

Projekt z PSZT, semestr letni 2017

Zasady ogólne:

1. Dany projekt jest prowadzony przez jednego z sześciu prowadzących:
Waldemar Grabski, w.grabski@elka.pw.edu.pl
Tomasz Trzcinski, t.trzcinski@ii.pw.edu.pl
Paweł Wawrzyński, p.wawrzynski@elka.pw.edu.pl
2. Projekty są oznaczone inicjałami prowadzącego oraz literami wskazującymi na tematykę.
3. Każdy z prowadzących określa szczegółowe zasady, na podstawie których zalicza oraz przyznaje punkty za swoje projekty, niekiedy określa także dodatkowe zasady dotyczące projektów w ramach tej samej tematyki. Należy się z tymi zasadami zapoznać.
4. O ile prowadzący nie określi tego inaczej, zadania należy realizować używając języka programowania średniego poziomu: C++, C#, Java, Python itp.
5. O ile prowadzący nie określi tego inaczej, oddanie projektu obejmuje przekazanie działającego programu/programów oraz 3-stronnicowej dokumentacji zawierającej: (i) zestawienie kluczowych decyzji projektowych, (ii) instrukcję dla użytkownika wystarczającą do uruchomienia programu i jego głównych funkcji, (iii) opis struktury programu, (iv) wnioski dotyczące osiągniętych rezultatów.

Harmonogram:

Do 13 czerwca 2017 projekty są oddane.

Przydział zespołów do tematów i prowadzących

Zespół	Projekt
Mateusz Dziwulski, Kuba Szczepański, Adam Sierpiński	TT.SN.1
Robert Piwowarek, Agata Kloss, Marcin Kondras	WG.GD.1
Irena Mikucka, Adam Kowalewski, Eryk Ratyński	TT.SN.2
Rafał Galczak, Wojciech Gruszka, Jakub Gajownik	TT.SN.3
Bartłomiej Niespodziany, Piotr Poskart, Natalia Wochtman	TT.SN.4
Jakub Lański, Michał Lichac, Artur Mruk	TT.SN.5
Albert Buczak, Agata Chadaj, Adrian Hawrylak	PW.AE.1
Mateusz Koroś, Ksawery Pasikowski, Bartek Marchocki	PW.LR.1
Damian Rogowski, Tomasz Buchholz, Maksym Bakhmut	TT.SN.6
Wojciech Janaszek, Paweł Ostaszewski, Marcin Chrostowski	WG.GD.2
Hubert Buczyński, Adam Korcza, . Maciej Pczek Kędzielsi.	TT.SN.7
Agnieszka Dunajska, Ewa Koguciuk, Bartosz Grudniewski	PW.AE.2
Sebastian Przybył, Rafał Ambroziak, Paweł Kaczmarczyk	PW.AE.3
Marcin Baran, Mateusz Perciński, Łukasz Kilaszewski	WG.AE.1
Michał Sieczkowski, Bartosz Wojciechowski, Igor Klemenski	TT.P.1
Katarzyna Rybacka, Konrad Kucharczyk, Karol Chęciński	WG.GD.3
Dardziński Jakub, Kultys Jakub, Olszewski Kamil	WG.GD.4
Martyna Czwarno, Agnieszka Filimon, Kamila Kulas	TT.SN.8
Paweł Chudzik, Paweł Kaźmierczyk	PW.AE.4
Kurek Dariusz, Sykulski Piotr, Wardak Aleksandra	WG.AE.2
Milena Jachimowska, Michał Misiewicz, Michał Mokrogulski	WG.AE.3
Piotr Denert, Mateusz Mariański, Konrad Stalęga	PW.LR.2
Paweł Koprowski, Marcin Stelmaszczyk, Miłosz Wyrzykowski	PW.AE.5
Karolina Cabaj, Konrad Szczepaniak, Piotr Czyż	PW.AE.6
Mateusz Marzec	WG.GD.5
Aleksander Rurarz, Jan Ambroziak, Michał Bloch	WG.GD.6
Mateusz Marzec, Jacek Palczewski	PW.W.1

Część WG – prowadzący Waldemar Grabski

WG Zasady i zalecenia

Konsultacje: wtorek 14-16 pok. 320, e-mail: W.Grabski@elka.pw.edu.pl

Projekt powinien być napisany przez studentów bez użycia zewnętrznych bibliotek wysokiego poziomu. W razie wątpliwości dotyczących zastosowania konkretnej biblioteki należy jej zastosowanie skonsultować z prowadzącym. Algorytm (ewolucyjny/grający) powinien być oddzielony od rozwiązywanego problemu. W dokumentacji końcowej poza krótkim opisem rozwiązania ma znaleźć się raport z przeprowadzonych testów oraz informacja jakie zadania wykonał każdy z członków zespołu.

Zalecane są co najmniej dwie konsultacje:

- po przydziale projektów – doprecyzowanie zadania,
- z projektem rozwiązania – uzgodnienie zakresu implementacji i dokumentacji końcowej.

Termin oddania/prezentacji projektu to: 13.06.2017. Oddanie projektu po tej dacie będzie uzależnione od dostępności prowadzącego, a ocena za taki projekt będzie niższa o 10pkt. Projekty mogą być konsultowane do 7 dni przed terminem oddania, tj. do 6.06.2017.

Przed oddaniem/prezentacją projektu należy wysłać do prowadzącego (e-mail z tematem PSZT <nazwa projektu>) kod oraz dokumentację.

WG.AE1 Rozwożenie mebli

Zaplanować trasę samochodu ciężarowego rozwożącego meble. Każdy mebel ma określoną wagę oraz miasto przeznaczenia. Zużycie paliwa przez samochód ciężarowy jest zależne od masy przewożonego ładunku. Zaplanować trasę rozwiezienia mebli która jest optymalna ze względu na zużycie paliwa. Program powinien na bieżąco prezentować jakość znalezionego rozwiązania w funkcji numeru pokolenia.

WG.AE2 Kampania w grze strategicznej

W grze na planszy 1000x1000 pól każdy gracz posiada 1 miasto oraz armię (piechota lub kawaleria). Piechota porusza się z prędkością 1 pole na godzinę a kawaleria 3 pola na godzinę. Dla klanu graczy A zaplanować kampanię przeciwko klanowi B. w której w jak najkrótszym czasie gracze klanu A mają zaatakować wszystkie miasta graczy klanu B. Armia gracza klanu A po powrocie może wykonać kolejny atak. Każde miasto klanu B ma być zaatakowane dokładnie raz. Program powinien na bieżąco prezentować jakość znalezionego rozwiązania w funkcji numeru pokolenia.

WG.AE3 Planowanie symulacji

Zaplanować proces przeprowadzenia N symulacji na farmie M stacji roboczych.

Każda stacja robocza ma określoną moc obliczeniową oraz ilość dostępnej pamięci. Każda symulacja wymaga określonej ilości pamięci i wykonuje określoną ilość operacji. Na stacji roboczej wykonywane są symulacje którym przydzielamy określoną ilość pamięci (taką jak jest wymagana) i % mocy obliczeniowej (im większa moc obliczeniowa przydzielona tym szybciej symulacja się zakończy) – wartości te nie zmieniają się podczas wykonania symulacji. Program powinien na bieżąco prezentować jakość znalezionego rozwiązania w funkcji numeru pokolenia.

WG.GD1 – GD3 Reversi

Napisać program grający w Reversi (plansza gry o rozmiarze od 8x8 do 32x32). Opis gry: <https://pl.wikipedia.org/wiki/Reversi>

Algorytm Alfa-Beta z iteracyjnym pogłębianiem. Program ma ograniczony czas na wykonanie ruchu.

Program ma pracować w 2 trybach:

- gra z człowiekiem (prosty interfejs graficzny),
- programem innego zespołu (ma być prezentowana plansza po każdym ruchu oraz czas w jakim ruch został wykonany), sposób komunikacji pomiędzy programami zostanie ma zostać uzgodniony pomiędzy zespołami i opisany w projekcie wstępnym.

Na koniec projektu zostanie przeprowadzony turniej. Zajęte miejsce będzie miało wpływ na ocenę projektu.

WG.GD4 – GD6 Go

Napisać program grający w Go (plansza gry o rozmiarze od 9x9 do 19x19). Opis gry:

<https://pl.wikipedia.org/wiki/Go>

Algorytm Alfa-Beta z iteracyjnym pogłębianiem. Program ma ograniczony czas na wykonanie ruchu.

Program ma pracować w 2 trybach:

- gra z człowiekiem (prosty interfejs graficzny),
- programem innego zespołu (ma być prezentowana plansza po każdym ruchu oraz czas w jakim ruch został wykonany), sposób komunikacji pomiędzy programami zostanie ma zostać uzgodniony pomiędzy zespołami i opisany w projekcie wstępnym.

Na koniec projektu zostanie przeprowadzony turniej. Zajęte miejsce będzie miało wpływ na ocenę projektu.

WG.GD7 Pente

Napisać program grający z człowiekiem w Pente. Opis gry:

<https://pl.wikipedia.org/wiki/Pente>

Algorytm Alfa-Beta z iteracyjnym pogłębianiem. Program ma ograniczony czas na wykonanie ruchu. Aplikacja ma działać w 2 trybach:

- gra z człowiekiem,
- gra programu sam ze sobą (wyświetlanie planszy po każdym ruchu).

Program ma posiadać prosty interfejs graficzny.

Część TT – prowadzący Tomasz Trzciński

Konsultacje: wtorek 14-16 pok. 415A (proszę o maila przed pojawieniem się, jeśli to możliwe)

Projekt oceniany będzie na podstawie następujących kryteriów:

- Dokumentacja: powinna być zwięzła i dotyczyć zastosowanego algorytmu.
- Kod: powinien być przejrzysty oraz odpowiednio skomentowany/opisany.
- Poprawność rozwiązania: odpowiedź na zadany problem oraz ocena jego skuteczności.
- Wydajność: powinna być wystarczająca, żeby zaprezentować funkcjonujący kod na obronie projektu.

Projekt powinien być napisany przez studentów bez użycia zewnętrznych bibliotek wysokiego poziomu. W razie wątpliwości dotyczących zastosowania konkretnej biblioteki należy jej zastosowanie skonsultować z prowadzącym.

W ramach projektu udostępnione dane powinny być rozdzielone na część treningową oraz testową, a rezultaty powinny zostać poddane walidacji k-fold (gdzie $k=5$ lub 10).

Termin oddania: 13.06.2017. Oddanie projektu po tej dacie będzie uzależnione od dostępności prowadzącego, a ocena za taki projekt będzie niższa o 10pkt.

Projekty mogą być konsultowane do 7 dni przed terminem oddania, tj. do 6.06.2017.

Przed wejściem na obronę należy wysłać do prowadzącego plik .zip z nazwą PSZT_[NAZWAPROJEKTU].zip zawierający kod wraz z dokumentacją.

TT.NN.1

Sieci neuronowe z wieloma ukrytymi warstwami są często stosowane do rozwiązywania problemów klasyfikacji obrazów. Zadaniem jest zbudowanie sieci neuronowej z kilkoma ukrytymi warstwami w celu klasyfikacji cyferek na obrazkach. Zbiór danych jest na przykład dostępny w bazach MNIST (https://en.wikipedia.org/wiki/MNIST_database).

TT.NN.2

Stwórz sieć neuronową, która diagnozuje postać raka na podstawie parametrów wyekstraktowanych ze zdjęć takich jak tekstura czy promień. [dane](#)

TT.NN.3

Stwórz sieć neuronową, która rozpoznaje język na podstawie paragrafu (10-20 słów). Sieć powinna rozpoznawać co najmniej 5 języków.

TT.NN.4

Na podstawie załączonych danych, stwórz algorytm klasyfikacji przy użyciu sieci neuronowej, który przewidzi czy dany pacjent nie przyjdzie na planowaną konsultację. [dane](#)

TT.NN.5

Celem zadania jest stworzenie algorytmu klasyfikacji z użyciem sieci neuronowych, który klasyfikuje typ cukrzycy na podstawie podanych danych. [dane](#)

TT.NN.6

Stwórz algorytm predykcji w oparciu o sieć neuronową, który wskaże, czy danej osobie

zostanie wydana wiza amerykańska typu H1B czy nie. [dane](#)

TT.NN.7

Na podstawie załączonych danych, stwórz algorytm klasyfikacji przy użyciu sieci neuronowej, który przewidzi czy dany lot zostanie opóźniony lub nie. [dane](#)

TT.NN.8

Napisz algorytm określający płeć mówcy na podstawie wyekstraktowanych parametrów mowy. Metoda powinna być oparta o sieć neuronową. [dane](#)

TT.NN.9

Na podstawie danych pochodzących ze spisu statystycznego, zaproponuj algorytm, który w oparciu o stworzoną sieć neuronową przewiduje czy dochód danej osoby przewyższa lub nie wartość \$50'000. [dane](#)

TT.P.1

Zadanie dotyczy ruchu gońca (skoczka) po planszy do gry w szachy. Goniec startuje ze skrajnego rogu planszy. Zaproponuj algorytm, który wskaże ścieżkę gońca w taki sposób, że każdy z kwadratów tworzących planszę zostanie odwiedzony dokładnie jeden raz.

Część PW – prowadzący Paweł Wawrzyński

PW. Zasady i zalecenia

- Do 6 czerwca możliwe jest skonsultowanie wstępnej wersji projektu. Po tej dacie projekt może być tylko oddany.
- Terminem na oddanie projektu jest 13 czerwca. Oddanie projektu po tej dacie będzie uzależnione od dostępności prowadzącego a ocena za taki projekt będzie niższa o 10pkt.
- Zalecane są przynajmniej dwukrotne konsultacje przed ostatecznym oddaniem projektu: konsultacje wstępne, których celem jest sprecyzowanie zadania i konsultacje przed-koncowe, z prawie działającym programem, których celem jest szczegółowe ustalenie wymagań i zakresu dokumentacji. Projekty robiące co innego niż mają robić są punktowane bardzo nisko.
- Wymagania wobec programu obejmują wyraźne oddzielenie algorytmu ewolucyjnego/grającego/itd. od rozwiązywanego problemu. Struktura programu powinna umożliwiać zarówno zastosowanie algorytmu do innego problemu jak i innego algorytmu do danego problemu. W obiektowym języku programowania oznacza to, że algorytm ma działać na abstrakcyjnych interfejsach.

PW.AE.1. Optymalne przyspieszenie żużlowca

Opis: Żużlowiec porusza się po okrągłym torze o szerokości 10m i zewnętrznym promieniu 40m. W każdej chwili jego przyspieszenie wynosi

$$a = a_0 + B v + c x$$

w układzie współrzędnych, którego oś x pokrywa się z promieniem, na którym znajduje się żużlowiec, x to odległość między jego pozycją a środkiem toru, zaś v to prędkość w tym układzie współrzędnych; a_0 oraz c to wektory, zaś B jest macierzą. W sumie na a_0 , B, c składa się 8 parametrów, które powinny zostać zoptymalizowane w taki sposób, aby żużlowiec jak najszybciej przejeżdżał okążenie, lub przejeżdżał jak najwięcej przed wypadnięciem z toru. Sposób symulowania przejazdu żużlowca należy omówić z prowadzącym.

Algorytm: (1+1) równoległe

Interfejs: pokazuje postępowanie procesu optymalizacji, po czym pokazuje tor jazdy żużlowca.

PW.AE.2. Maksymalizacja funkcji z użyciem algorytmu $(\mu+\lambda)$ i programowania ewolucyjnego.

Opis: Program znajduje maksimum funkcji

$$f: [-20, 20]^n \rightarrow \mathbb{R}, \quad f(x) = \prod_{i=1}^n \cos(x_i) - 0.01 \sum_{i=1}^n x_i^2 / i$$

Algorytm: algorytmu ewolucyjny $(\mu+\lambda)$ i algorytm programowania ewolucyjnego. Należy ustalić w przybliżeniu najlepsze parametry działania obu algorytmów dla $n=10, 30, 100$.

Interfejs: Program powinien demonstrować na bieżąco populację i wykres jakości znalezionej rozwiązania w funkcji numeru pokolenia.

PW.AE.3. Minimalizacja funkcji z użyciem algorytmu (μ, λ) i równoległego (1+1).

Opis: Program znajduje minimum funkcji

$$f: [-20, 20]^n \rightarrow \mathbb{R}, \quad f(x) = \prod_{i=1}^n \sin^2(x_i) + 0.01 \sum_{i=1}^n x_i^2 / i$$

Algorytm: algorytmu ewolucyjny (μ, λ) i algorytm równoległy (1+1). Należy ustalić w

przybliżeniu najlepsze parametry działania obu algorytmów dla $n=10,30,100$.

Interfejs: Program powinien demonstrować na bieżąco populację i wykres jakości znalezionej rozwiązania w funkcji numeru pokolenia.

PW.AE.4. Problem komiwojażera, algorytm ewolucyjny.

Opis: należy wybrać N miast polskich i znaleźć najkrótszy cykl łączący je wszystkie.

Odległości między miastami można znaleźć np. pod adresem

<http://www.odleglosci.pl/tabele-odleglosci.php>

Algorytm: ewolucyjny $(\mu+\lambda)$ i (μ,λ) . Należy zaprojektować operatory krzyżowania i mutacji. Należy eksperymentalnie ustalić w przybliżeniu najlepsze parametry działania obu algorytmów.

Interfejs: pokazuje postęp procesu optymalizacji, podaje ostateczną kolejność miast.

Należy sprawdzić program dla $N=10,20,30$.

PW.AE.5. Problem komiwojażera, algorytm genetyczny.

Opis: należy wybrać N miast polskich i znaleźć najkrótszy cykl łączący je wszystkie.

Odległości między miastami można znaleźć np. pod adresem

<http://www.odleglosci.pl/tabele-odleglosci.php>

Algorytm: genetyczny $(\mu+\lambda)$ i (μ,λ) . Należy sposób kodowania osobników. Należy ustalić w przybliżeniu najlepsze parametry działania obu algorytmów.

Interfejs: pokazuje postęp procesu optymalizacji, podaje ostateczną kolejność miast.

Należy sprawdzić program dla $N=10,20,30$.

PW.AE.6. Projekt organizmu

Opis: Organizm składa się z M pól na kratce N na N . Jest on kodowany genetycznie w ten sposób, że kolejne chromosomy wyznaczają pozycje $(x$ oraz $y)$ pól zajętych przez organizm. Przystosowanie organizmu jest równe jego momentowi bezwładności podzielonemu przez liczbę jego spójnych kawałków (spójny kawałek jest zdefiniowany w ten sposób, że z każdego jego pola da się dotrzeć do każdego innego poprzez sąsiadów, przy czym sąsiedztwo „na ukos” także liczy się).

Algorytm: genetyczny $(\mu+\lambda)$ i (μ,λ) . Należy ustalić w przybliżeniu najlepsze parametry działania obu algorytmów.

Interfejs: pokazuje postęp procesu optymalizacji, pokazuje ostateczną postać organizmu.

należy sprawdzić program dla $M=20, N=8$.

PW.LR.1. Sterowanie helikopterem

Napisać grę przypominającą

<http://www.helicoptergame.net/>

w którą będzie mógł grać człowiek.

Napisać sterownik helikoptera, oparty na logice rozmytej.

PW.LR.2. Sterowanie samochodem

Napisać grę przypominającą

http://pl.y8.com/games/super_drift_2

w którą będzie mógł grać człowiek.

Napisać sterownik samochodu, oparty na logice rozmytej.

PW.W.1. Maszyna wnioskująca przez rezolucję i zaprzeczenie – porównanie strategii

Patrz: zasady ogólne, PW. Zasady i zalecenia

Napisać program, który przyjmuje:

- klauzule w rachunku predykatów pierwszego rzędu
- tezę do udowodnienia.

Następnie, przeprowadza rezolucyjne wnioskowanie przez zaprzeczenie. Wynik demonstruje graficznie. Do wyboru są trzy strategie sterowania wnioskowaniem: zbioru uzasadnień, z preferencją dla krótkich klauzul i liniowa.