



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Algorytmy i modele inspirowane biologicznie

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Sztuczna inteligencja

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1/2

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

Liczba godzin

Wykład

15

Ćwiczenia

Laboratoria

15

Projekty/seminaria

Inne (np. online)

Liczba punktów ECTS

3

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Maciej Komosiński

email: maciej.komosinski@put.poznan.pl

tel: 61 665-2931

Instytut Informatyki

ul. Piotrowo 2, 60-965 Poznań

Odpowiedzialny za przedmiot/wykładowca:

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę na temat złożoności obliczeniowej, algorytmów uczenia maszynowego i sztucznych sieci neuronowych.

Powinien posiadać umiejętność modelowania i rozwiązywania prostych problemów optymalizacyjnych, umiejętności programistyczne oraz umiejętność pozyskiwania informacji ze wskazanych źródeł.

Powinien również rozumieć konieczność poszerzania swoich kompetencji. Ponadto w zakresie kompetencji społecznych student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, oraz szacunek dla innych ludzi.

Cel przedmiotu

1. Przekazanie wiedzy na temat algorytmów optymalizacyjnych inspirowanych biologicznie takich jak



algorytmy ewolucyjne, mrówkowe, roju cząstek, pszczele.

2. Przekazanie wiedzy o roli mechanizmów biologicznych stosowanych w algorytmach, ich zaletach i wadach.
3. Przekazanie wiedzy o wspólnych cechach i o jednorodnym ujęciu wszystkich algorytmów optymalizacji.
4. Przekazanie wiedzy z zakresu sztucznego życia oraz symulacji modeli biologicznych.
5. Rozwinięcie u studentów umiejętności wydajnej implementacji oraz oceny efektywności algorytmów optymalizacji – zarówno czasowej, jak i jakościowej.
6. Kształtowanie u studentów umiejętności wyciągania wniosków z samodzielnie prowadzonych badań i tworzenia raportów z eksperymentów obliczeniowych oraz właściwej wizualizacji rezultatów

Przedmiotowe efekty uczenia się

Wiedza

ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu algorytmów inspirowanych biologicznie [K2st_W2]

ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu informatyki [K2st_W3]

ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki i innych, wybranych, pokrewnych dyscyplin naukowych [K2st_W4]

ma zaawansowaną i szczegółową wiedzę o procesach zachodzących w cyklu życia systemów informatycznych [K2st_W5]

zna zaawansowane metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań wymagających tworzenia modeli symulacyjnych [K2st_W6]

Umiejętności

potrafi pozyskiwać informacje o konstrukcji algorytmów z literatury, baz danych oraz innych źródeł (w języku polskim i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinie [K2st_U1]

potrafi planować i przeprowadzać eksperymenty, w tym pomiary i symulacje komputerowe, interpretować uzyskane wyniki i wyciągać wnioski oraz formułować i weryfikować hipotezy [K2st_U3]

potrafi wykorzystać do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych metody analityczne, symulacyjne oraz eksperymentalne [K2st_U4]

potrafi — przy formułowaniu i rozwiązywaniu zadań — integrować wiedzę z różnych obszarów informatyki, takich jak biologia czy nauki społeczne (a w razie potrzeby także wiedzę z innych dyscyplin



naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne [K2st_U5]

potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych [K2st_U6:]

potrafi dokonać krytycznej analizy istniejących rozwiązań algorytmicznych oraz zaproponować ich ulepszenia (usprawnienia) [K2st_U8]

potrafi – stosując m.in. koncepcyjnie nowe metody – rozwiązywać złożone zadania optymalizacji, w tym zadania nietypowe oraz zadania zawierające komponent badawczy [K2st_U10]

Kompetencje społeczne

rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe [K2st_K1]

rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu algorytmów optymalizacji w rozwiązywaniu problemów badawczych i praktycznych [K2st_K2]

ma świadomość potrzeby rozwijania dorobku zawodowego oraz przestrzegania zasad etyki zawodowej [K2st_K4]

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się 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 sprawdzianie składającym się z kilkunastu pytań o charakterze testu lub krótkich zadań. Przekroczenie 50% punktów pozwala uzyskać ocenę dostateczną.

- omówienie wyników sprawdzianu,

b) w zakresie laboratoriów weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę umiejętności związanych z realizacją ćwiczeń laboratoryjnych,

- ocenianie ciągłe, na każdym zajęciach (odpowiedzi ustne) - premiowanie przyrostu umiejętności posługiwania się poznanymi zasadami i metodami,



- ocenę sprawozdań przygotowywanych częściowo w trakcie zajęć, a częściowo po ich zakończeniu, z możliwością wykorzystania platformy Moodle,
- dokonanie prezentacji wyników własnych eksperymentów.

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- przeprowadzenie rozszerzonych, nieobowiązkowych eksperymentów w ramach zadań laboratoryjnych oraz ich opisanie w sprawozdaniu,
- uwagi pozwalające udoskonalić materiały dydaktyczne.

Treści programowe

Program wykładu obejmuje następujące zagadnienia:

Architektury i ocena działania algorytmu ewolucyjnego. Techniki selekcji, krzyżowania, mutacji, skalowania ocen rozwiązań. Twierdzenie o schematach i epistaza. Nieporządnym algorytm genetyczny. Hierarchiczny algorytm genetyczny. Mechanizmy inspirowane naturą w algorytmach ewolucyjnych. Strategie ewolucyjne. Ewolucja różnicowa. Programowanie ewolucyjne: reprezentacja zmiennoprzecinkowa, embriogeneza, operatory genetyczne a globalna wypukłość. Programowanie genetyczne i regresja symboliczna. Systemy klasyfikatorowe: ewolucyjne uczenie maszynowe. Rozwiązywanie problemów wielokryterialnych za pomocą algorytmów ewolucyjnych. Równoległe algorytmy ewolucyjne. Architektury koewolucyjne – kooperatywne i konkurencyjne. Problemy i patologie w koewolucji oraz sposoby ich minimalizowania. Omówienie innych inspirowanych biologicznie technik optymalizacji: algorytmów mrówkowych i stygmergii (AA/ACO), algorytmów roju cząstek (PSO), sztucznych systemów odpornościowych (AIS), algorytmów pszczelich (ABC). Ewolucja spontaniczna a ukierunkowana. Ewolucja ograniczona a otwarta. Projektowanie ewolucyjne i robotyka ewolucyjna; mapowanie genotyp-fenotyp, morfogeneza i modularność.

Zajęcia laboratoryjne są poświęcone następującym zagadnieniom:

Wybór problemu optymalizacji kombinatorycznej, określenie przestrzeni rozwiązań, reprezentacji rozwiązania, definicji funkcji celu. Propozycja algorytmu heurystycznego. Implementacja algorytmu przeszukiwania lokalnego oraz jednej z metaheurystyk inspirowanych biologicznie. Porównanie jakości uzyskiwanych rozwiązań oraz czasu działania algorytmów. Eksperymenty z projektowaniem ewolucyjnym konstrukcji trójwymiarowych i robotyką ewolucyjną.

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.

Ćwiczenia laboratoryjne: prezentacja ilustrowana przykładami podawanymi na tablicy oraz wykonanie zadań podanych przez prowadzącego – ćwiczenia praktyczne.

Literatura



Podstawowa

1. D.E. Goldberg, Algorytmy genetyczne i ich zastosowania, WNT, Warszawa, 2009.
2. Z. Michalewicz, Algorytmy genetyczne + struktury danych = programy ewolucyjne, WNT, Warszawa, 2003.

Uzupełniająca

1. Jason Brownlee, Clever Algorithms: Nature-Inspired Programming Recipes, 2012.
<https://github.com/clever-algorithms/CleverAlgorithms>
2. El-Ghazali Talbi, Metaheuristics: From Design to Implementation, Wiley, 2009.

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	75	3,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	30	1,5
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwium, wykonanie implementacji i przeprowadzenie eksperymentów) ¹	45	1,5

¹ niepotrzebne skreślić lub dopisać inne czynności