

Algorytmy ewolucyjne

Zasada działania, obszary zastosowań, biblioteki programistyczne, zastosowanie do wybranego problemu optymalizacyjnego oraz eksperymenty

Robert Smoter

<https://github.com/SMOKURsmoter/DEAP>

1. Zasada działania i zastosowania

Algorytmy ewolucyjne są metaheurystykami inspirowanymi procesami ewolucji biologicznej, które służą do rozwiązywania problemów optymalizacyjnych i przeszukiwania przestrzeni rozwiązań.

Zasadnicza idea polega na iteracyjnym ewoluowaniu populacji osobników, gdzie każdy osobnik reprezentuje potencjalne rozwiązanie problemu. Algorytmy ewolucyjne opierają się na trzech podstawowych operacjach: reprodukcji, rekombinacji i mutacji.

```
t := 0;
wygenerowanie A(t);
ocena A(t);
while not warunek stopu do
begin
  A1(t) := reprodukcja A(t);
  A2(t) := rekombinacja A1(t);
  A3(t) := mutacja A2(t);
  ocena A3(t);
  A(t+1) := sukcesja (A3(t) ∪ A4(t));
  t := t + 1;
end
```

Algorytmy ewolucyjne mają szerokie zastosowanie w różnych dziedzinach, takich jak:

1. Optymalizacja problemów: optymalizacja funkcji matematycznych, planowanie produkcji, zarządzanie energią, itp.
2. Projektowanie parametrów; znajdują zastosowanie w automatycznym projektowaniu i optymalizacji parametrów w modelach i systemach.
3. Tworzenie sztucznej inteligencji:
4. Projektowanie układów elektronicznych; takich jak układy scalone, filtry czy anteny.

2. Biblioteki programistyczne

DEAP (Distributed Evolutionary Algorithms in Python) to popularna biblioteka programistyczna napisana w języku Python, która umożliwia implementację różnorodnych algorytmów ewolucyjnych.

Biblioteka zapewnia narzędzia do implementacji różnych typów algorytmów, takich jak ;

Algorytmy genetyczne (GA) - technika optymalizacji inspirowana procesami ewolucyjnymi w naturze. Wykorzystuje operatory selekcji, krzyżowania i mutacji do iteracyjnego przeszukiwania przestrzeni rozwiązań w celu znalezienia najlepszego rozwiązania. GA znajduje zastosowanie w różnych dziedzinach(optymalizacja problemów, projektowanie parametrów, planowanie produkcji itp.)

Programowanie genetyczne (GP); metoda tworzenia programów komputerowych przy użyciu technik inspirowanych ewolucją biologiczną. GP ewoluuje populację programów, wykorzystując operatory genetyczne, takie jak krzyżowanie i mutacja. Celem jest znalezienie programu, który najlepiej rozwiązuje dany problem. GP znajduje zastosowanie w dziedzinach takich jak tworzenie sztucznej inteligencji, generowanie programów, projektowanie układów elektronicznych itp.

Strategia ewolucyjna (ES) - rodzaj algorytmu ewolucyjnego, który skupia się na optymalizacji problemów ciągłych. ES wykorzystuje operatory mutacji i rekombinacji, aby iteracyjnie przeszukiwać przestrzeń rozwiązań. ES różni się od innych algorytmów ewolucyjnych poprzez swoje specyficzne operatory i strategie selekcji. Znajduje zastosowanie w optymalizacji problemów ciągłych, jak np. optymalne zarządzanie energią, optymalizacja funkcji matematycznych itp.

Optymalizacja rojem cząstek (PSO) - technika optymalizacji inspirowana zachowaniem stada lub roju. PSO symuluje zachowanie cząstek poruszających się w przestrzeni rozwiązań, które poruszają się w kierunku najlepszego rozwiązania na podstawie informacji o lokalnym i globalnym najlepszym rozwiązaniu. PSO jest wykorzystywane do optymalizacji problemów ciągłych i dyskretnych.

Algorytmy estymacji rozkładu (EDA) są rodzajem algorytmów ewolucyjnych, które wykorzystują modele statystyczne do estymacji rozkładu prawdopodobieństwa rozwiązań. EDA iteracyjnie tworzy modele rozkładu na podstawie populacji rozwiązań, a następnie generuje nowe rozwiązania na podstawie tych modeli. EDA znajdują zastosowanie w różnych dziedzinach, takich jak optymalizacja, sztuczna inteligencja, analiza danych itp.

Zalety Biblioteki

- Prosta w użyciu: oferuje czytelną i intuicyjną składnię, co ułatwia tworzenie algorytmów ewolucyjnych.
- Elastyczna; zapewnia elastyczność w dostosowywaniu algorytmów do różnych problemów poprzez konfigurowalne operatory i narzędzia.
- Wydajna: zoptymalizowana pod kątem wydajności, co umożliwia efektywne przetwarzanie dużych populacji i złożonych problemów optymalizacyjnych

Jenetics to zaawansowana biblioteka algorytmów genetycznych, algorytmów ewolucyjnych i programowania genetycznego, napisana w nowoczesnej wersji języka Java. Została zaprojektowana z wyraźnym podziałem na różne koncepcje algorytmów, takie jak gen, chromosom, genotyp, fenotyp, populacja i funkcja dopasowania. Jenetics umożliwia minimalizację lub maksymalizację zadanej funkcji dopasowania bez konieczności jej modyfikacji. W przeciwieństwie do innych implementacji algorytmów genetycznych, biblioteka wykorzystuje koncepcję strumienia ewolucji (EvolutionStream) do wykonywania kroków ewolucji.

3. Zaimplementowany algorytm rozwiązujący wybrany problem optymalizacyjny

Jako modelowy problem wybrano problem komiwojażera (TSP - ang. Traveling Salesman Problem), który jest klasycznym problemem optymalizacyjnym w informatyce i matematyce. Polega na znalezieniu najkrótszej możliwej trasy dla handlowego podróżnika, który musi odwiedzić zbiór miast i powrócić do miasta startowego, pokonując każde miasto dokładnie raz.

Do rozwiązania problemu najkrótszej trasy pomiędzy przykładowymi miastami w Polsce użyto algorytm ewolucyjny o nazwie `eaSimple` z biblioteki DEAP. Algorytm `eaSimple` jest jednym z podstawowych algorytmów ewolucyjnych, który stosuje operatory selekcji, krzyżowania i mutacji do ewolucji populacji.

```
algorithms.eaSimple(pop, toolbox, 0.7, 0.2, 40, stats=stats, halloffame=hof)
```

algorytm został zainicjowany parametrami

- `toolbox`: zestaw narzędzi i operatorów ewolucyjnych
- `0.7`: prawdopodobieństwo krzyżowania (70%)
- `0.2`: prawdopodobieństwo mutacji (20%)
- `40`: liczba pokoleń
- `stats=stats`: obiekt przechowujący statystyki ewolucji
- `halloffame=hof`: obiekt przechowujący najlepsze rozwiązanie (najlepszego osobnika)

4. Wyniki eksperymentów

Do eksperymentu wykorzystano następujący kod

<https://github.com/SMOKURsmoter/DEAP/blob/main/komiwojadzer.ipynb>

W wyniku użycia programu uzyskano wyniki:

Gen	nevals	avg	std	min	max
0	300	1776	140.442	1550	2050
1	221	1714.33	128.301	1550	2050
2	206	1680.67	117.514	1550	2050
3	225	1684.83	134.394	1550	2050
4	235	1681.33	136.815	1550	2050
5	234	1665.33	124.358	1550	2050
6	226	1655	126.194	1550	2050
7	242	1669.17	142.154	1550	2050
8	217	1654	133.544	1550	2000
9	233	1649.33	136.746	1550	2050
10	235	1657.17	131.809	1550	2050
11	242	1637.83	117.162	1550	2000
12	221	1639.5	121.304	1550	2050
13	231	1627.83	118.885	1550	2050
14	237	1642.33	131.686	1550	2050
15	234	1600.83	103.719	1550	2050
16	228	1587	87.3556	1550	1900
17	236	1582.17	80.4589	1550	1900
18	232	1581.5	78.9478	1550	1900
19	234	1575.5	72.6275	1550	1900
20	218	1568.17	61.0599	1550	1900
21	239	1579.83	84.2712	1550	1850
22	227	1564.83	61.0052	1550	1850
23	229	1562.33	54.4477	1550	1850
24	223	1556.5	39.7838	1550	1850
25	220	1559.5	47.7991	1550	1850
26	237	1562.33	55.8082	1550	1900
27	215	1562.33	55.3584	1550	1850
28	252	1564	60.4483	1550	1850
29	240	1558.67	43.6832	1550	1900
30	227	1559.33	53.6615	1550	2000
31	215	1560.5	54.9067	1550	2000
32	228	1561.33	53.7422	1550	1900
33	219	1560	50.4975	1550	1850
34	242	1558.67	44.8132	1550	1850
35	250	1558.83	48.1003	1550	1850
36	227	1557.67	43.8698	1550	2000
37	244	1559.33	47.7447	1550	1850
38	237	1557.5	43.229	1550	1850
39	224	1560.17	50.5467	1550	1850
40	243	1560	48.9898	1550	1850

Najlepsza trasa: [4, 2, 1, 0, 3]

Najlepsza odległość: 1550

Wyniki eksperymentu pokazują przebieg ewolucji przez 40 pokoleń. Osiągnięte wyniki prezentują wartości dla średniej (avg), odchylenia standardowego (std), wartości minimalnej (min) i maksymalnej (max) dla funkcji celu, która jest odległością trasy dla problemu komiwojażera.

W początkowych pokoleniach można zauważyć wzrost wartości średniej odległości, co sugeruje, że populacja jeszcze nie znalazła optymalnej trasy. Jednak wraz z postępem ewolucji, średnia odległość maleje, co oznacza, że populacja ewoluuje w kierunku lepszych rozwiązań.

Odchylenie standardowe maleje wraz z postępem ewolucji, co wskazuje na stopniowe zmniejszanie się różnorodności w populacji. Oznacza to, że populacja koncentruje się wokół lepszych rozwiązań.

Najlepsze rozwiązanie (najmniejsza odległość) znalezione w całym procesie ewolucji wynosi 1560. Poniżej przedstawiono optymalną trasę, która przechodzi przez miasta o indeksach 4, 2, 1, 0, 3. Dodatkowo, przedstawiono odległości między kolejnymi miastami na tej trasie.

Ogólnie rzecz biorąc, wyniki eksperymentu pokazują, że algorytm ewolucyjny był skuteczny w rozwiązaniu problemu komiwojażera dla danego zestawu danych, znajdując optymalną trasę o minimalnej odległości.

Bibliografia

1. Zhi-Hua Zhou, Yang Yu, Chao Qian, *Evolutionary Learning: Advances in Theories and Algorithms*, Springer, 2019
2. Devin Soni, *Introduction to Evolutionary Algorithms*, <https://towardsdatascience.com/introduction-to-evolutionary-algorithms-a8594b484ac>, Dostęp: 15.03.2021
3. Dokumentacja biblioteki DEAP <https://deap.readthedocs.io/en/master/>
4. Dokumentacja biblioteki Jenetics <https://jenetics.io/>
5. dr hab. inż. Rafał Dreżewski, Wykład *Sztuczna inteligencja i systemy ekspertowe*, WSZIB 2023