Analiza skupień w populacjach generowanych klasycznym algorytmem ewolucyjnym

Piotr Bródka, Jacek Myna

June 1, 2020

Spis treści

1 Założenia projektu

Celem projektu była analiza skupień dla Mutacyjnego Algorytmu Ewolucyjnego, który optymalizuje dowolną funkcję celu w R^n.

Autorzy zbadali dynamikę populacji podczas działania algorytmu. Sposób interpretacji pojęcia *dynamika populacji* został przedstawiony w dalszej części dokumentu.

1.1 Zmienne w eksperymentach

Badanie zostało zilustrowane na przykładzie ewolucji dla funkcji celu będących:

- funkcją stałą,
- funkcją Gaussa,
- sumą dwóch funkcji Gaussa (o niekoniecznie tym samym współczynniku σ),
- funkcja Rastrigina: $f(x) = An + \sum_{i=1}^{n}[x_i^n - Acos(2\pi x_i)]$

.

Oprócz wyboru funkcji celu, zmiennymi podczas analizy były również:

- wielkość turnieju wykorzystana w selekcji turniejowej,
- wymiar przestrzeni (zbadany dla n=1, 2, 3)

1.2 Narzędzia i techonologie

Do implementacji *algorytmu ewolucyjnego* i analizy klastrów został wykorzystany język Python3 wraz z bibliotekami numerycznymi (np. *NumPy* oraz *Pandas*) i wizualizacyjnymi (np. *MatplotLib*), a także biblioteka *scikit-learn* służąca do uczenia maszynowego.

2 Załączniki

W raporcie pojawiły się odwołania do plików zewnętrznych z wizualizacjami. Pliki znajdują się w folderach:

- **populations**: w tym folderze dla każdego eksperymentu zostały zapisane wykresy funkcji wraz z elementami populacji w kolejnych iteracjach (na przykład w folderze o nazwie **rastrigin_A_10.0_dim_2_tournament_2** znajdują się wyniki dla funkcji Rastrigina dla wymiaru problemu 2 i rozmiaru turnieju 2),
- **graphs**: folder z dokumentami .pdf zawierającymi wykresy, wizualizacje wyników klastrowania

3 Metodologia analizy skupień

W ramach analizy klastrów kluczowe było rozwiązanie dwóch podproblemów:

- wyznaczenie optymalnej liczby klastrów,
- wybór cech, które będą opisywały klastry.

Jako metoda segmentacji został wykorzystany algorytm K-średnich. Początkowo w każdym eksperymencie do wyznaczania optymalnej liczby skupień posługiwano się zarówno wykresem statystyki **Gap** [1] jak i klasyczną miarą odległości pomiędzy elementami należącymi do tego samego klastra daną wzorem:

$$WCSS = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} d_{ij}$$

gdzie $WCSS$ jest sumą kwadratów odległości między obserwacjami należącymi do tego samego skupienia. Co ciekawe algorytm K-średnich opiera się na obserwacji, iż sumę $WCSS$ można zastąpić wyrażeniem:

$$\sum_{i=1}^{n} d(x_i, m_{C(i)})$$

będącym sumą odległości pomiędzy i-tą obserwacją, a środkiem skupienia, do którego należy dana obserwacja [2]. Suma $WCSS$ liczona dla różnej liczby skupień jest natomiast wykorzystywana do wyznaczania optymalnej liczby K.

Planowane wyznaczanie liczby klastrów za pomocą statystyki Gap sprawiło następujące problemy:

- W zdecydowanej większości eksperymentów optymalna liczba klastrów zwracana przy pomocy statystyki Gap była równa 1.
- Po modyfikacji systemu, tak by nie zwracał on jednego klastra, a co najmniej dwa, pojawił się inny problem. Liczba klastrów w kolejnych iteracjach nie była stała. Problematyczne było badanie własności klastrów, gdy w iteracji k były tworzone 4 klastry, w iteracji $k+1$ - 6 klastrów, a w iteracji $k+2$ - 4 klastry.

W tej sytuacji autorzy postanowili ustalić "na sztywno" liczbę klastrów na 1, 2, 3, 4 i 5. Cechami opisującymi klastry podczas działania algorytmu ewolucyjnego były:

- liczba elementów należących do klastra,
- odchylenie standardowe wewnątrz klastra,
- odległości klastrów od maksimum funkcji.

Badanie dynamiki w klastrach polegało zatem na:

1. Posortowaniu klastrów malejąco pod względem liczności
2. Wypisaniu dla kolejnych iteracji:
    - liczby elementów należących do klastra,
    - odchylenia standardowego wewnątrz klastrów,
    - odległości klastrów od maksimum funkcji.

4 Stałe i zmienne w eksperymentach

4.1 Parametry algorytmu ewolucyjnego

Parametrami algorytmu ewolucyjnego o stałych wartościach podczas eksperymentów były:

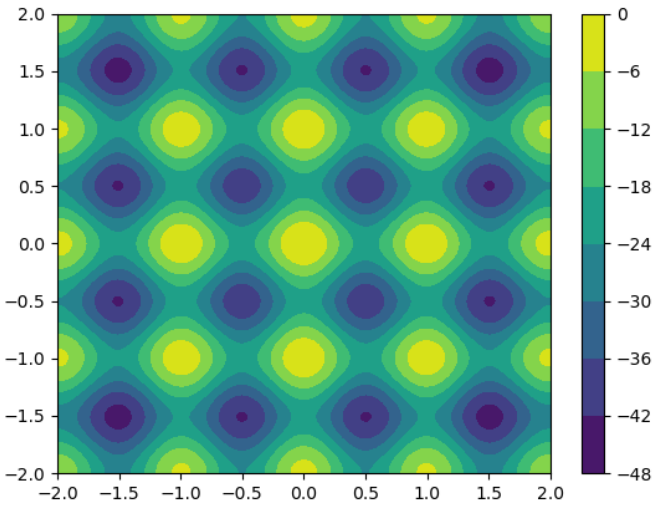
- rozmiar populacji: 200,
- współczynnik sigma w mutacji gaussowskiej: 1.0,
- prawdopodobieństwo krzyżowania: 0.5,
- liczba iteracji: 100,
- selekcja turniejowa,
- sukcesja generacyjna,
- szukamy maksimum funkcji,

4.2 Funkcje celu

Badanie zostało zilustrowane na przykładzie ewolucji dla funkcji celu będących:

- funkcją stałą: $f(x) = 1$,
- funkcją Gaussa, parametry: $\mu = 0.0, \sigma = 0.8$,
- sumą dwóch funkcji Gaussa, parametry: $\mu_1 = 0.0, \sigma_1 = 0.8, \mu_2 = 4.0, \sigma_2 = 1.0$
- funkcja Rastrigina: $f(x) = -(An + \sum_{i=1}^{n}[x_i^n - Acos(2\pi x_i)]),\ A = 10$

Uwaga 1: Funkcje (poza funkcją stałą) mają swoje maksimum w punkcie **0**.
Uwaga 2: Funkcja Rastrigina w swojej najpopularniejszej formie to $f(x) = An + \sum_{i=1}^{n}[x_i^n - Acos(2\pi x_i)]$, jednak w tej formie w punkcie **0** ma ona swoje minimum. Stąd minus na początku - by zamienić zadanie na maksymalizację,
Uwaga 3: Funkcje można "podejrzeć" (dla wymiaru dziedziny 1 oraz 2) w odpowiednich plikach w folderze populations.
Uwaga 4: Funkcja Rastrigina cechuje się dużą liczbą ekstremów lokalnych, w szerokiej perspektywie na wykresach nie widać tego dokładnie, na poniższym wykresie można zobaczyć, jak zachowuje się ta funkcja bliżej zera:



4.3 Inne zmienne

Oprócz wyboru funkcji celu, zmiennymi podczas analizy były również:

- wielkość turnieju wykorzystana w selekcji turniejowej (zbadany dla 2, 4, 6, 16),
- wymiar przestrzeni (zbadany dla n=1, 2, 3, 5)

5 Wyniki eksperymentów

Uwaga 1: Wykresy z zapisem eksperymentów znajdują się w folderze **graphs** w plikach **1 klaster.pdf**, **2 klastry.pdf**, **3 klastry.pdf**, **4 klastry.pdf** i **5 klastrów.pdf**
Uwaga 2: Wykresy zostały wygładzone dla lepszej interpretacji (za pomocą średniej kroczącej z wielkością kroku równą 6).
Uwaga 3: Dla każdego eksperymentu, zostały uruchomione 3 niezależne próby. Na wykresach przedstawione są wartości uśrednione z tych trzeb prób (by uniknąć artefaktów, wynikających z losowości).

Dla każdego badanego zestawu zmiennych przeprowadzono następujące eksperymenty:

- Porównanie wykresów dynamiki dla wszystkich funkcji dla 1d

- Porównanie wykresów dynamiki dla wszystkich funkcji dla 2d
- Porównanie wykresów dynamiki ze względu na wielkość turnieju:
  - Funkcja Gaussa,
  - Suma dwóch funkcji Gaussa,
  - Funkcja stała,
  - Funkcja Rastrigina
- Porównanie wykresów dynamiki ze względu na wymiar:
  - Funkcja Gaussa,
  - Suma dwóch funkcji Gaussa,
  - Funkcja stała,
  - Funkcja Rastrigina

Przy omawianiu wyników przyjęto następującą kolejność:

- obserwacje dla całej populacji (dla jednego klastra),
- obserwacje dla trzech klastrów,
- obserwacje dla innej liczby klastrów - jak liczba klastrów wpływa sposób prezentacji wyników?

W komentarzach umieszczono odwołania do wizualizacji populacji (w przypadku populacji jedno- lub dwuwymiarowej). Aby zobaczyć daną wizualizację należy przejść do folderu **populations**, następnie wybrać folder odpowiadający konkretnemu eksperymentowi, a następnie folder odpowiadający liczbie klastrów. Plik o nazwie **5.png** obrazuje stan populacji w piątej iteracji. Kolory, którymi są oznaczone punkty (oraz wykresy w dokumentach .pdf) mają następujące znaczenie:

- czerwony: pierwszy co do liczebności klaster,
- zielony: drugi co do liczebności klaster,
- zółty: trzeci co do liczebności klaster,
- fioletowy: czwarty co do liczebności klaster,
- czarny: piąty co do liczebności klaster.

Foldery dla eksperymentów zostały nazwane według schematu:
funkcja_parametryfunkcji_wymiar_rozmiarturnieju
Dla ułatwienia nawigacji - przy opisie obserwacji zostały podpowiedzi, których folderów dotyczą obserwacje (tzn. w których folerach należy szukać odpowiednich obrazów).
Uwaga: W raporcie suma dwóch funkcji Gaussa była nazywana skrótowo jako *Suma*.

5.1 Obserwacje i wnioski dla całej populacji

5.1.1 Porównanie wykresów dynamiki dla wszystkich funkcji dla 1d

Foldery, zawierające `dim_1_tournament_2`

- Dla funkcji Gaussa zbieżność można zaobserwować już po kilku iteracjach. Odchylenie standardowe zbiega do 1 (trzeba pamiętać, że jest to wartość graniczna, gdyż tyle wynosi odchylenie standardowe w mutacji),
- Dla Sumy zbieżność centrum populacji do maksimum jest osiągana wolniej niż dla funkcji Gaussa. Widać, że w jeszcze w ósmej iteracji znaczna część klastra znajduje się w okolicy maksimum lokalnego.
- Widać, że optymalizacja funkcji Rastrigina jest ciężkim zadaniem, populacja oddala się od maksimum,

- Dla funkcji stałej, populacja oddala się od początkowego położenia, przy tym wariancja populacji jest duża w porównaniu do innych funkcji - algorytm znajduje cały czas nowe położenia, jest stale nastawiony na eksplorację (powodem tego jest fakt, że nie ma okazji do eksploatacji). W populacji widać tzw. outliery, które są głównie efektem mutacji punktów, uprzednio wybranych w fazie selekcji.
- Przeciwnie - dla funkcji Rastrigina - wariancja jest mała, algorytm skupia się na eksploatacji. Dzieje się tak, ponieważ cały czas natrafia on na maksima lokalne.

5.1.2 Porównanie wykresów dynamiki dla wszystkich funkcji dla 2d

Foldery, zawierające `dim_2_tournament_2`

- Różnica między funkcją Gaussa, a Sumą jest jeszcze wyraźniejsza. W przypadku Sumy nie można zaobserwować zupełnej zbieżności do maksimum. Widoczny jest wpływ przyciągania maksimów lokalnych. Widać to także na wykresach.

5.1.3 Porównanie wykresów dynamiki ze względu na wielkość turnieju

Foldery, zawierające `dim_1`

- Dla funkcji Gaussa i Sumy widać polepszenie zbieżności wraz ze zwiększeniem wielkości turnieju. Interpretacja: im więcej elementów w turnieju, tym bardziej prawdopodobne, że trafi się wśród nich lepszy kandydat, który następnie wygra szranki, a następnie „pamięć" o nim pozostanie w populacji (przez „pamięć" rozumiana jest obecność tego osobnika bądź jego mutanta).
- Dla funkcji stałej widać większe „wędrowanie" przy zwiększaniu rozmiaru turnieju. W przypadku funkcji stałej, selekcja turniejowa degeneruje się do losowania jednego z p osobników, gdzie p to rozmiar turnieju. Populacja „wędruje" na skutek tego, że został wylosowany punkt z obrzeży populacji. W przypadku większego rozmiaru turnieju taki punkt może być szybciej wybrany niż w przypadku mniejszego rozmiaru turnieju.

5.1.4 Porównanie wykresów dynamiki ze względu na wymiar

Foldery, zawierające `tournament_2`

- Dla wszystkich funkcji, zarówno odległość od maksimum, jak i odchylenie standardowe wzrasta (nie jest to dziwne, jest to konsekwencja definicji tych miar).

5.2 Obserwacje i wnioski dla trzech klastrów

Wnioski w tej sekcji nie skupiają się na konkretnych eksperymentach, a mają charakter przeglądowy.

- Dla większości eksperymentów, proporcje liczebności klastrów nie zmieniają się. Dla eksperymentów, w których porównywane są wyniki dla wymiarów zadania, wielkości klastrów są mniej rozbieżne, klastry wykazują tendencję do równoliczności.
- dla jednowymiarowej funkcji Gaussa jest wyraźna różnica między środkiem klastra głównego, a środkami dwóch mniejszych klastrów. Dla dwuwymiarowej funkcji Gaussa ta tendencja zanika. Na wykresach widać z czego to wynika. W przypadku jednowymiarowym można wyznaczyć klastry: środkowy, na lewo od środkowego, na prawo od środkowego. W przypadku dwuwymiarowym, po ustabilizowaniu się algorytmu, klastry ustawiają się, zabierając mniej więcej po $\frac{1}{3}$ koła każdy.
- Dla jednowymiarowej funkcji Sumy, najmniejszy klaster po około czterdziestej iteracji nie stabilizuje swojej pozycji.
- Klastry w przypadku funkcji Rastrigina są spójne, jeśli chodzi o odległości od maksimum, dotyczy to zarówno przypadków jednowymiarowych, jak i wielowymiarowych.

5.3 Obserwacje i wnioski dla innej liczby klastrów

Wnioski w tej sekcji nie skupiają się na konkretnych eksperymentach, maja charakter przeglądowy.

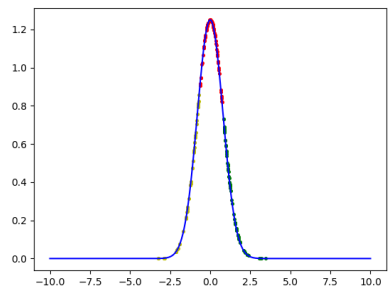
- Obserwacja o proporcjach liczebności klastrów zostaje podtrzymana. Można to wytłumaczyć zjawiskiem, zwanym „przekleństwem wielowymiarowości". Jedną z konsekwencji tego zjawiska jest to, że w wraz ze wzrostem wymiarowości, punkty (a także klastry) stają się do siebie coraz mniej podobne.
- (a propos obserwacji dla trzech klastrów, dotyczącej dwuwymiarowej liczby klastrów) Przy 5 klastrach wyniki są dobre, na wykresach można wtedy zaobserwować (oczywiście w przybliżeniu) klastry takie jak: centralny, dwa góra/dół, dwa lewo/prawo,
- Przy zwiększeniu wymiarowości problemu warto zastanowić się nad przeprowadzeniem analizy dla większej liczby klastrów

6 Podsumowanie i propozycje dalszych badań

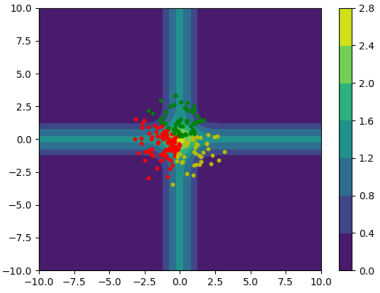
Autorzy dokonali analizy skupień w populacjach generowanych algorytmem ewolucyjnym. Dla funkcji Gaussa algorytm szybko radził sobie z osiąganiem zbieżności, a odchylenie standardowe w klastrach było niskie. Wyniki dla sumy dwóch funkcji Gaussa różniły się. Algorytm osiągał zbieżność po większej liczbie iteracji (problemy wystąpiły na przykład dla populacji jednowymiarowej). Zbieżność można było poprawić zwiększając rozmiar turnieju co wynika ze wzrostu prawdopodobieństwa wylosowania dobrego osobnika. Zupełnie inne zachowanie algorytmu zaobserwowano dla funkcji stałej, dla której populacja ciągle zmieniała położenie, a także powstawały elementy odstające będące efektem mutacji punktów wybranych w fazie selekcji. Zwiększanie rozmiaru turnieju tylko przyspieszało ten proces „wędrowania" po układzie współrzędnych. Przy funkcji Rastrigina populacja także ciągle zmieniała położenie, ale nie obserwowano wielu elementów odstających (tak, jak dla funkcji stałej), klastry poruszały się spójnie.

Autorzy zauważyli drobny problem, związany z tym, że klastrowanie według standardowej metryki (euklidesowa odległość między punktami) może zwracać nieintuicyjne wyniki. Wydaje się, że powinno być tak, że najliczniejszy (czerwony) klaster powinien skupiać się mniej więcej wokół optimum. Gdy algorytm już zbiegnie do celu to mutuje wokół optimum z rozkładem normalnym (w praktyce oznacza to, że przy centrum jest najwięcej elementów, a na obrzeżach mniej). Ciekawe wyniki przedstawione są na poniższych wykresach:

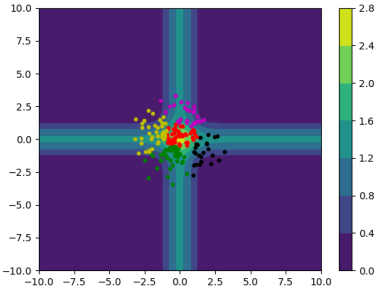
Na wykresie a) jest tak, jak możnaby się spodziewać. Na wykresach b) i c) - jest przedstawione klastrowanie w przypadku funkcji dwuwymiarowej dla tej samej iteracji. Widać, że w przypadku trzech klastrów - klastry są do siebie podobne i każdy z nich składa się mniej więcej z 1/3 koła, ponadto odległości optimum do punktu średniego w klastrach jest mniej więcej taka sama. Stąd na wykresach (w plikach .pdf) bierze się efekt, że wszystkie klastry są mniej więcej równo oddalone od optimum, co może sprawiać wrażenie, że mamy do czynienia z błędem (ale tak naprawdę chodzi jedynie o niefortunny w tym przypadku sposób przedstawienia danych). Użycie 5 klastrów poprawia sytuację - największy klaster jest w centrum - widać to na wykresie c). O ile sytuacja z wykresu c) jest dominująca, to nie jest tak zawsze, na wykresie d) jest przedstawione klastrowanie populacji 9 iteracji później.



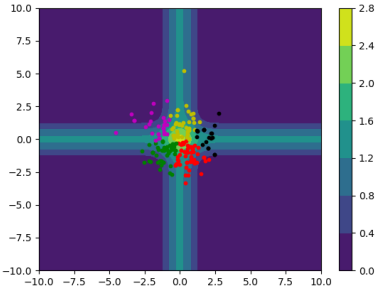
(a) funkcja jednowymiarowa



(b) funkcja dwuwymiarowa - 3 klastry



(c) funkcja dwuwymiarowa - 5 klastrów



(d) funkcja dwuwymiarowa - 5 klastrów, problem

Dalsze badania można rozpocząć od analizy skupień z wykorzystaniem innych metryk odległości stosowanych w algorytmie klastrowania. Taką metryką może być różnica w odległości punktów od minimum funkcji. Z pewnością pojawią się pewne problemy. Przykładowym problemem, przewidywanym przez autorów pracy jest sytuacja, w której po zbiegnięciu się algorytmu wszystkie klastry będą miały zerową średnią odległość od ekstremum, co nie pozwoli na rozróżnienie klastrów wedle tej wielkości.

7 Bibliografia

References

[1] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a data set via the gap statistic, 2001.

[2] Jacek Koronacki. *Statystyczne systemy uczące się*. Akademicka Oficyna Wydawnicza EXIT, 2005.