

Wykorzystanie algorytmu genetycznego w rozwiązywaniu problemu QAP.

Franciszek Magiera, Dawid Lech

09.01.2019

Spis treści

1	Wstęp	3
2	Model matematyczny	3
3	Opis algorytmu	4

1 Wstęp

Naszym zadaniem było znalezienie takiego rozmieszczenia fabryk, znając przepływ materiałów oraz odległości między nimi, aby energia potrzebna do transportu materiałów była minimalna. Jest to przypadek NP-trudnego problemu QAP. Rozwiązania problemu poszukiwaliśmy przy pomocy algorytmu genetycznego.

2 Model matematyczny

Mamy dany zbiór n fabryk oraz n dostępnych na budowę miejsc, które numerujemy od 1 do n . Dana jest również macierz przepływu $n \times n$, której element w i -tej kolumnie i j -tym wierszu oznacza ilość transportowanych materiałów z fabryki oznaczonej numerem i do fabryki oznaczonej numerem j oraz symetryczną macierz odległości pomiędzy poszczególnymi miejscami, również o wymiarach $n \times n$, której element w i -tej kolumnie i j -tym wierszu oznacza odległość miejsca oznaczonego numerem i od miejsca oznaczonego numerem j (w obydwu przypadkach $i, j \in [1, n]$). Rozwiązania problemu szukamy w postaci n -elementowej permutacji, gdzie element na i -tym miejscu o wartości j oznacza przyporządkowanie fabryki o j -tym numerze na miejsce o i -tym numerze. Oznaczając macierz przepływu materiałów pomiędzy fabrykami przez W , macierz odległości pomiędzy dostępnymi miejscami przez D oraz permutację będącą rozwiązaniem poprzez s , funkcję celu możemy zapisać w postaci:

$$F(s) = \sum_{i=1}^n \sum_{j=1}^n D_{i,j} W_{s(i),s(j)} \longrightarrow \min \quad (1)$$

W przypadku symetrycznych macierzy przepływu można by zmodyfikować powyższą funkcję dzieląc ją przez 2, bądź rozpoczynając drugie sumowanie od $j=i$ do n , co skróciłoby czas potrzebny na obliczenie wartości funkcji celu.

3 Opis algorytmu

Aby znaleźć optymalne rozwiązanie opisanego problemu, przeszukujemy przestrzeń dopuszczalnych rozwiązań. Korzystamy przy tym z algorytmu genetycznego, należącego do szerszej klasy algorytmów ewolucyjnych. Algorytm genetyczny wymaga odpowiedniej reprezentacji przestrzeni rozwiązań, dla której elementów łatwo jest wykonać proste operacje krzyżowania, mutacji itp. oraz funkcji celu, co sprawia, że dobrze nadaje się do rozwiązania naszego problemu.

Algorytmy genetyczne wykorzystują mechanizmy zainspirowane ewolucją biologiczną takie jak krzyżowanie, mutacja czy selekcja. Przykładowe rozwiązanie problemu można skojarzyć z osobnikiem należącym do populacji (zbioru kilku rozwiązań). Funkcję celu można rozumieć jako miarę przystosowania danego osobnika - determinuje ona jakość rozwiązania. Populacja początkowa jest generowana losowo. Ewolucja populacji zachodzi po wielokrotnym stosowaniu wcześniej wymienionych mechanizmów. W efekcie z każdą generacją uzyskujemy coraz to lepiej przystosowaną populację (składającą się z osobników o niższych wartościach funkcji celu).

Potencjalnym problemem na jaki można się natknąć wykorzystując algorytm genetyczny jest zbyt szybkie osiągnięcie jednolitej populacji. Z jednej strony prowadzi to do szybkiego znalezienia rozwiązania, jednak najczęściej jest ono jedynie optimum lokalnym dalekim od rozwiązania optymalnego. Aby uniknąć tego problemu należy zadbać o losowość działania algorytmu np. poprzez dodanie mutacji oraz stosować techniki selekcji, które utrzymują zróżnicowaną populację.

Czynnikami limitującymi stosowanie algorytmów genetycznych jest złożoność funkcji celu, która musi być wielokrotnie obliczana podczas wykonania algorytmu oraz rozmiar problemu do którego silni proporcjonalny jest rozmiar przestrzeni rozwiązań.

Należy również pamiętać, że algorytmy genetyczne to metody przybliżone, co oznacza, że nie mamy pewności co do tego czy znalezione przez nas rozwiązanie jest optymalne, a nawet jeśli takie znaleźliśmy, to nie możemy tego stwierdzić.

Stworzony przez nas algorytm podąża według następującego schematu: wczytujemy parametry algorytmu takie jak prawdopodobieństwo mutacji i rodzaj operatorów z pliku, później losowo generujemy bądź wczytujemy wcześniej wygenerowaną populację o określonych rozmiarach. Następnie w pętli wykonujemy kolejno selekcję n rodziców z dotychczasowej populacji, następnie n razy wybieramy losowo dwóch rodziców, których krzyżujemy ze sobą. Ich potomków z pewnym prawdopodobieństwem mutujemy, a następnie korzystając z operatora sukcesji wybieramy osobników, którzy przechodzą do następnej populacji (kolejnej generacji). Pętlę zatrzymujemy po wykonaniu określonej ilości iteracji, bądź gdy wartość najlepszego rozwiązania nie zmieni się przez określoną ilość iteracji. Schemat ten przedstawiamy poniżej w postaci pseudokodu: