

Politechnika Białostocka									
Kierunek studiów	Informatyka						Poziom i forma studiów	drugiego stopnia stacjonarne	
Specjalność / Ścieżka dyplomowania	Inżynieria Oprogramowania						Profil kształcenia	ogólnoakademicki	
Nazwa przedmiotu	Metody optymalizacji i programowanie liniowe						Kod przedmiotu	INF2MOP	
							Rodzaj przedmiotu	obieralny	
Forma zajęć i liczba godzin	W	Ć	L	P	Ps	T	S	Semestr	2
	15				30			Punkty ECTS	3
Przedmioty wprowadzające									
Cele przedmiotu	Celem przedmiotu jest zapoznanie studentów z różnymi metodami i algorytmami optymalizacji zadań. Student nabywa umiejętność wyboru odpowiedniej metody do różnych typów zadań optymalizacyjnych. Potrafi zastosować nowoczesne metody i algorytmy optymalizacji w różnych dziedzinach informatyki. Potrafi zaimplementować wybrane algorytmy, porównać je i ocenić ich efektywność.								
Treści programowe	<p>Wykład:</p> <ol style="list-style-type: none"> <li>1. Wprowadzenie do formułowania zadań optymalizacji.</li> <li>2. Metoda eliminacji Gaussa.</li> <li>3. Programowanie liniowe - metoda graficzna, metoda sympleks, metody sztucznej bazy, zrewidowana metoda sympleks, zagadnienia dualne, ekonomiczna interpretacja zagadnień dualnych.</li> <li>4. Algorytmy punktu wewnętrznego.</li> <li>5. Zagadnienia transportowe - metody wyznaczania dopuszczalnego rozwiązania bazowego, metoda potencjałów.</li> <li>6. Bezgradientowe metody optymalizacji - szacowanie przedziału poszukiwań, szukanie minimum w przedziale, metody optymalizacji wielowymiarowej.</li> <li>7. Gradientowe metody optymalizacji - metody dla zadań nieliniowych bez ograniczeń, metody dla zadań nieliniowych z ograniczeniami.</li> <li>8. Metody niedeterministyczne.</li> <li>9. Optymalizacja wielokryterialna.</li> <li>10. Podejście Pareto.</li> <li>11. Redukcja problemów wielokryterialnych.</li> </ol> <p>Pracownia specjalistyczna:</p> <ol style="list-style-type: none"> <li>1. Wprowadzenie do środowiska Matlab.</li> <li>2. Metoda eliminacji Gaussa.</li> <li>3. Programowanie liniowe - metoda sympleks.</li> <li>4. Zagadnienia transportowe - metody wyznaczania dopuszczalnego rozwiązania bazowego, metoda potencjałów, implementacja algorytmu szukania cykli.</li> <li>5. Algorytmy szacowania przedziału poszukiwań rozwiązania optymalnego.</li> <li>6. Metody bezgradientowe.</li> <li>7. Metody gradientowe.</li> <li>8. Algorytmy zadań z ograniczeniami i bez ograniczeń.</li> <li>9. Optymalizacja wielokryterialna.</li> </ol>								
Metody dydaktyczne	programowanie z użyciem komputera, wykład problemowy,								
Forma zaliczenia	<p>Wykład - zaliczenie pisemne.</p> <p>Pracownia specjalistyczna - wybór jednej z dwóch ścieżek:</p> <p>1) ścieżka standardowa - w ciągu semestru student oddaje raporty z sześciu jednostek tematycznych (łącznie 60%) i wykonuje jeden duży projekt programistyczny (40%). Do wykonania zadań wykorzystywane jest środowisko Matlab oraz własne implementacje wybranych algorytmów (może być dowolny język programowania). Student musi zdobyć min. 50% punktów, by uzyskać zaliczenie przedmiotu.</p> <p>2) ścieżka standardowo-projektowa - w ciągu semestru student oprócz wybranych raportów z jednostek tematycznych przygotowuje projekt - może to być projekt programistyczny z metod optymalizacji lub projekt z zastosowania optymalizacji w rzeczywistym problemie. Rozliczenia z projektu - co 2 tygodnie pokazywanie kolejnych dokonań, raport końcowy.</p>								
Symbol efektu uczenia się	Zakładane efekty uczenia się						Odniesienie do kierunkowych efektów uczenia się		
EU1	ma wiedzę na temat zadań optymalizacji, metod i algorytmów optymalizacji						INF2_W01		
EU2	potrafi sformułować zadanie optymalizacji w postaci matematycznej						INF2_U01		
EU3	potrafi przetestować, wybrać i użyć odpowiedniej metody optymalizacji do uzyskania rozwiązania						INF2_W01 INF2_W04 INF2_U06		
EU4	potrafi przygotować raport zawierający omówienie wyników realizacji zadania inżynierskiego						INF2_U02		
Symbol efektu uczenia się	Sposób weryfikacji efektu uczenia się						Forma zajęć na której zachodzi weryfikacja		
EU1	sprawdzian z wykładu						W		
EU2	ocena raportów z pracowni						Ps		
EU3	sprawdzian z wykładu, ocena raportów z pracowni						W, Ps		
EU4	ocena raportów z pracowni						Ps		
Bilans nakładu pracy studenta (w godzinach)								Liczba godz.	
Wyliczenie	1 - Udział w wykładach - 15x1h						15		
	2 - Udział w pracowni specjalistycznej - 15x2h						30		
	3 - Projekt, implementacja, testowanie aplikacji -						20		
	4 - Udział w konsultacjach -						5		
	5 - Przygotowanie do zaliczenia -						5		
<b>RAZEM:</b>								<b>75</b>	
Wskaźniki ilościowe								GODZINY	ECTS
Nakład pracy studenta związany z zajęciami wymagającymi bezpośredniego udziału nauczyciela								50 (4)+(1)+(2)	2,0
Nakład pracy studenta związany z zajęciami o charakterze praktycznym								50 (2)+(3)	2,0

<b>Literatura podstawowa</b>	<ol style="list-style-type: none"> <li>1. K. Amborski, Podstawy metod optymalizacji, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2009.</li> <li>2. D. P. Bertsekas, Nonlinear programming, Athena Scientific, Belmont, 1999.</li> <li>3. W. Findeisen, J. Szymanowski, A. Wierzbicki, Teoria i metody obliczeniowe optymalizacji, PWN 1980.</li> <li>4. S. I. Gass, Programowanie liniowe. Metody i zastosowania, PWN, Warszawa, 1976.</li> <li>5. A. Ostanin, Metody i algorytmy optymalizacji, Wydawnictwo Politechniki Białostockiej, Białystok, 2003.</li> <li>6. A. Stachurski, A. P. Wierzbicki, Podstawy optymalizacji, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2001.</li> </ol>	
<b>Literatura uzupełniająca</b>	<ol style="list-style-type: none"> <li>1. A. Ostanin, Optymalizacja liniowa i nieliniowa, Wydawnictwo Politechniki Białostockiej, Białystok, 2005.</li> <li>2. M. Ferris, O. Mangasarian, S. Wright S, Linear Programming with MATLAB, MPS-SIAM, Philadelphia, 2007.</li> <li>3. P. Venkataraman, Applied Optimization with MATLAB Programming, Wiley &amp; Sons, NY, 2002.</li> <li>4. J. Seidler, A. Badach, W. Molisz, Metody rozwiązywania zadań optymalizacji, WNT, Warszawa, 1980.</li> <li>5. A. Ruszczyński, Nonlinear optimization, Princeton University Press, 2006.</li> </ol>	
<b>Jednostka realizująca</b>	Katedra Oprogramowania	Data opracowania programu
<b>Program opracował(a)</b>	dr inż. Magdalena Topczewska	22 maja 2020