Wydział Informatyki
i Sztucznej Inteligencji

LAB4: Zastosowanie metod ewolucyjnych do rozwiązania zadania odwrotnego

dr inż. Konrad M. Gruszka,*

4 marca 2025

Streszczenie

Ten dokument stanowi instrukcję do wykonania ćwiczenia z zagadnienia wykorzystania metod ewolucyjnych do rozwiązania problemu odwrotnego znalezienia warunków brzegowych w problemie transferu ciepła w stacjonarnym przypadku dla materiału jednorodnego, do przedmiotu Rozwiązywanie Zadań Odwrotnych. Bazując na niniejszym opisie należy zaprojektować i zaimplementować algorytmy ewolucyjne w oparciu o kryteria przedstawione w dalszej części tego dokumentu.

1 Metody ewolucyjne

UWAGA! Razem z tym dokumentem dostarczono również materiały z którymi należy się zapoznać zanim zaczniecie Państwo realizować poniższe zadania!

Zadania do wykonania oraz do implementacji algorytmów w Pythonie

1. Zaimplementuj ewolucyjny algorytm dla przypadku jednowymiarowego, jednorodnego rozwiązania algorytmu wyprzedzającego ("do przodu"). Kod ma odtwarzać rozkład temperatury w materiale jednorodnym i porównywać otrzymany rozkład z rokładem otrzymanym na drodze symulacji **MRS**.
2. Zaimplementuj algorytm ewolucyjny 1D rozwiązujący problem odwrotny polegający na określeniu warunków brzegowych w jednorodnym materiale. W tym celu wygeneruj rozkład metodą MRS a następnie zaimplementuj następujące funkconalności:
   - kod ma umożliwić zmianę parametrów: **pop_size** - ilość osobników w jednej populacji, **num_generations** - maksymalna liczba generacji, **mutation_rate** - prawdopodobieństwo mutacji (0..1) oraz **crossover_rate** - prawdopodobieństwo krzyżowania (0..1)
   - po każdej generacji ma wypisywać jej numer oraz parametr "best fitness"
   - program ma umożliwiać zdefiniowanie wartości dla "best fitness"po kótrej można zakończyć generacje
   - na końcu program ma wyświetlić parametry: oryginalne warunki brzegowe, oszacowane z AG warunki brzegowe, bezwzględną różnicę miedzy tymi wartościami (błąd)
3. Zaimplementuj algorytm ewolucyjny rozwiązyujący problem odwrotny dla przypadku 2D (określenie warunków brzegowych w jednorodnym materiale 2D). Kod ma spełniać te same funkcjonalnoiści co kod 1D.
4. *Zadanie opcjonalne: Na podstawie algorytmu genetycznego z zadania 1 'do przodu' stwórz jego analogiczne rozwiązanie w 2D.
5. Testowanie i dokumentacja:

*Katedra Informatyki, Wydział Informatyki i Sztucznej Inteligencji (kgruszka@icis.pcz.pl)

- Przetestuj algorytm 1D dla różnych wartości ‘N‘, ‘T_0‘, ‘T_N‘, ‘max_iter‘, ‘tolerance‘, oraz różnej ilości generacji.
- Przetestuj algorytm 2D dla różnych wartości ‘N‘, ‘T_{up}‘, ‘T_{down}‘,‘T_{left}‘,‘T_{right}‘, ‘max_iter‘, ‘tolerance‘, oraz różnej ilości generacji.
- Dokumentujcie każdy test oraz wyniki.

---

2 Forma i ocena wykonania ćwiczenia

Forma oddania zadania
Zadanie należy oddać w formie pliku źródłowego (.ipynb) zawierającego implementację algorytmu oraz wygenerowane wykresy wraz z opisami
Należy przeprowadzić test dla następujących parametrów (1D):

- Liczba węzłów N: 50
- $T_0 = 150$
- $T_N = 50$
- $tolerancja = 0.01K$

Należy przeprowadzić test dla następujących parametrów (2D):

- Liczba węzłów Nx: 20, Ny:20
- $T_U = 300$, $T_D = 100$
- $T_L = 200$, $T_R = 0$
- $tolerancja = 0.01K$

Ponadto w raporcie (np .ipynb, .pdf) należy umieścić następujące informacje:

- Udokumentowanie przeprowadzonego testu dla parametrów podanych powyżej, wyniki oraz odpowiednie komentarze i wyjaśnienia.
- Uzyskane czasy trwania optymalizacji pojedynczej generacji
- Wykres przedstawiający rokład temperatur w pręcie.
- Wykres przedstawiający rokład temperatur w 2D.

Ocena ćwiczenia będzie bazować na:

1. Poprawności implementacji algorytmu.
2. Kompletności przeprowadzonych testów.
3. Jakości dokumentacji i wyjaśnień teoretycznych.
4. Efektywności i optymalizacji kodu.