

KI L^AT_EX DOKUMENT

Materiały do przedmiotu "Rozwiązywanie zadań odwrotnych"

Metody ewolucyjne - zastosowanie metod ewolucyjnych do rozwiązywania zadania odwrotnego

dr inż. Konrad M. Gruszka,*

Abstract. W tym dokumencie skupimy się na odtworzeniu warunków brzegowych z "empirycznego" rozkładu temperatury uzyskanego na drodze MRS w jednorodnym, jednowymiarowym i dwuwymiarowym materiale wykorzystując do tego celu algorytm genetyczny (AG). Końcowym efektem będzie skrypt w Python który realizuje postawione zadanie w najprostszy możliwy sposób.

1 Plan działania

1. Obliczamy docelowy rozkład temperatury metodą **MRS**, używając znanych, określonych na początku warunków brzegowych $T_{left}=120$ i $T_{right}=80$.
2. Tworzymy populację AG, gdzie każdy osobnik to potencjalna para warunków brzegowych $[T_{left}, T_{right}]$.
3. Ewolucja AG:
 - Funkcja przystosowania (fitness): Porównujemy temperatury uzyskane z AG z tymi z MRS (minimalizacja błędu).
 - Selekcja turniejowa: Wybieramy lepszych kandydatów.
 - Krzyżowanie i mutacja: Ewolucja w kierunku najlepszego rozwiązania.
4. Porównanie wyników:
 - Rysujemy docelowy i odtworzony rozkład temperatury.
 - Wypisujemy prawdziwe i oszacowane wartości T_{left} i T_{right} .
 - Obliczamy błąd bezwzględny.

2 Algorytm

1. **Rozwiązanie problemu bezpośredniego** – Metoda Różnic Skończonych (MRS)
Najpierw potrzebujemy sposobu na wygenerowanie rozkładu temperatury w pręcie dla znanych warunków brzegowych. Użyjemy metody różnic skończonych (MRS) w stanie ustalonym. Użyj funkcji *solve_temperature_homogeneous*(T_{left} , T_{right} , N), która

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

oblicza temperatury wzdłuż pręta, mając podane wartości T_{left} i T_{right} . Przyjmij, że materiał jest jednorodny, a wymiana ciepła odbywa się wzdłuż pręta zgodnie ze znanym równaniem Laplace'a:

$$T_i = \frac{T_{i-1} + T_{i+1}}{2}$$

Zastosujmy iteracyjną metodę aż do uzyskania stanu ustalonego (gdy wartości przestaną się zmieniać w istotny sposób).

2. **Algorytm genetyczny do znalezienia warunków brzegowych:** Teraz przechodzimy do części odwrotnej – znalezienia T_{left} i T_{right} , które wygenerowały dany rozkład temperatury.

- (a) **Chromosom:** Każdy osobnik w populacji to para liczb $[T_{left} \text{ i } T_{right}]$.

Populacja: Składa się z wielu takich par o losowych wartościach początkowych.

Zadanie: zaimplementuj funkcję *initialize_population(pop_size)*, która zwróci tablicę losowych par $[T_{left} \text{ i } T_{right}]$ w zadanym zakresie temperatur.

- (b) **Funkcja przystosowania (fitness):**

Każdy osobnik posiada własny zestaw warunków brzegowych. Aby ocenić jego jakość, symulujemy rozkład temperatury za pomocą MRS i porównujemy go z rozkładem docelowym. Fitness powinno być miarą błędu, np. sumą kwadratów różnic pomiędzy wartościami węzłów:

$$fitness = \sum (T_{pred} - T_{target})^2$$

Zadanie: Zaimplementuj funkcję *fitness(individual)*, która obliczy błąd między rozkładem temperatury z AG a rzeczywistym.

- (c) **Selekcja turniejowa:** Wybieramy dwa losowe osobniki z populacji. Osobnik z lepszą wartością fitness przechodzi do kolejnej generacji.

Zadanie: Napisz funkcję *tournament_selection(population)*, która tworzy nową populację wyselekcjonowanych osobników.

- (d) **Krzyżowanie jednopunktowe:** Wymieniamy jedną wartość między dwoma osobnikami. Wybieramy losowo, czy zamieniamy T_{left} , czy T_{right} .

Zadanie: Zaimplementuj funkcję *crossover(parent1, parent2)*, która tworzy dwójkę nowych osobników.

- (e) **Mutacja:** Losowa zmiana wartości T_{left} lub T_{right} z małą perturbacją.

Ograniczamy wartości do sensownego zakresu.

Zadanie: Zaimplementuj funkcję *mutate(individual)*, która zmienia jedną z temperatur w losowy sposób.

- (f) **Główna pętla algorytmu:** Po zaimplementowaniu powyższych kroków, musisz:

- Inicjalizować populację.
- Iteracyjnie ewoluować populację przez określoną liczbę pokoleń:
 - Wybierać najlepsze osobniki (selekcja).
 - Tworzyć nowe osobniki (krzyżowanie).

- Mutować nowe osobniki.
 - Rejestrować najlepsze rozwiązanie w każdym pokoleniu.
 - Na końcu wypisać wyniki i porównać z rzeczywistymi warunkami brzegowymi.
- (g) **Analiza wyników:** Porównanie rozkładów temperatury:
- Narysuj rozkład prawdziwy (otrzymany z MRS).
 - Rozkład znaleziony przez AG.
- Policz i wyświetl bezwzględne różnice między prawdziwymi a odtworzonymi wartościami.

3 Algorytm w 2D

Rozbudowa algorytmu pracującego w przestrzeni jednowymiarowej do przestrzeni dwuwymiarowej jest dość prosta. Zamiast par osobników $[T_{left}$ i $T_{right}]$ opisujących warunki brzegowe na dwóch końcach pręta 1D, wystarczy rozszerzyć pary na $[T_{left}, T_{right}, T_{top}, T_{bottom}]$ i postępować wg poniższego planu:

1. Generowanie temperatury w siatce 2D (MRS)
 - Warunki brzegowe: $T_{top}, T_{bottom}, T_{left}, T_{right}$
 - Rozwiązanie równania Laplace'a metodą różnic skończonych.
2. Implementacja algorytmu genetycznego
 - Osobnik: Zestaw $[T_{top}, T_{bottom}, T_{left}, T_{right}]$
 - Funkcja przystosowania: Różnica między siatką temperatur z AG a rzeczywistą.
 - Selekcja turniejowa.
 - Krzyżowanie i mutacja.
3. Porównanie wyników
 - Wizualizacja siatek temperatur.
 - Analiza błędów rekonstrukcji.

4 Podsumowanie

Aby zrozumieć działanie algorytmu genetycznego w kontekście rozkładu temperatury, należy zrozumieć, jak algorytm manipuluje populacją rozwiązań (temperatur) przez selekcję, krzyżowanie i mutację, aby znaleźć najlepszy możliwy rozkład temperatury, który minimalizuje różnicę między obliczonym i docelowym rozkładem temperatury. Parametry algorytmu, takie jak rozmiar populacji, liczba pokoleń, współczynniki krzyżowania i mutacji, mają kluczowy wpływ na efektywność tego procesu.