

KARTA OPISU MODUŁU KSZTAŁCENIA		
Nazwa modułu/przedmiotu Optymalizacja kombinatoryczna		Kod 1010511331010510332
Kierunek studiów Informatyka	Profil kształcenia (ogólnoakademicki, praktyczny) ogólnoakademicki	Rok / Semestr 2 / 3
Ścieżka obieralności/specjalność -	Przedmiot oferowany w języku: polski	Kurs (obligatoryjny/obieralny) obligatoryjny
Stoień studiów: I stopień	Forma studiów (stacjonarna/niestacjonarna) stacjonarna	
Godziny Wykłady: 30 Ćwiczenia: - Laboratoria: 15 Projekty/seminaria: -		Liczba punktów 3
Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny) kierunkowy		(ogólnouczelniany, z innego kierunku) z danego kierunku
Obszar(y) kształcenia i dziedzina(y) nauki i sztuki		Podział ECTS (liczba i %)
Odpowiedzialny za przedmiot / wykładowca:		
<p>Prof. dr hab. inż. M. Drozdowski, prof. nadzw. PP. email: Maciej.Drozdowski@cs.put.poznan.pl tel. 6652981 Instytut Informatyki ul. Piotrowo 2, 60-965 Poznań</p>		
Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:		
1	Wiedza:	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z matematyki dyskretnej (teoria mnogości, logika, kombinatoryka, teoria grafów), wiedzę o metodach optymalizacji, konstrukcjach programistycznych, abstrakcyjnych typach danych (np. listy, stosy, kolejki, drzewa, dowolne grafy), typowych algorytmach (np. sortowanie, wyszukiwanie danych), podstawową wiedzę o złożoności obliczeniowej problemów i algorytmów.
2	Umiejętności:	Powinien posiadać umiejętność rozwiązywania podstawowych problemów algorytmicznych, zaprogramowania rozwiązań tych problemów, rozpoznawania struktur dyskretnych, szacowania złożoności algorytmów oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.
3	Kompetencje społeczne	Powinien również rozumieć konieczność poszerzania swoich kompetencji i mieć gotowość do podjęcia współpracy w ramach zespołu. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi.
Cel przedmiotu:		
Poznanie wybranych problemów optymalizacji kombinatorycznej i metod ich rozwiązywania, a w szczególności:		
<ol style="list-style-type: none"> Przekazanie studentom podstawowej wiedzy o optymalizacji w odniesieniu do problemów, które mają dyskretną naturę. Zademonstrowanie i uświadomienie bariery rozwiązywalności wynikającej z wykładniczej złożoności obliczeniowej algorytmów i trudności obliczeniowej problemów. Rozwijanie u studentów umiejętności rozpoznawania trudnych problemów optymalizacji. Zapoznanie z metodologią analizowania i praktycznego rozwiązywania trudnych obliczeniowo zadań optymalizacji w przypadku problemów o dyskretnej naturze. 		
Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia		
Wiedza:		
<ol style="list-style-type: none"> ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie kluczowych zagadnień informatyki, oraz wiedzę szczegółową w zakresie wybranych zagadnień tej dyscypliny nauki - [K1s_W4] ma wiedzę o istotnych kierunkach rozwoju i najważniejszych osiągnięciach informatyki oraz innych pokrewnych dyscyplin naukowych, w szczególności elektroniki, telekomunikacji oraz automatyki i robotyki - [K1st_W5] zna podstawowe techniki, metody oraz narzędzia wykorzystywane w procesie rozwiązywania zadań informatycznych, głównie o charakterze inżynierskim, z zakresu kluczowych zagadnień informatycznych z zakresu analizy złożoności obliczeniowej algorytmów i problemów - [Kst1_W7] 		
Umiejętności:		

1. potrafi właściwie zaplanować oraz wykonać eksperymenty, w tym pomiary oraz symulacje komputerowe, dokonać interpretacji uzyskanych rezultatów, oraz poprawnie wyciągnąć płynące z nich wnioski uzyskane wyniki i wyciągać wnioski - [K1st_U3]
2. potrafi, formułując i rozwiązując zadania informatyczne, zastosować odpowiednio dobrane metody, w tym metody analityczne, symulacyjne lub eksperymentalne - [K1st_U4]
3. potrafi ocenić złożoność obliczeniową algorytmów i problemów - [K1st_U8]
4. ma umiejętność formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi - [K1st_U11]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K1st_K1]
2. ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich oraz zna przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K1st_K2]

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 ćwiczeń:
 - na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę sprawozdania przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu; ocena ta obejmuje także umiejętność pracy w zespole,
- ocenę wiedzy i umiejętności z wykładu na kolokwium.

Uzyskiwanie punktów dodatkowych na laboratorium za:

- terminowość realizacji zadania,
- efektywność zastosowanego rozwiązania,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi umożliwiające doskonalenie procesu dydaktycznego.

Treści programowe

Program przedmiotu obejmuje następujące zagadnienia: Algorytmy programowania dynamicznego o złożoności pseudowielomianowej dla problemów: plecakowego, podziału zbioru. Silna NP-zupełność. Złożoność obliczeniowa problemów optymalizacyjnych ? NP-trudność. Pojęcie algorytmu aproksymacyjnego z przykładami. Trudność problemów aproksymacji. Problemy optymalizacji kombinatorycznej, dla których istnieją algorytmy wielomianowe: Wyznaczanie najkrótszych dróg w grafie: algorytm Dijkstry; algorytm dla grafów acyklicznych; algorytm wyznaczania odległości między wszystkimi parami wierzchołków. Domknięcie przechodnie relacji: algorytm Floyd-Warshalla.

Przepływy w sieciach i pokrewne zagadnienia: wyznaczanie maksymalnego przepływu w sieci metodą Dinica; wyznaczanie przepływu w sieci z dolnymi ograniczeniami na przepływy łukowe; wyznaczanie przepływu o najniższym koszcie; wyznaczanie maksymalnego skojarzenia: w grafie dwudzielnym, przykłady zastosowania problemu wyznaczania maksymalnego przepływu do rozwiązywania zagadnień szeregowania zadań z minimalizacją spóźnienia i do wyznaczania optymalnego podziału sieci komunikujących się procesów na dwa procesory. Minimalne drzewo rozpinające - algorytmy Prima i Kruskala. Pojęcie matroidu. Problem kolorowania grafu: sformułowanie, zastosowania, metody rozwiązania. Problem pakowania ? sformułowanie, zastosowania, algorytmy rozwiązujące ten problem.

W ramach zajęć laboratoryjnych studenci rozwiązują NP-trudne problemy kombinatoryczne. Należy przygotować co najmniej dwie metody rozwiązujące zadany problem: szybką (np. algorytm zachłanny) i konstruującą rozwiązania o dobrej jakości (np. algorytm podziału i ograniczeń, metaheurystykę).

Metody dydaktyczne:

1. wykład: prezentacja ilustrowana przykładami podawanymi na tablicy, rozwiązywanie zadań.
2. ćwiczenia laboratoryjne: praktyczne zapoznanie się z wyzwaniami optymalizacji kombinatorycznej przez implementację algorytmów rozwiązujących zadane problemy, wykonywanie eksperymentów obliczeniowych oceniających jakość rozwiązań i czas ich uzyskania, dyskusja, praca w zespole,

Literatura podstawowa:		
1. Złożoność obliczeniowa problemów kombinatorycznych, J. Błażewicz , WNT, W-wa, 1988		
2. Scheduling Computer and Manufacturing Processes, J. Błażewicz, K. Ecker, E.Pesch, G. Schmidt, J. Węglarz , Springer, Berlin, New York, 2001		
3. Kombinatoryka dla programistów, W. Lipski , WNT, W-wa, 1982		
4. Computers and intractability: A guide to the theory of NP-completeness, M.R.Garey, D.S.Johnson, W.H.Freeman, San Francisco, 1979		
5. Combinatorial optimization, W.Cook, W.Cunningham, W.Pulleyblank, A.Schrijver, Wiley & Sons, 1998		
6. Algorytmy optymalizacji dyskretnej z programami w języku Pascal, M.Sysło, N.Deo, J.Kowalik, PWN, Warszawa, 1993		
7. Wprowadzenie do algorytmów, T.Cormen, C.Leiserson, R.Rivest, C.Stein, WNT, Warszawa, 2005		
8. Optymalizacja dyskretna modele i metody kolorowania grafów, pod red. M.Kubale, WNT, Warszawa, 2003.		
Literatura uzupełniająca:		
1. M.Drozdowski, D.Kowalski, J.Mizgajski, D.Mokwa, G.Pawlak, Mind the gap: a heuristic study of subway tours, Journal of Heuristics vol.20, Issue 5, October 2014, pp 561-587, DOI 10.1007/s10732-014-9252-3		
2. Jakub Marszałkowski, D.Mokwa, M.Drozdowski, Ł.Rusiecki, H.Narożny, Fast algorithms for online construction of web tag clouds, Engineering Applications of Artificial Intelligence, vol. 64 (2017) pp. 378-390 DOI: 10.1016/j.engappai.2017.06.023		
Bilans nakładu pracy przeciętnego studenta		
Czynność	Czas (godz.)	
1. udział w zajęciach ćwiczeniach:	15	
2. dokończenie (w ramach pracy własnej) sprawozdań z ćwiczeń:	5	
3. napisanie programów, uruchomienie, weryfikacja, testowanie wydajności (czas poza zajęciami)	10	
4. udział w wykładach	30	
5. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	10	
6. przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym.	10	
Obciążenie pracą studenta		
forma aktywności	godzin	ECTS
Łączny nakład pracy	80	3
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2
Zajęcia o charakterze praktycznym	30	2