobre lecz nie najlepsze rozwiązanie zdominuje całą populację). Z drugiej strony zbyt duża liczebność populacji zmniejsza szybkość działania algorytmu. By zapewnić jak największą różnorodność chromosomów, początkowa populacja składa się z chromosomów, których wszystkie geny (bity) są losowane (0 lub 1).

3. Ocena osobników

W tym kroku badamy wartość przystosowania każdego osobnika z populacji. Wartość ta jest oceniana przez funkcję przystosowania dla każdego chromosomu. Wartość przystosowania chromosomu jest tym większa, im bliższa jest zeru suma wybranych liczb, a więc im mniejsza wartość przystosowania osobnika, tym jest on lepiej przystosowany.

4. Selekcja

Krok najważniejszy dla całej genetyki. Tworzymy nową populację na podstawie już istniejącej. W zależności od wartości funkcji oceny obliczanej w poprzednim kroku, dany osobnik ma większe lub mniejsze szanse na dopuszczenie do krzyżowania.

Szansę poszczególnych osobników obliczamy na zasadzie koła ruletki, która polega na n-krotnym losowaniu osobników ze starej populacji, gdzie osobniki mają różne prawdopodobieństwa wylosowania.

Prawdopodobieństwo dostania się chromosomu do krzyżowania obliczamy ze wzoru:

$$probability = 1 - \frac{\text{fitness\_value}}{\text{sum\_of\_all\_fitness\_values}}$$

gdzie:

fitness_value – wartość przystosowania aktualnie rozpatrywanego chromosomu

sum_of_all_fitness_values – obliczona suma wartości przystosowania wszystkich chromosomów osobników w populacji.

5. Krzyżowanie

Zadaniem kroku krzyżowania jest wymiana "materiału genetycznego" pomiędzy dwoma rozwiązaniami w populacji. W wyniku krzyżowania na podstawie dwóch rozwiązań (rodzice) tworzone są dwa nowe osobniki (dzieci). Po wykonaniu krzyżowania dzieci zastępują w populacji rodziców, tworząc tym samym nową populację.

Algorytm przez nas przedstawiony stosuje krzyżowanie jednopunktowe, połowiczne. Mając binarną reprezentację chromosomu, zaimplementowane krzyżowanie polega na podziale dwóch chromosomów na dwie połowy z których algorytm tworzy nowe chromosomy. Pierwsze dziecko składa się z początkowej części pierwszego rodzica i końcówki drugiego natomiast drugie dziecko odwrotnie - początek drugiego rodzica i koniec pierwszego. Dla przykładu mając chromosomy rodziców: [10011101] i [10111000] dzieci będą wyglądały następująco: [10011000] i [10111101].

6. Mutacja

Mutacja polega na zmianie dokładnie jednego, losowo wybranego bitu w chromosomie osobnika. Każdy osobnik ma szansę na zmutowanie równą *mutateProbability* – wartość możliwa do zmiany w GUI programu. Numer bitu do inwersji wybierany jest losowo.

7. Powrót do oceny osobników(2) lub wyjście z pętli

Teraz sprawdzany jest warunek stopu algorytmu. Jeśli którykolwiek chromosom w populacji wskazuje na sumę liczb równą 0, algorytm kończy się. W przeciwnym razie nowa populacja stworzona przez krzyżowanie i mutację przekazywana jest jako populacja wejściowa do oceny osobników, a algorytm zaczyna kolejną iterację.

Algorytm w każdej populacji szuka najlepszego chromosomu (czyli takiego, który wskazuje na sumę najbliższą zeru) i zapamiętuje go w tablicy. Od momentu znalezienia najlepszego rozwiązania uruchamia licznik. Jeśli dojdzie on do wartości ustalanego parametru *searchTolerance,* algorytm kończy się z wynikiem najniższej sumy z zapamiętanej tablicy. Zawartość tej tablicy jest rysowana na wykresie, który przedstawia wykres zbliżających się sum do zera.