

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Metody optymalizacji		Kod 1010515321010510493
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 1 / 2
Ścieżka obieralności/specjalność Technologie wytwarzania oprogramowania	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stopień studiów: II stopień	Forma studiów (stacjonarna/niestacjonarna) niestacjonarna	
Godziny Wykłady: 16 Ćwiczenia: - Laboratoria: 16 Projekty/seminaria: -		Liczba punktów 4
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) (ogólnouczelniany, z innego kierunku) kierunkowy z danego kierunku		
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki nauki techniczne		Podział ECTS (liczba i %) 4 100%
Odpowiedzialny za przedmiot / wykładowca: dr hab. inż. Maciej Komosiński email: maciej.komosiński@put.poznan.pl tel. 61 6652931 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z algebry liniowej i geometrii (proste operacje na wektorach i macierzach) i analizy matematycznej (różniczkowanie prostych funkcji).
2	Umiejętności:	Powinien także posiadać umiejętność projektowania, implementowania i testowania prostych programów komputerowych (w dowolnym języku) realizujących podstawowe operacje wektorowo-macierzowe.
3	Kompetencje społeczne	Z kolei w zakresie kompetencji społecznych do pożądanych cech należą: ciekawość poznawcza i wytrwałość w dążeniu do poszerzania swojej wiedzy, a także spora doza uczciwości i kultury osobistej.
Cel przedmiotu: 1. Przekazanie studentom szczegółowej wiedzy dotyczącej: a) zagadnienia optymalizacji i celów optymalizacji, b) optymalizacji ciągłej z ograniczeniami oraz optymalizacji kombinatorycznej, c) wybranych metod rozwiązywania problemów optymalizacji, przede wszystkim przybliżonych metod iteracyjnych, metod heurystycznych opartych na przeszukiwaniu sąsiedztwa oraz metaheurystyk. 2. Rozwijanie u studentów umiejętności a) identyfikowania i formułowania zadań optymalizacyjnych, b) tworzenia i testowania programów implementujących omawiane metody rozwiązywania przedstawionych problemów.		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza: 1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów i złożoności oraz języków programowania w dziedzinie sztucznej inteligencji i analizie danych - [K_W4] 2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki, takimi jak: formułowanie i rozwiązywanie (dokładne oraz przybliżone) zadań optymalizacji ciągłej oraz kombinatorycznej, z uwzględnieniem zastosowań w dziedzinie sztucznej inteligencji i analizie danych - [K_W5] 3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce, w szczególności w zakresie optymalizacji, oraz w wybranych pokrewnych dyscyplinach naukowych, wykorzystujących mechanizmy optymalizacji - [K_W6] 4. ma podstawową wiedzę o cyklu życia programowych systemów informatycznych implementujących algorytmy optymalizacji - [K_W7] 5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z obszaru optymalizacji - [K_W8]		

Umiejętności:
1. potrafi pozyskiwać informacje z literatury oraz innych źródeł w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie - [K_U1] 2. potrafi określić kierunki dalszego uczenia się i zrealizować proces samokształcenia - [K_U5] 3. potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne - [K_U9] 4. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K_U10] 5. potrafi formułować i testować hipotezy związane z problemami inżynierskimi i prostymi problemami badawczymi - [K_U12] 6. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]
Kompetencje społeczne:
1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1] 2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życie - [K_K4] 3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]

Sposoby sprawdzenia efektów kształcenia
Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób: Ocena formująca: a) w zakresie wykładów: - na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach b) w zakresie laboratoriów: - na podstawie oceny bieżącego postępu realizacji zadań Ocena podsumowująca: a) w zakresie wykładów weryfikowanie założonych efektów kształcenia realizowane jest przez: - ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym w formie kilkunastu krótkich zadań. b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez: - ocenianie ciągle, na każdych zajęciach (odpowiedzi ustne) ? ocena przyrostu umiejętności posługiwania się poznanymi zasadami i metodami - ocenę wiedzy i umiejętności poprzez 2 kartkówki w semestrze, - weryfikację działania kodu źródłowego, - ocenę samodzielnie wykonanej prezentacji. Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za: - omówienia dodatkowych aspektów zagadnienia, - uwagi związane z udoskonaleniem materiałów dydaktycznych,
Treści programowe
Program wykładu obejmuje następujące zagadnienia: Wprowadzenie do optymalizacji, złożoność problemów optymalizacyjnych. Idea optymalizacji, problem optymalizacji matematycznej i jego składowe (funkcja celu i ograniczenia), minima lokalne i globalne, problemy łatwe i trudne, ograniczenia. Programowanie matematyczne i jego warianty. Metoda Branch and Bound. Programowanie dynamiczne. Algorytmy lokalnej optymalizacji. Algorytm Symulowanego Wyżarzania i jego parametryzacja. Sposoby ustalania temperatury początkowej oraz schematu chłodzenia i ich wpływ na działanie algorytmu. Algorytm przeszukiwania Tabu oraz techniki zwiększające jego efektywność. Typowe problemy występujące podczas optymalizacji metodami metaheurystycznymi. Algorytm genetyczny i ewolucyjny oraz jego elementy składowe. Techniki selekcji, krzyżowania, mutacji, skalowania ocen rozwiązań. Metody hybrydowe w optymalizacji. Optymalizacja wielokryterialna. Zajęcia laboratoryjne prowadzone są w formie czterech 4-godzinnych ćwiczeń, odbywających się w laboratorium komputerowym. Ćwiczenia realizowane są indywidualnie. Program laboratorium obejmuje następujące zagadnienia: Modelowanie (zapisywanie) matematyczne zadań z treścią oraz rozwiązywanie prostych zadań metodą graficzną. Rozwiązywanie zadań programowania matematycznego (PM) za pomocą dostępnych programów, "solverów". Implementacja losowego przeszukiwania oraz local search dla problemu TSP; pomiar czasu działania algorytmu. Indywidualne prezentacje: wybrany problem praktyczny i jego optymalizacja poznanymi metodami. Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta. Metody dydaktyczne: 1. wykład: prezentacja multimedialna, ilustrowana przykładami i zadaniami 2. ćwiczenia laboratoryjne: rozwiązywanie zadań i implementowanie rozwiązań, wykonywanie eksperymentów, dyskusja rezultatów i wyników

Literatura podstawowa:		
1. W. Grabowski: Programowanie matematyczne, PWE, Warszawa, 1980		
2. R. Wit: Metody programowania nieliniowego, WNT, Warszawa, 1986		
3. W. Findeisen, J. Szymanowski, A. Wierzbiński: Teoria i metody obliczeniowe optymalizacji, PWN, Warszawa, 1977		
Literatura uzupełniająca:		
1. S. Boyd, L. Vandenberghe: Convex Optimization, Cambridge University Press, 2004		
2. D. Bertsekas: Nonlinear Programming, Athena Scientific, Nashua, USA, 1999		
3. Z. Jędrzejczak, J. Skrzypek, K. Kukuła, A. Walkost: Badania operacyjne w przykładach i zadaniach, PWN, Warszawa, 2002		
Bilans nakładu pracy przeciętnego studenta		
Czynność		Czas (godz.)
1. udział w zajęciach laboratoryjnych / ćwiczeniach		16
2. udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / projektu (częściowo mogą być realizowane drogą elektroniczną)		3 24
3. przygotowanie prezentacji i stworzenie oprogramowania (czas poza zajęciami laboratoryjnymi)		20
4. przygotowanie do sprawdzianu		16
5. udział w wykładach		20
6. przygotowanie do egzaminu i obecność na egzaminie (2h), omówienie wyników egz. (1h)		
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	99	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	38	1
Zajęcia o charakterze praktycznym	40	2