



Wrocław University
of Science and Technology

Algorytmy optymalizacji inspirowane naturą

Problem optymalizacyjny: Kolorowanie grafu
Algorytm metaheurystyczny: Algorytm genetyczny

Wydział Informatyki i Telekomunikacji
Kierunek: Informatyka Techniczna

Skład grupy projektowej:
Paweł SZYNAL 226026
Kamil ZDEB 235871

Prowadzący: Dr inż. Piotr Lechowicz

Kod grupy: K02-73b

Wrocław 2021 r.

Spis treści

1	Cel projektu	2
2	Wprowadzenie teoretyczne	2
2.1	Problem NP	2
2.2	Pojęcie grafu	3
2.3	Kolorowanie grafu	3
2.3.1	Kolorowanie wierzchołków	4
2.3.2	Sformułowanie problemu	6
2.4	Algorytm genetyczny	6
2.4.1	Pseudokod	7
2.4.2	Rozmnażanie	7
3	Przegląd literaturowy	9
3.1	Przegląd algorytmów inspirowanych naturą	9
3.2	Binarny algorytm nietoperza dla problemu kolorowania grafu	9
3.3	Wyszukiwanie sąsiedztwa w problemie kolorowania grafu	9
3.4	Algorytm genetyczny	10
3.5	Złożony algorytm genetyczny	10
4	Wybór technologii	11
4.1	Dlaczego Python?	11
4.2	Opis wykorzystywanych „frameworków”	11
5	Implementacja symulatora dla wybranego problemu optymalizacyjnego	11
6	Implementacja algorytmu referencyjnego	11
7	Implementacja algorytmu inspirowanego naturą dla wybranego problemu optymalizacyjnego	1
8	Badania	11
9	Podsumowanie	11

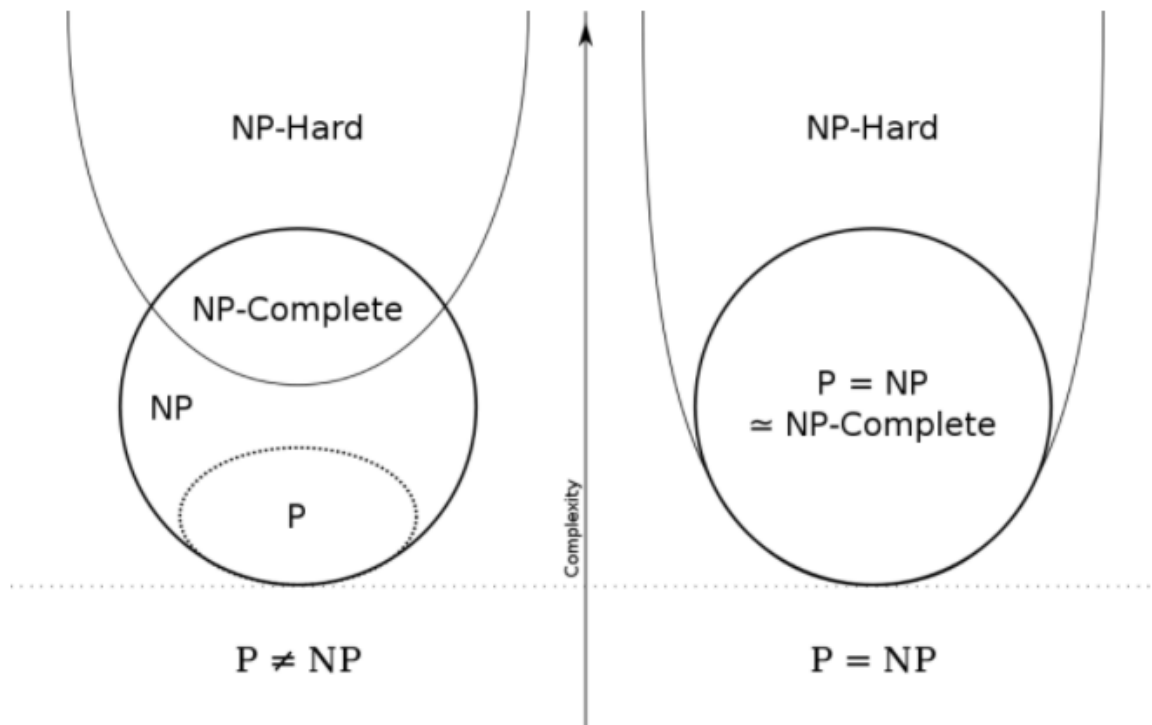
1 Cel projektu

Celem projektu jest zaprojektowanie i zaimplementowanie algorytmu optymalizacji inspirowanego naturą. Wybrany, przez nas problem optymalizacyjny jest kolorowanie grafu. W ramach projektu należy również przygotować symulator komputerowy dla danego problemu optymalizacyjnego, w którym jest możliwe przetestowanie zaimplementowanego algorytmu genetycznego. Po zaimplementowaniu algorytmu należy przeprowadzić testy skuteczności w odniesieniu do algorytmu referencyjnego.

2 Wprowadzenie teoretyczne

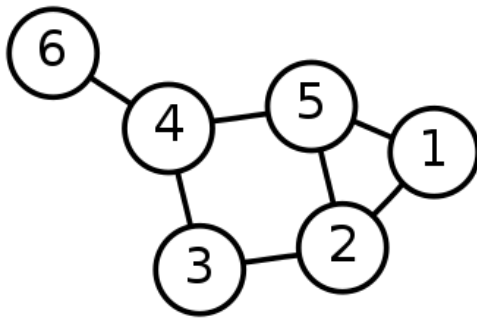
2.1 Problem NP

Problem NP (ang. nondeterministic polynomial) jest to problem decyzyjny, dla którego rozwiązanie można zweryfikować w czasie wielomianowym. Wykonanie czegoś w czasie wielomianowym oznacza, że czas potrzebny do otrzymania wyniku jest funkcją wielomianową wielkości problemu, czyli $O(n^x)$.

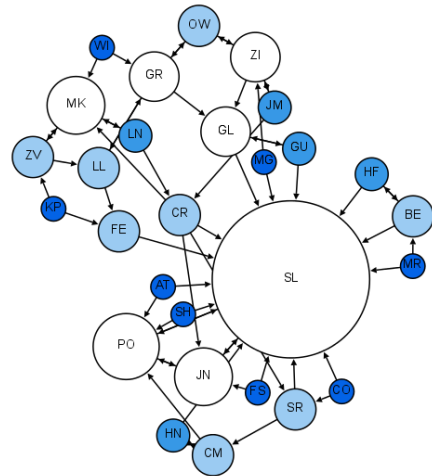


2.2 Pojęcie grafu

Grafem nazywamy jedną z podstawowych struktur danych używaną do modelowania wszelkiego rodzaju układów oraz sieci, które charakteryzują się skomplikowanymi zależnościami pomiędzy ich składnikami. Graf opisuje para elementów $G = (V, E)$, gdzie V oznacza zbiór wierzchołków grafu, zaś E – zbiór krawędzi. Rozróżnia się grafy nieskierowane, w których krawędzie łączą symetrycznie dwa wierzchołki, oraz grafy skierowane, w których krawędzie łączą dwa wierzchołki asymetrycznie. Grafy są jednym z głównych przedmiotów badań matematyki dyskretnej.



(a) Graf nieskierowany



(b) Graf skierowany na przykładzie „Teoria Grafów w socjologii”: Moreno Sociogram (1953).[17]

Rysunek 1: Przykład grafów skierowanych i nieskierowanych

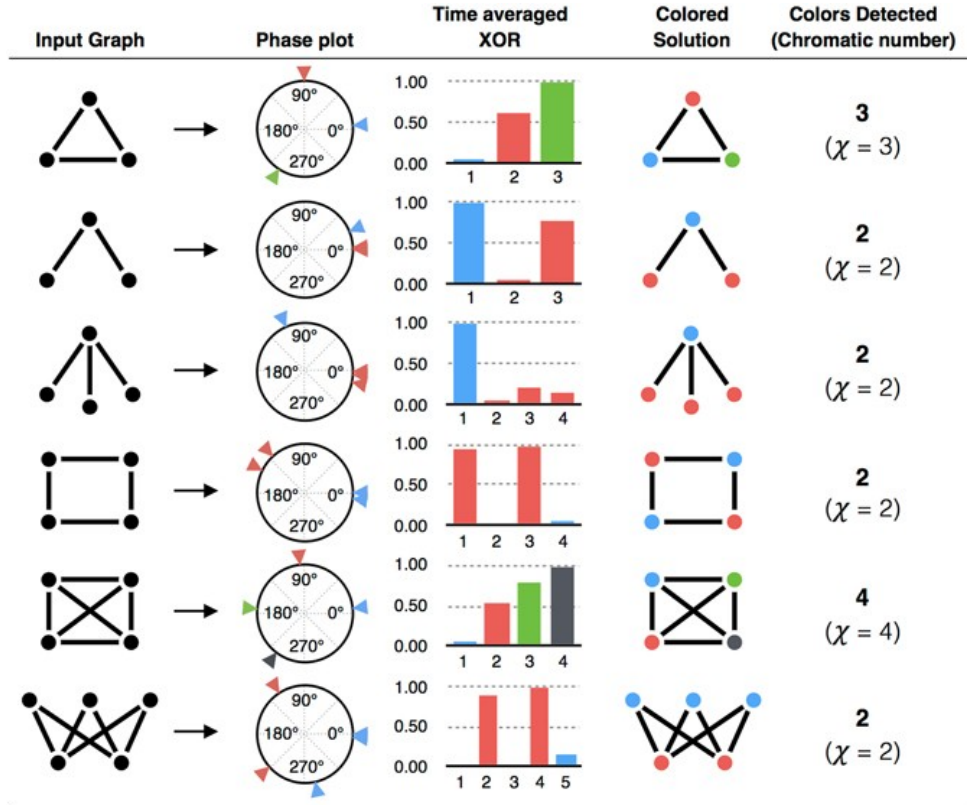
2.3 Kolorowanie grafu

W teorii grafów, kolorowanie grafów (GCP -Graph Coloring Problem) jest szczególnym przypadkiem etykietowania grafów. Intuicyjnie kolorowanie to przypisanie wierzchołkom grafu kolorów w taki sposób, by każde dwa sąsiednie wierzchołki miały różne kolory. W najprostszej formie jest to sposób na kolorowanie wierzchołków grafu w taki sposób, że żadne dwa sąsiadujące ze sobą wierzchołki nie są tego samego koloru. Podobnie, kolorowanie krawędzi przypisuje kolor do każdej krawędzi, tak że żadne dwie sąsiednie krawędzie nie są tego samego koloru. GCP to dobrze znany problem kombinatoryczny i jest wykorzystywany w rozwiązywaniu wielu realnych problemów, takich jak problem przydziału częstotliwości [26], harmonogramowanie załogi [27], alokacja rejestrów [28] itd.

2.3.1 Kolorowanie wierzchołków

Terminologia używania kolorów dla etykiet wierzchołków pochodzi z kolorowania mapy. Etykiety takie jak czerwony i niebieski są używane tylko wtedy, gdy liczba kolorów jest niewielka i zwykle rozumie się, że etykiety są rysowane z liczb całkowitych 1, 2, 3,

Def:Niech G będzie grafem prostym. Przez kolorowanie wierzchołków rozumiemy takie etykietowanie elementów $V(G)$ liczbami naturalnymi, że sąsiednie wierzchołki otrzymują różne liczby (kolory, etykiety).



Rysunek 2: Kolorowanie wierzchołków grafów <https://www.nature.com/articles/s41598-017-00825-1> [dostęp dnia 17.10.2021]

Kolorowanie grafu, które minimalizuje liczbę kolorów k , nazywa się minimalnym kolorowaniem grafu

Minimum k , dla którego możliwe jest prawidłowe pokolorowanie, nazywa się liczbą chromatyczną grafu.

Graf, który można pokolorować przy użyciu co najwyżej k kolorów, nazywany jest grafem k -częściowym. Podział k zbioru, podobnie jak zbiór węzłów, to grupowanie elementów zbioru w k grup. Przeformułujemy cel znalezienia przypisania koloru do znalezienia permutacji węzłów, tak aby te same kolorowe węzły pojawiały się razem. Nazywamy to przeformulowaniem jako sortowaniem kolorów wierzchołków, a odpowiednią wersję optymalną jako sortowaniem minimalnym kolorów wierzchołków. Obliczanie przypisania kolorów z sortowania kolorów to $O(n^2)$.

Def: Liczba chromatyczna grafu G jest to najmniejsza liczba k taka, że istnieje pokolorowanie G za pomocą k kolorów i jest oznaczana symbolem $\chi(G)$.

Problem kolorowania grafu znany jest jako NP-zupełny, co oznacza, że dotychczas nie powstał algorytm, który dla każdego zbioru danych znajdzie optymalne rozwiązanie w czasie wielomianowym. Zatem, w praktyce użyteczne są oszacowania.

Istnieje kilka dokładnych podejść do tego problemu, które opierają się na domniemanym wyliczeniu algorytmu [3] oraz algorytm branch-and-bound i jego warianty [4, 5]. Z drugiej strony konstruktywne podejścia zostały szeroko zaproponowane do rozwiązania kolorowania wykresu problem. Konstruktywne podejścia stopniowo budują rozwiązanie, w tej kategorii możemy przytoczyć opracowany algorytm Welsh i Powell [6], stopień saturacji (DSA TUR) [7] oraz rekurencyjny największy pierwszy algorytm (RLF) [8]. Co więcej, wiele rodzajów metaheurystyk i ich hybrydyzacje zostały użyte do rozwiązania GCP jak tabu Poszukiwanie Hertza i de Werra[9], symulowane wyżarzanie zaproponowane przez Johnsona i in.[10], algorytmy genetyczne [11], ant kolonia [12], przeszukiwanie ze zmiennym sąsiedztwem VNS [13], Zmienna Przestrzeń poszukiwań VSP, która jest rozszerzeniem VNS [14], a algorytm memetyczny autorstwa Zhipeng i Jin-Kao Hao [15], hybryda Algorytm sztucznej kolonii pszczół dla wykresu 3-kolorowanie[16], itp.

2.3.2 Sformułowanie problemu

Graf jest k -kolorowalny wtedy i tylko jeśli można go pokolorować za pomocą k kolorów. Formalnie k -kolorowanie będzie reprezentowane przez zbiór $S = \{C(v_1), C(v_2), \dots, C(v_n)\}$ taki jak $C(v_i)$ to kolor przypisany do wierzchołka v_i .

Jeśli dla wszystkich $\{u, v\} \in E$, $C(u) \neq C(v)$, to S jest prawdziwym k -kolorowaniem. W przeciwnym razie S jest niewykonalne k -barwienie. W optymalizacyjnej wersji grafu koloryzacji, głównym celem jest zminimalizowanie całkowitej liczby kolorów użytych do pokolorowania danego wykresu.

Biorąc pod uwagę k -kolorowanie $S = \{C(v_1), C(v_2), \dots, C(v_n)\}$, że wraz ze zbiorem $V = \{v_1, \dots, v_n\}$ wierzchołków, funkcja oceny f zlicza liczbę sprzecznych wierzchołków wytworzonych przez S w taki sposób, że:

$$f(S) = \sum \{u, v\} \in E^{\delta_{uv}}$$

Gdzie:

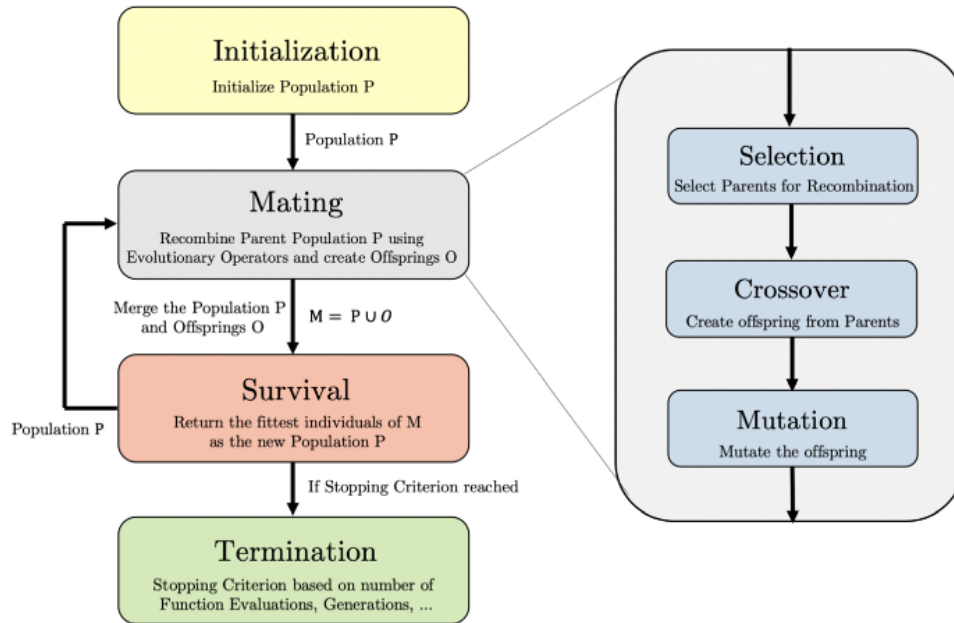
$$\delta_{uv} = \begin{cases} 1, & \text{if } C(u) = C(v) \\ 0 & \text{else} \end{cases}$$

W konsekwencji kolorowanie S z $f(S) = 0$ odpowiada możliwemu „ k -kolorowanie”.

2.4 Algorytm genetyczny

Algorytm genetyczny to taki algorytm, którego działanie jest zainspirowane selekcją naturalną gatunków. Należy do szerszej klasy algorytmów ewolucyjnych. Koncepcje oparte na ewolucji biologicznej są wykorzystywane do rozwiązywania wielu różnych problemów, a zwłaszcza szeroko znanych problemów optymalizacyjnych. Jest on wykorzystywany do wyboru cech w zbiorze danych czy rozpoznawania wzorców.

Algorytm genetyczny podczas swojego działania nie korzysta z jednego rozwiązania, ale z wielu, nazywanych populacją. Rozmiar populacji jest jednym z parametrów, które są ustalone przed rozpoczęciem wykonywania algorytmu. Każde z rozwiązań jest nazywane osobnikiem. Każda iteracja algorytmu genetycznego nazywana jest pokoleniem.



Rysunek 3: Schemat działania algorytmu genetycznego

Algorytm rozpoczyna swoje działanie od losowego wygenerowania populacji o predefiniowanym rozmiarze. Głównym celem inicjalizacji jest pokrycie jak największych połaci przestrzeni rozwiązań. Następnie następuje rozmnażanie, które dzieli się na trzy fazy. Najpierw następuje selekcja osobników do rozmnażania. Następnie ich krzyżowanie, a na końcu mutacje. Po zakończeniu rozmnażania do kolejnej iteracji wybieranych jest dalej tyle osobników, ile wynosi rozmiar populacji. Jeżeli osiągnięty został warunek stopu, to algorytm kończy swoje działanie, a jeżeli nie, to następuje kolejna iteracja.

Celem jest osiągnięcie jak najlepszej wartości funkcji celu. W naszym przypadku funkcją celu jest liczba wykorzystanych do pokolorowania grafu kolorów, a zatem celem jest minimalizacja funkcji celu.

2.4.1 Pseudokod

```

1  Zdefiniuj rozmiar_populacji, numer_pokolenia;
2  P = inicjalizuj(rozmiar_populacji)
3  ewaluacja(P)
4
5  for k in (1, ..., n_gen)
6      rodzice = selekcja(P)
7      O = mutacja(crossover(rodzice))
8      ewaluacja(O)
9      M = polacz(P, O)
10     P = selekcja(M)
11 endfor

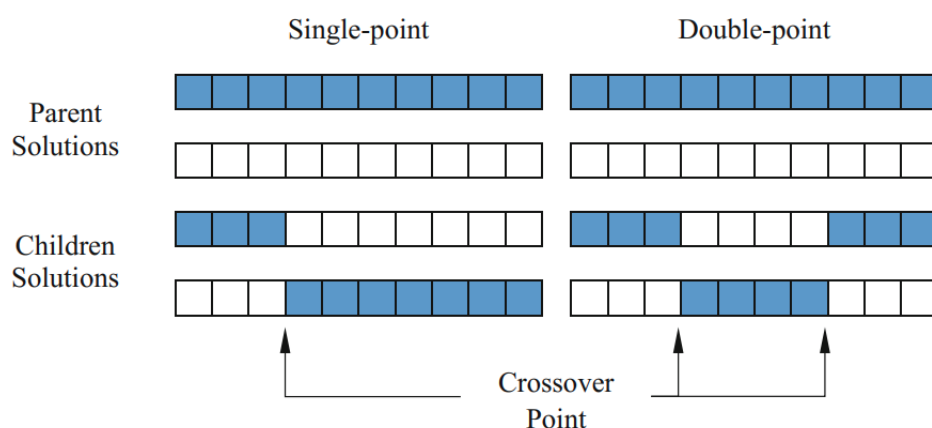
```

2.4.2 Rozmnażanie

Występuje wiele metod selekcji cech. Przykładowymi są:

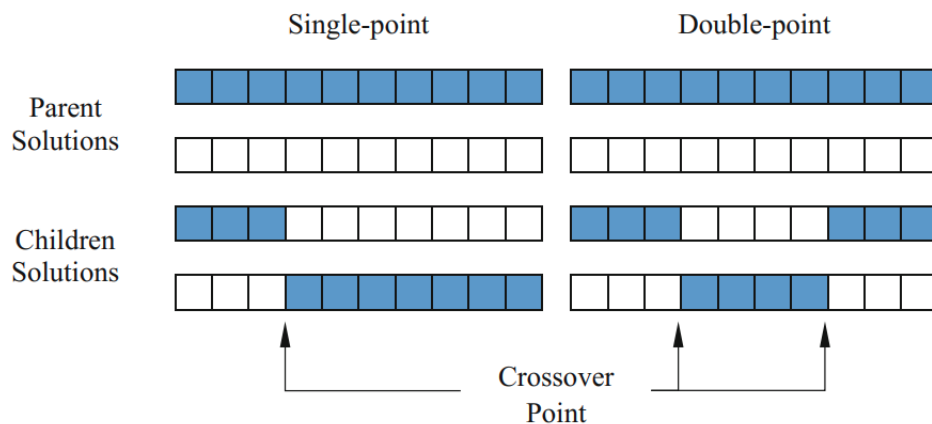
- Metoda ruletki - każdy z osobników posiada szansę na wzięcie udziału w rozmnażaniu proporcjonalną do swojej wartości funkcji celu.
- Metoda turniejowa - osobniki są sortowane na podstawie ich wartości funkcji celu, a następnie wybierana jest określona liczba najlepszych osobników

Krzyżowanie ze sobą organizmów również może przebiegać na kilka różnych sposobów. Najpopularniejszymi sposobami jest krzyżowanie jednopunktowe oraz krzyżowanie dwupunktowe. W przypadku tego pierwszego wyznaczany jest jeden punkt, a następnie chromosomy rodziców są w nim rozcinane, a następnie następuje zamiana części chromosomu znajdującej się za punktem przecięcia. W ten sposób uzyskujemy 2 nowe chromosomy. W krzyżowaniu dwupunktowym natomiast wyznaczamy dwa takie punkty oraz zamieniamy ze sobą części chromosomów znajdujących się między nimi.



Rysunek 4: Krzyżowanie jedno i dwupunktowe

Po krzyżowaniu następują ewentualne mutacje, w których każdy gen ma szansę na to, że wystąpi w nim zmiana. Pozwala to na zachowanie większej różnorodności w populacji. Pozwala to zmniejszyć szansę na osiągnięcie ekstremum lokalnego funkcji celu oraz pozostania w nim.



Rysunek 5: Przykład mutacji

3 Przegląd literaturowy

3.1 Przegląd algorytmów inspirowanych naturą

Tytuł publikacji: „A Review of Nature-Inspired Algorithms ”

Autorzy: Hongnian Zang, Shujun Zhang, Kevin Hapeshi (Department of Computing, the University of Gloucestershire, Cheltenham, GL50 2RH, UK)

Słowa kluczowe: Przegląd algorytmów optymalizacji bionicznej, Optymalizacja kolonii mrówek, Algorytm pszczół, Algorytm genetyczny, Firefly Algorytm

Streszczenie: Badanie bioniki łączy funkcje, struktury biologiczne i zasady organizacyjne występujące w przyrodzie z naszymi nowoczesnymi technologiami oraz liczne algorytmy matematyczne i metaheurystyczne zostały opracowane wraz z wiedzą przeniesienie procesu z form życia do ludzkich technologii. Wyniki badań bionicznych obejmują nie tylko produkty fizyczne, ale także różne metody obliczeniowe, które można zastosować w różnych obszarach. Ludzie nauczyli się od systemów biologicznych i struktury do projektowania i rozwijania szeregu różnych rodzajów algorytmów optymalizacyjnych, które były szeroko stosowane w obu studiach teoretycznych i zastosowania praktyczne. W niniejszym artykule przedstawia się szereg wybranych algorytmów inspirowanych naturą sprawdzone i przeanalizowane. Chociaż artykuł koncentruje się głównie na pierwotnej zasadzie stojącej za każdym z algorytmów, omówiono również ich zastosowania.

Konkluzja Algorytmy inspirowane naturą mogą zopstać połączone z innymi algorytmami, aby poprawić się, bądź być szybsze i wydajniejsze.

3.2 Binarny algorytm nietoperza dla problemu kolorowania grafu

Tytuł publikacji: „Binary Bat Algorithm for Graph Coloring Problem ”

Autorzy: Halima Djelloul, Sara Sabba, Salim Chikhi (Computer Science department MISC Laboratory Constantine 2 University Algeria)

Słowa kluczowe: algorytm nietoperza, algorytm inspirowany naturą, discrete bat algorithm, problem kolorowania grafów, heurystyka, optymalizacja dyskretna

3.3 Wyszukiwanie sąsiedztwa w problemie kolorowania grafu

Tytuł publikacji: „A variable neighborhood search for graph coloring ”

Autorzy: Cedric Avanthay, Alain Hertz, Nicolas Zufferey (Department of Mathematics, Swiss Federal Institute of Technology, Lausanne CH-1015, Switzerland)

Słowa kluczowe: Variable neighborhood search; Graph coloring

Streszczenie:

Metody zstępujące do optymalizacji kombinatorycznej polegają na wykonaniu sekwencji zmian lokalnych na rozwiązaniu początkowym, które każdorazowo poprawiają wartość funkcji celu aż do znalezienia optimum lokalnego. W artykule zaproponowano kilka metaheurystyk, które na różne sposoby rozszerzają ten schemat i pozwalają uniknąć uwięzienia w lokalnych optimach. Na przykład Hansen i Mladenovic zaproponowali metodę wyszukiwania

zmiennych sąsiedztwa, która nie została jeszcze zastosowana w wielu problemach optymalizacji kombinatorycznej. Celem artykułu jest zaproponowanie adaptacji tej nowej metody do problemu kolorowania grafów.

3.4 Algorytm genetyczny

Tytuł publikacji: „[Studies in Computational Intelligence Volume 780] Evolutionary Algorithms and Neural Networks || Genetic Algorithm”

Autorzy: Mirjalili, Seyedali

Słowa kluczowe: Creating Children Solutions; Masked Crossover; Main Search Mechanism; Worse Average Results; Half Uniform Crossover

Streszczenie: Algorytm genetyczny jest jednym z pierwszych w historii algorytmów stochastycznych bazujących na populacji. Podobnie do innych algorytmów ewolucyjnych głównymi operacjami algorytmu genetycznego są selekcja, krzyżowanie i mutacja. Ten rozdział zwięźle prezentuje ten algorytm i stosuje go do kilku studiów przypadku, aby zaobserwować jego wydajność.

Konkluzja: Ten rozdział zaprezentował algorytm genetyczny jako jeden z najbardziej popularnych w literaturze algorytmów ewolucyjnych. Po przedyskutowaniu głównych operacji ewolucyjnych, przeprowadzono kilka eksperymentów, aby zobaczyć wpływ zmiany parametrów na wydajność algorytmu. Zaobserwowano, że współczynnik mutacji powinien być dostosowywany ostrożnie, ponieważ potrafi znacząco obniżyć wydajność kiedy jego wartość jest wysoka. Zaobserwowano także, że współczynnik krzyżowania jest istotny w uzyskiwaniu dobrych rezultatów, ale jego wartość może być wysoka i nie przynosić negatywnych konsekwencji w przeciwieństwie do współczynnika mutacji. Do trenowania sieci neuronowych powinna zostać użyta ciągła wersja algorytmu genetycznego.

3.5 Złożony algorytm genetyczny

Tytuł publikacji: The compact genetic algorithm"

Autorzy: Harik, G.R.; Lobo, F.G.

Słowa kluczowe: Genetic algorithms; Algorithm design and analysis; Laboratories; Computer simulation; Genetic engineering; Probability distribution; Computational modeling; History; Convergence; Mathematical model

Streszczenie: Wprowadza złożony algorytm genetyczny (cGA), który reprezentuje populację opartą na rozkładzie prawdopodobieństwa zbioru rozwiązań i jest operacyjnie równoważny zachowaniu prostego algorytmu genetycznego z rozkładem jednostkowym. Złożony algorytm przetwarza niezależnie każdy gen i wymaga mniej pamięci niż prosty GA. Rozwój cGA jest kierowany przez właściwe zrozumienie parametrów algorytmu genetycznego i jego operacji. Tekst ilustruje mapowanie parametrów prostego GA na odpowiednik w złożonym algorytmie genetycznym. Porównuje działanie obu wersji algorytmu pod względem jakości i szybkości. Tekst porusza również to w jaki sposób użycie informacji w algorytmie genetycznym może doprowadzić do zaprojektowania lepszego algorytmu genetycznego.

Konkluzja: Pomimo tego, że złożony algorytm genetyczny w przybliżeniu naśladuje zachowanie prostego algorytmu, nie może być traktowany jako jego zastępstwo.

4 Wybór technologii

4.1 Dlaczego Python?

Python to język skryptowy ogólnego przeznaczenia, który ma przejrzystą i prostą składnię. Wraz z szybkim rozwojem dojrzałych, zaawansowanych i otwartych bibliotek i pakietów do obliczeń naukowych, Python stał się jednym z najpopularniejszych i najpotężniejszych języków obliczeń naukowych. Co więcej, Python działa na wielu platformach, co działa z różnymi systemami operacyjnymi i ma możliwość dostępu do bibliotek napisanych w różnych językach programowania i środowisk komputerowych. Python może również obsługiwać małe urządzenia, systemy wbudowane i mikrokontrolery. Ponadto Python wymaga bardzo minimalnej procedury konfiguracji na początek. Python wykorzystuje programowanie modułowe i obiektowe, które jest popularne metodologią organizowania klas, funkcji i procedur w hierarchiczne przestrzenie nazw.

4.2 Opis wykorzystywanych „frameworków”

5 Implementacja symulatora dla wybranego problemu optymalizacyjnego

6 Implementacja algorytmu referencyjnego

7 Implementacja algorytmu inspirowanego naturą dla wybranego problemu optymalizacyjnego

8 Badania

9 Podsumowanie

Literatura

- [1] Julian Blank. *An Introduction to Genetic Algorithms: The Concept of Biological Evolution in Optimization*. URL: <https://towardsdatascience.com/an-introduction-to-genetic-algorithms-the-concept-of-biological-evolution-in-optimization-fc96e78fa6db>.
- [2] Seyedali Mirjalili. “Genetic Algorithm”. W: *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham: Springer International Publishing, 2019, s. 43–55. ISBN: 978-3-319-93025-1. DOI: 10.1007/978-3-319-93025-1_4. URL: https://doi.org/10.1007/978-3-319-93025-1_4.