



KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Praktyka i teoria szeregowania zadań

Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

-

Poziom studiów

pierwszego stopnia

Forma studiów

niestacjonarne

Rok/semestr

4/7

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obieralny

Liczba godzin

Wykład

12

Laboratoria

16

Inne (np. online)

0

Ćwiczenia

0

Projekty/seminaria

0

Liczba punktów ECTS

4

Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Małgorzata Sterna, prof. PP

Odpowiedzialny za przedmiot/wykładowca:

dr Maciej Machowiak

Wymagania wstępne

Student rozpoczynający przedmiot powinien posiadać podstawową wiedzę z zakresu algorytmów i struktur danych, złożoności obliczeniowej oraz programowania w dowolnym języku wysokiego poziomu. Winien posiadać umiejętność analizy zagadnień o charakterze praktycznym, rozwiązywania problemów kombinatorycznych, zaimplementowania algorytmów rozwiązujących takie problemy, oceny jakości algorytmów, a także umiejętność pozyskiwania informacji ze wskazanych źródeł. Ponadto student powinien wykazywać się samodzielnością, odpowiedzialność za pracę wykonywaną indywidualnie oraz w grupie, zdolnością prezentacji w formie ustnej i pisemnej ogólnej koncepcji i sposobu rozwiązania analizowanych problemów.

Cel przedmiotu

Przekazanie studentom podstawowej wiedzy na temat problematyki szeregowania zadań na przykładzie podstawowych modeli z pojedynczą maszyną, maszynami równoległymi i dedykowanymi.

Przedstawienie wybranych algorytmów szeregowania zadań, wykorzystujących różne techniki algorytmiczne, m.in. z zakresu teorii grafów czy geometrii, oraz prezentacja sposobu analizy i

rozwiązywania zagadnień szeregowania z wykorzystaniem osiągnięć teorii złożoności obliczeniowej.

Rozwijanie u studentów umiejętności rozwiązywania problemów szeregowania od etapu modelowania



zagadnienia praktycznego, przez dobór/projektowanie algorytmu, po ocenę jego efektywności w eksperymencie obliczeniowym. Kształtowanie u studentów umiejętności przeprowadzania prostych eksperymentów obliczeniowych i opracowywania ich wyników oraz umiejętności pracy zespołowej w trakcie rozwiązywania zadań podczas zajęć laboratoryjnych.

Przedmiotowe efekty uczenia się

Wiedza

1. Student ma wiedzę ogólną dotyczącą podstaw teorii szeregowania zadań ze szczególnym uwzględnieniem systemów obsługi czyli systemów z procesorami dedykowanymi.
2. Zna klasyczne algorytmy szeregowania zadań w szczególności dla pojedynczej maszyny, maszyn równoległych i dla dwu-maszynowych systemów obsługi.
3. Zna przykłady wykorzystania różnych technik algorytmicznych oraz metod złożoności obliczeniowej w rozwiązywaniu problemów szeregowania zadań.

Umiejętności

1. Student potrafi zaproponować algorytmy, głównie heurystyczne, rozwiązujące problem szeregowania zadań.
2. Potrafi zaimplementować i ocenić efektywność zaproponowanego algorytmu z różnych punktów widzenia.
3. Potrafi uczestniczyć w dyskusji nad różnymi koncepcjami rozwiązania problemu szeregowania zadań i oceną ich jakości.
4. Potrafi zaprojektować i przeprowadzić eksperyment obliczeniowy mający na celu ocenę efektywności prostych algorytmów szeregowania zadań.
5. Potrafi dostosować dostępne w literaturze algorytmy i metody do rozwiązania postawionego problemu szeregowania.

Kompetencje społeczne

1. Student ma świadomość ciągłego rozwoju technik algorytmicznych wykorzystywanych do rozwiązywania problemów praktycznych.
2. Ma świadomość wsparcia jakie w rozwiązywaniu problemów praktycznych dostarczają wyniki badań naukowych.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza i umiejętności nabyte podczas wykładu weryfikowane są podczas 90-minutowego kolokwium zaliczeniowym przeprowadzanego pod koniec semestru. Kolokwium ma formę testu, składającego się z kilkunastu różnie punktowanych pytań otwartych i zamkniętych, wymagających wykazania się wiedzą teoretyczną oraz samodzielnego zastosowania technik algorytmicznych i metod modelowania



prezentowanych w trakcie wykładu. Do uzyskania oceny dostatecznej konieczne jest przekroczenie progu 50% maksymalnej liczby punktów, progi kolejnych ocen ulegają zwiększeniu o 10%.

Umiejętności nabyte w trakcie zajęć laboratoryjnych weryfikowane są na podstawie sposobu wykonania 2 zadań. Każde zadanie wymaga zaprojektowania, implementacji oraz przetestowania w eksperymencie obliczeniowym algorytmów rozwiązujących proste problemy szeregowania zadań i procedur pomocniczych, oraz przygotowania pisemnego sprawozdania. Do uzyskania oceny dostatecznej konieczne jest zaliczenie obu zadań laboratoryjnych.

Treści programowe

W ramach wykładu przedstawiane są następujące zagadnienia:

1. Ogólna definicja problemu szeregowania zadań, podstawowe parametry zadań i procesorów oraz funkcje kryterialne wraz z ich interpretacją praktyczną.
2. Klasyfikacja problemów szeregowania zadań z użyciem notacji trójpolowej wraz z jej odniesieniem do problemów spotykanych w praktyce.
3. Przykłady dowodzenia trudności zagadnień szeregowania, klasyfikacja problemów szeregowania z uwagi na złożoność obliczeniową.
4. Wybrane problemy szeregowania na pojedynczej maszynie, algorytmy listowe, algorytm Jacksona, algorytm Horna, reguła Smitha, algorytm Hodgsona.
5. Wybrane problemy szeregowania na maszynach równoległych, algorytm McNaughtona, algorytm Hu, algorytmy listowe, przykłady algorytmów aproksymacyjnych, programowania dynamicznego, wykorzystania algorytmów grafowych w rozwiązywaniu zagadnień szeregowania.
6. Ogólna definicja problemu szeregowania zadań w systemach obsługi wraz z przykładowymi zastosowaniami tego modelu w rozwiązywaniu zagadnień praktycznych.
7. Wykorzystanie modeli grafowych do efektywnej reprezentacji problemów szeregowania zadań.
8. System otwarty: algorytm Gonzaleza-Sahniego i trudność problemów wielomaszynowych, przykład podejścia opartego o programowanie matematyczne.
9. System przepływowy: algorytm Johnsona i algorytmy heurystyczne dla systemów wielomaszynowych (metoda agregacji), metoda geometryczna.
10. System ogólny: algorytm Jacksona i uogólniona metoda geometryczna, algorytmy listowe.

W ramach zajęć laboratoryjnych studenci realizują 2 zadania, dotyczące podanych problemów szeregowania, z których każde obejmuje:

1. Indywidualne zaprojektowanie, zaimplementowanie i weryfikację w eksperymencie obliczeniowym prostego algorytmu heurystycznego rozwiązującego zadany problem szeregowania zadań (konkurs na najbardziej efektywny algorytm).



2. Indywidualne zaprojektowanie i zaimplementowanie generatora instancji testowych (konkurs na najlepszy generator instancji) i weryfikatora poprawności uzyskiwanych rozwiązań.
3. Zespołowe (w grupie laboratoryjnej) wykonanie eksperymentów obliczeniowych z wykorzystaniem wygenerowanych instancji, w tym weryfikacja własnego algorytmu i algorytmów innych członków grupy oraz analiza uzyskanych wyników.
4. Przygotowanie pisemnego sprawozdania z wykonanego ćwiczenia laboratoryjnego.

Metody dydaktyczne

1. Wykład ilustrowany prezentacją multimedialną zawierającą omawiane treści programowe, wzbogaconą przykładami.
2. Ćwiczenia laboratoryjne ilustrujące materiał prezentowany podczas wykładu, polegające na analizie i rozwiązywaniu prostych problemów szeregowania zadań, obejmujące samodzielne projektowanie i implementację algorytmów, planowanie i przeprowadzanie eksperymentów obliczeniowych, przygotowanie sprawozdań, dyskusje i ustne prezentacje rozwiązań, negocjacje w szerokiej grupie studenckiej sposobu przygotowania wspólnego środowiska testowego oraz zasad oceny jakości algorytmów.

Literatura

Podstawowa

1. Handbook on scheduling: from theory to practice, J. Błażewicz, K. Ecker, E. Pesch, M. Sterna, G. Schmidt, J. Węglarz, Springer, Cham, 2019.
2. Planning and scheduling in manufacturing and services, M. Pinedo, Springer, New York, 2007.
3. Late work scheduling in shop systems, M. Sterna, Rozprawy nr 405, Wydawnictwo PP, Poznań, 2006.
4. Badania operacyjne dla informatyków, J. Błażewicz, W. Cellary, R. Słowiński, J. Węglarz, WNT, Warszawa, 1983.

Uzupełniająca

1. Scheduling: theory, algorithms, and systems, M. Pinedo, Springer, New York, 2016.
2. Scheduling algorithms, P. Brucker, Springer, Berlin, 2007.
3. Handbook of scheduling: algorithms, models, and performance analysis, J.Y.-T. Leung (eds.), CRC Press, Boca Raton, 2004.
4. Operations Research. A Practical Introduction, M.W. Carter, C.C. Price, CRC Press, Boca Raton, 2001.
5. Handbook of combinatorial optimization, D.-Z. Du, P.M. Pardalos (eds.), Kluwer Academic Publishers, Boston, 1998.



6. The disjunctive graph machine representation of the job shop scheduling problem, J. Błażewicz, E. Pesch, M. Sterna, *European Journal of Operational Research* 127/2 (2000), 317-331 ([https://doi.org/10.1016/S0377-2217\(99\)00486-5](https://doi.org/10.1016/S0377-2217(99)00486-5))
7. Open shop scheduling problems with late work criteria, J. Błażewicz, E. Pesch, M. Sterna, F. Werner, *Discrete Applied Mathematics* 134 (2004), 1-24 ([https://doi.org/10.1016/S0166-218X\(03\)00339-1](https://doi.org/10.1016/S0166-218X(03)00339-1))
8. Scheduling on parallel identical machines with late work criterion. Offline and online cases. X. Chen, M. Sterna, X. Han, J. Błażewicz, *Journal of Scheduling* 19/6 (2016), 729-736 (<https://doi.org/10.1007/s10951-015-0464-7>)
9. A survey of scheduling problems with late work criteria, M. Sterna, *Omega* 39/2 (2011), 120-129 (<https://doi.org/10.1016/j.omega.2010.06.006>)
10. Polynomial time approximation scheme for two parallel machines scheduling with a common due date to maximize early work, M. Sterna, K. Czerniachowska, *Journal of Optimization Theory and Applications* 174/3 (2017), 927-944, (<https://doi.org/10.1007/s10957-017-1147-7>)

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	103	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	28	1,0
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych, przygotowanie do kolokwium, przygotowanie sprawozdań z wykonania zadań laboratoryjnych) ¹	75	3,0

¹ niepotrzebne skreślić lub dopisać inne czynności