

 <p>Politechnika Wroclawska</p>	<p>Dr. inż. Ewa Szlachcic Katedra Automatyki, Mechatroniki i Systemów Sterowania</p>
<p>Wydział Elektroniki Kier: Automatyka i Robotyka Studia magisterskie II stopnia</p>	<p>Teoria i metody optymalizacji AREU0003 Projekt</p>
<p>Zadania programowania nieliniowego PN – algorytmy optymalizacji nieliniowej dla zmiennych ciągłych</p>	

1. Opis projektu z przedmiotu “ Teoria i metody optymalizacji” w zakresie optymalizacji nieliniowej dla zmiennych ciągłych powinien zawierać:

- sformułowanie zadania optymalizacji: należy sformułować postać zadania z uwzględnieniem przyjętych ograniczeń oraz podać przyjętą metodę rozwiązania,
- szczegółowe omówienie algorytmu optymalizacji:
 - typ algorytmu – algorytm optymalizacji lokalnej lub algorytm optymalizacji globalnej,
 - stosowane kryteria stopu - zbieżność algorytmu,
 - ograniczenia algorytmu (np.: metoda min. w kierunku - uzasadnienie wyboru metody, metoda min. bez ograniczeń, metoda min. z ograniczeniami),
 - w algorytmie heurystycznym szczegółowe omówienie operatorów – uzasadnienie wyboru
- informacje ogólne o programie – środowisko programistyczne, zastosowane biblioteki lub gotowe moduły,
- zasady wprowadzania danych początkowych – reguły doboru współczynników algorytmu – jeżeli takie są ,
- przykłady testowe ze szczegółowym opisem techniki rozwiązania zadania.

Zadania programowania nieliniowego (PN):

Wybór zadań testowych dla zagadnień z dziedziny programowania nieliniowego:

- **Trzy zadania testowe konieczne do załączenia do opisu projektu** można wybrać spośród zadań optymalizacji funkcji nieliniowych podanych w pliku funkcji testowych. Przykłady muszą dotyczyć funkcji posiadających co najmniej kilka minimów lokalnych. Wszystkie minima lokalne muszą zostać obliczone.
 - **Czwarte nieliniowe zadanie testowe powinno posiadać interpretację praktyczną, z odniesieniem do literatury (na ocenę bardzo dobry lub wyższą).**
- dyskusja wpływu doboru punktu startowego na szybkość działania algorytmu – jeżeli taki punkt jest wykorzystany w algorytmie,
 - ilustracja wyników pracy algorytmu optymalizacji:
 1. ilustracja kolejnych iteracji algorytmu z wyszczególnieniem istotnych dla iteracji danych (jeżeli jest to możliwe)
 2. rozwiązanie optymalne z wartością funkcji celu.
 3. dla zadania o wymiarze $n=2$ warstwie funkcji celu oraz punkt optymalny z jego wartością funkcji celu (**na ocenę bardzo dobry lub wyższą**),
 4. dla algorytmów optymalizacji lokalnej i $n=2$: droga dojścia do rozwiązania optymalnego (jeżeli jest możliwa) – aproksymacja odcinkami, wprowadzenie możliwości skalowania ekranu, wizualizacja 2D lub 3D (muszą być widoczne kolejne kroki algorytmu, natomiast wybór rozmiaru przestrzeni do wizualizacji powinien być podyktowany czytelnością rysunku z punktu widzenia kolejnych kroków algorytmu optymalizacji) - (**na ocenę bardzo dobry lub wyższą**).
 5. dla algorytmów optymalizacji globalnej – wizualizacja procesu obliczeń odpowiadająca określonemu algorytmowi (zachowanie się operatorów wg ustaleń szczegółowych w projekcie) – **na ocenę dobry plus lub wyższą**.

6. Podać ciekawe przykłady, dla różnych typów warstwic - dla $n=2$ wąskie doliny czy zagęszczone warstwice/ oraz wyjaśnić różnice w pracy algorytmu dla tych przykładów.
7. Zrobić rysunki warstwic wraz z trajektorią dojścia od punktu początkowego do punktu optymalnego lub z najlepszym osobnikiem w populacji wraz z zaznaczonym obszarem dopuszczalnym dla omawianych przykładów (na ocenę bardzo dobry lub wyższą).

■ wykaz literatury, zawierający pozycje cytowane w opisie.

2. Należy przyjąć następujące oznaczenia:

n - ilość zmiennych, m - ilość ograniczeń, x – wektor zmiennych decyzyjnych, zbiór rozwiązań dopuszczalnych : X , dokładność obliczeń ϵ , liczba iteracji L , punkt optymalny x^* , wartość optymalna funkcji celu: $f(x^*) = f^*$.

-dobór parametrów metody /uzasadnić, czy parametry zostały przetestowane i dla jakich zadań testowych podać przykłady (wykazać prawidłowość doboru) wskazać literaturę/.

3. Należy umieścić dyskusję błędów wraz z uzasadnieniem przyczyn ich powstawania z punktu widzenia techniki optymalizacji. Omówić wprowadzone zabezpieczenia.

4. Program powinien zawierać pakiet organizujący przebieg procesu optymalizacji – program powinien pozwalać na wielokrotną minimalizację tej samej funkcji przy zmieniających się ograniczeniach przy różnych danych wejściowych bez uciekania się do współpracy z systemem operacyjnym.