Laboratorium 4

Algorytm genetyczny w problemach optymalizacji dyskretnej

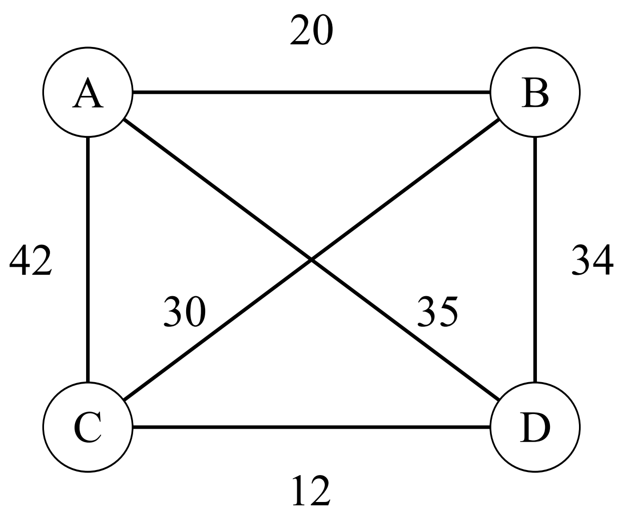
Problemami optymalizacji dyskretnej nazywamy problemy w których zmienne decyzyjne przyjmują wartości całkowitoliczbowe lub binarne. Do tej klasy problemów należy wiele praktycznych zagadnień m.in. z dziedziny szeregowania zadań, czy teorii grafów. Niestety większość tego typu problemów stanowią problemy *NP-trudne*, których złożoność obliczeniowa rośnie wykładniczo, wraz ze wzrostem liczby danych wejściowych. Powoduje to, że dla większości tego typu zagadnień jesteśmy zmuszeni do zastosowania algorytmów aproksymacyjnych, zwracających przybliżone rozwiązanie problemu w rozsądnym czasie. Do algorytmów aproksymacyjnych należą m.in. optymalizacja rojem cząstek, symulowane wyżarzanie, czy algorytmy genetyczne.

4.1 Problem komiwojażera

Problem komiwojażera jest jednym ze znanych NP-trudnych problemów optymalizacji dyskretnej. Zagadnienie polega na znalezieniu najkrótszego cyklu w pełnym grafie ważonym, obejmującego wybrane wierzchołki. Wierzchołki w takim grafie mogą reprezentować np. miasta, a wagi poszczególnych krawędzi odległości między miastami.

W celu zobrazowania problemu wyobraźmy sobie graf z czterema wierzchołkami: A, B, C, D oraz krawędziami, których wagi są przedstawione na Rysunku 4.1. Naszym zagadnieniem jest odnalezienie najkrótszej ścieżki rozpoczynającej się w wierzchołku A i przechodzącej przez pozostałe trzy wierzchołki: B, C, D. Musimy tutaj rozważyć następujące możliwości:

1. A → B → C → D → A: 97
2. A → B → D → C → A: 108



Rysunek 4.1: Przykładowy pełny graf ważony dla problemu komiwojażera.

3. A → C → B → D → A: 141
4. A → C → D → B → A: 108
5. A → D → B → C → A: 141
6. A → D → C → B → A: 97

Jak widzimy najkrótsza w tym przykładzie okazała się ścieżka: A → B → C → D → A (oraz jej symetryczne odbicie), której kosz wynosi: 97.

Złożoności najlepszego algorytmu dokładnego rozwiązującego problem komiwojażera wynosi: $O(n^2 2^n)$, gdzie n oznacza liczbę rozważanych wierzchołków. Jak widzimy czas działania algorytmu rośnie wykładniczo wraz ze w zrostem liczby rozważanych wierzchołków, co czyni go niepraktycznym już dla stosunkowo niewielkich n.

4.2 Algorytm genetyczny

Algorytmy genetyczne czerpią swoją inspiracje z biologicznego procesu doboru naturalnego. W ramach kolejnych iteracji (zwanych generacjami), algorytm genetyczny próbuje znaleźć optymalne rozwiązanie dla zadanego problemu, poprzez łączenie ze sobą dotychczasowy rozwiązań.

Zanim przejdziemy do szczegółowego omówienia algorytmu, zdefiniujmy kilka pojęć wykorzystywanych przez algorytmy genetyczne w kontekście problemu komiwojażera:

- Gen – pojedynczy wierzchołek w grafie.
- Osobnik – ścieżka łącząca wszystkie wierzchołki w grafie.
- Populacja – zbiór możliwych ścieżek (zbiór osobników).
- Rodzice – dwie ścieżki wykorzystywane do stworzenia nowej ścieżki.

Pojedyncza iteracja algorytmu genetycznego składa się z następujących kroków:

1. Selekcja – wybranie najlepszych osobników z danej populacji jako rodziców następnego pokolenia.
2. Krzyżowanie – łączenie ze sobą losowo wybranych rodziców w celu utworzenia nowych osobników stanowiących kolejną generację.
3. Mutacja – wprowadzenie drobnych losowych zmian do nowo powstałej populacji.

Działanie algorytmu rozpoczyna się od wylosowania pierwszego pokolenia osobników, a następnie wykonywana jest zadana liczba iteracji. Najlepszy osobnik z ostatniej generacji stanowi rozwiązanie zadanego problemu. Liczba iteracji oraz wielkość pojedynczej populacji są parametrami algorytmu.

Często w celu uzyskania lepszych parametrów algorytmu stosuje się tzw. strategie z częściową reprodukcją. Najpopularniejszym podejściem jest elityzm, w którym pewna liczba najlepszych osobników z danej populacji jest przenoszona bez zmian do kolejnej generacji.

4.2.1 Selekcja

Selekcja jest pierwszym etapem iteracji algorytmu genetycznego. W ramach tego kroku dokonujemy wyboru najlepiej przystosowanych osobników, którzy następnie zostaną wykorzystani w etapie krzyżowania. Istnieje kilka możliwych sposobów dokonania selekcji. Jednym z nich jest metoda koła ruletki[1].

W metodzie koła ruletki, prawdopodobieństwo wylosowania osobnika jako kandydata do krzyżowania jest proporcjonalne do jego dopasowania. Im lepiej dopasowany jest osobnik, tym większa szansa na jego wybór. W metodzie tej jeden osobnik może zostać wybrany więcej niż jeden raz.

W celu zastosowania metody koła ruletki do problemu komiwojażera, wygodnie jest przekształcić problem z minimalizacyjnego na maksymalizacyjny. W tym celu wystarczy odwrócić naszą funkcję dopasowania:

$$fitness'(path) = 1/fitness(path)$$

Algorytm selekcji metodą koła ruletki dla problemu komiwojażera został przedstawiony poniżej (Algorytm 1).

[1] https://en.wikipedia.org/wiki/Fitness_proportionate_selection

Algorytm 1 Algorytm selekcji metodą koła ruletki dla problemu komiwojażera. Zakładamy, że słownik zachowuje kolejność dodawanych elementów.

Wejście: *population* (poprzednia generacja)
Wyjście: *selection* (osobniki wybrane do etapu krzyżowania)
  $population_{fitness'} \leftarrow$ słowik odwzorowujący poszczególnych osobników na odpowiadające im wartości dopasowania $fitness'$
  $probability \leftarrow$ pusty słownik
  $sum_{fitness} \leftarrow$ suma wartości funkcji dopasowania wszystkich osobników
  $probability_{previous} \leftarrow 0.0$
  for $key, value \leftarrow sorted(population_{fitness'})$ (zaczynamy od najlepiej dopasowanego osobnika) **do**
    $probability[key] \leftarrow probability_{previous} + (value/sum_{fitness})$
    $probability_{previous} \leftarrow probability[key]$
  end for
  for $i \leftarrow 1$ to $len(population)$ **do**
    $rand \leftarrow$ losowa liczba z przedziału od 0 do 1
    for $key, value \leftarrow probability$ **do**
      if $rand <= value$ **then**
        dodaj osobnika *key* do *selection*
        break
      end if
    end for
  end for

W ramach etapu selekcji można zastosować elityzm, wybierając od razu pierwsze N najlepszych osobników niezależnie od wyników losowania.

4.2.2 Krzyżowanie

Etap krzyżowania polega na połączeniu cech dwóch losowo wybranych osobników w celu utworzenia nowego osobnika potomnego. W przypadku problemu komiwojażera, w celu zachowania poprawności nowo powstałej ścieżki (ścieżka musi stanowić listę wszystkich wierzchołków grafu, gdzie każdy z wierzchołków musi wystąpić dokładnie raz), należy zastosować specjalny wariant krzyżowania, zwany krzyżowaniem uporządkowanym.

Krzyżowanie uporządkowane najłatwiej wyjaśnić na przykładzie. Wyobraźmy sobie że chcemy skrzyżować dwa osobniki:

$$\text{parent}_1 = [7, 1, 6, 0, 5, 9, 2, 4, 8, 3]$$

$$\text{parent}_2 = [7, 1, 5, 9, 4, 3, 6, 2, 0, 8]$$

Pierwszym etapem naszego algorytmu jest wylosowanie fragmentu pierwszego rodzica, który następnie zostanie przeniesiony do rodzica drugiego. Załóżmy że wylosowaliśmy następujący fragment:

$$\text{slice} = [5, 9, 2, 4]$$

Następnym krokiem jest usunięcie z rodzica drugiego wszystkich elementów znajdujących się w wylosowanym fragmencie:

$$\text{offspring} = [7, 1, 3, 6, 0, 8]$$

Ostatecznie wstawiamy wylosowany fragment w odpowiednie miejsce pierwszego rodzica, tak aby liczba elementów znajdujących się za wylosowanym fragmentem pierwszego rodzica, była równa liczbie elementów znajdujących się za fragmentem w rozwiązaniu potomnym:

$$\text{offspring} = [7, 1, 3, 6, \mathbf{5}, \mathbf{9}, \mathbf{2}, \mathbf{4}, 0, 8]$$

Krzyżowanie osobników powtarzamy aż do zapełnienia całej nowej populacji (rozmiary populacji muszą się zgadzać pomiędzy kolejnymi iteracjami algorytmu). Tak samo jak w przypadku selekcji możemy tutaj wykorzystać elityzm i nie poddawać zmianom najlepszych osobników.

4.2.3 Mutacja

Ostatnim etapem algorytmu genetycznego jest mutacja. Etap ten polega na wprowadzeniu niewielkich losowych zmian do wygenerowanego pokolenia potomnego. W problemie komiwojażera można zastosować metodę polegającą na zamianie miejscami dwóch losowych elementów. Prawdopodobieństwo z jakim zmiana może zostać dokonana jest jednym z parametrów algorytmu. Algorytm 2 realizuje mutację pojedynczego osobnika.

Algorytm 2 Algorytm mutacji pojedynczego osobnika dla problemu komiwojażera.

Wejście: $individual$ (osobnik podlegający mutacji), $mutation_{rate}$ (prawdopodobieństwo dokonania mutacji pojedynczego genu)
Wejście: $individual_{mutated}$ (osobnik po mutacji)

```
for i ← 1 to liczba elementów ścieżki do
    rand ← losowa liczba z przedziału od 0 do 1
    if rand <= mutation_rate then
        swap ← nowa losowo wybrana pozycja dla wierzchołka i
        zamień miejscami na ścieżce wierzchołki na pozycjach i i swap
    end if
end for
```