Materiały organizacyjne i wskazówki

WSI - ćwiczenia 2023L

Jakub Łyskawa

27.02.2024

Organizacja

Kontakt

Prowadzący: mgr inż. Jakub Łyskawa
E-mail: jakub.lyskawa.dokt@pw.edu.pl
Strona: https://staff.elka.pw.edu.pl/~jlyskawa/
Materiały: https://staff.elka.pw.edu.pl/~jlyskawa/wsi

Oddawanie zadań

Zadanie trzeba oddać stacjonarnie.

Należy przygotować sobie repozytorium na raporty i kod. Prowadzący powinien być dodany w tym repozytorium z uprawnieniem do komentowania pull requestów. Na każde zadanie należy utworzyć brancha. W ramach oddawania należy utworzyć pull request brancha zadania do głównego brancha.

Dokumentacja powinna być w pliku .pdf, .html lub .ipynb.

Ocena

Oceniane są:

- Poprawność implementacji
- Jakość kodu
- Przeprowadzone eksperymenty
- Dokumentacja
- Wnioski
- Za każdy rozpoczęty tydzień opóźnienia utrata 20% punktów

Za prawidłowe rozwiązanie są maksymalnie 4 punkty, za raport 3.

Repozytorium

Jak używać gita?

(bardzo skrótowo)

	main branch	feature branch
git checkout -b ...	from here	created from current branch
git add, git commit	...	changed
git pull origin master	...	updated to main branch
git push	...	pushed to server

Na końcu tworzy się merge requesta(gitlab)/pull requesta (github) z feature brancha na main brancha. Jak zostanie zaakceptowany, to zmiany z feature brancha mogą zostać dołączone do main brancha.

Repozytorium

Rozwiązanie powinno być zgodne z szablonem dostępnym dla tego zadania w repozytorium `https://gitlab-stud.elka.pw.edu.pl/jlyskawa/wsi-template`.

W celu połączenia lokalnego repozytorium z szablonowym można użyć polecenia

Dodanie powiązania z repozytorim źródłowym

```
git remote add upstream https://gitlab-stud.elka.pw.edu.pl/jlyskawa/wsi-template
```

Repozytorim źródłowe będzie dostępne pod nazwą "upstream".

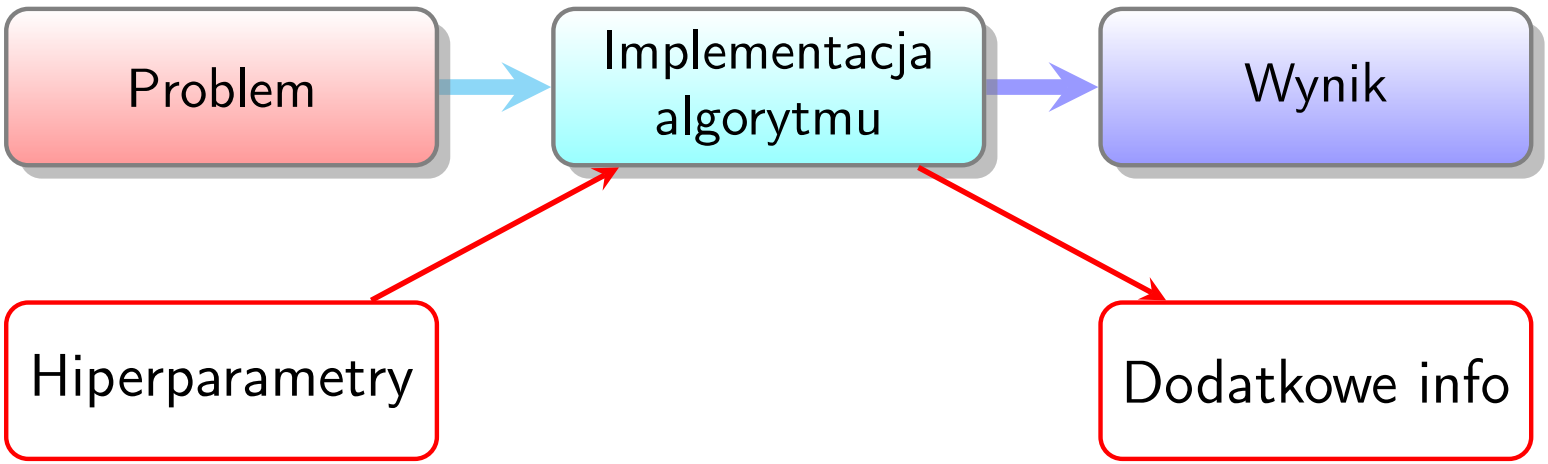
W celu zaciągnięcia najnowszych zmian z repozytorium źródłowego można użyć polecenia

Pobranie najnowszej wersji repozytorium

```
git pull upstream main
```

Implementacja

Struktura rozwiązania



Zewnętrzne pakiety

Można używać pakietów do obliczeń numerycznych, na przykład numpy oraz pakietów wskazanych w danym poleceniu (np. w celu ładowania zbioru danych w zadaniach, w których jakiś jest), a nawet jest to zalecane

Użycie innych pakietów za zgodą prowadzącego.

Do wizualizacji/opracowania wyników można używać dowolnych narzędzi.

Czego unikać

- Niewydzielona implementacja

Czego unikać

- Niewydzielona implementacja
- Implementacje pod konkretny problem

Czego unikać

- Niewydzielona implementacja
- Implementacje pod konkretny problem
- Magiczne stałe

Czego unikać

- Niewydzielona implementacja
- Implementacje pod konkretny problem
- Magiczne stałe
- Przekazywanie danych/parametrów/wyników przez zmienne globalne

Czego unikać

- Niewydzielona implementacja
- Implementacje pod konkretny problem
- Magiczne stałe
- Przekazywanie danych/parametrów/wyników przez zmienne globalne
- Kodu, który się nie odpali od początku do końca

Interfejsy

Co to?

Specyfikacja komunikacji między komponentami oprogramowania

Co to?

Specyfikacja komunikacji między komponentami oprogramowania

Czyli opis funkcjonalności (oraz tego, jak z nich korzystać), jakie

- implementacja interfejsu zapewnia
- programista wykorzystujący wie, że może użyć

Po co to?

Możliwość pracy zespołowej - równoległego tworzenia fragmentów oprogramowania

Po co to?

Możliwość łatwej podmiany na komponent implementujący inną funkcjonalność

```
solver = RandomSolver(
**params
)

result = solver.solve(
problem, pop0
)
```

```
solver = GeneticSolver(
**params
)

result = solver.solve(
problem, pop0
)
```

Po co to?

Możliwość reużywania narzędzi

```
class RandomSolver(Solver):
def solve(...)
...

class GeneticSolver(Solver):
def solve(...)
...

measure_quality(RandomSolver, problem, ...)
measure_quality(GeneticSolver, problem, ...)
```

Raport

Zawartość

Co powinien zawierać raport:

- Informacje o ewentualnych decyzjach projektowych
- Cel eksperymentów (na samym początku) i opis eksperymentów - co jest badane (np. wpływ hiperparametrów), jakie są oczekiwane wyniki (np. jak wpływa hiperparametr na wynik/zachowanie algorytmu)
- Wyniki (np. tabele i wykresy) z komentarzami
- Wnioski

Zawartość

Co powinien zawierać raport:

- Informacje o ewentualnych decyzjach projektowych
- Cel eksperymentów (na samym początku) i opis eksperymentów - co jest badane (np. wpływ hiperparametrów), jakie są oczekiwane wyniki (np. jak wpływa hiperparametr na wynik/zachowanie algorytmu)
- Wyniki (np. tabele i wykresy) z komentarzami
- Wnioski

Czego nie powinien zawierać raport:

- Kodu (wyjątkiem są pliki .ipynb)
- Skopiowanej treści zadania/teorii
- Dokumentacji kodu

Przykładowa prezentacja wyników

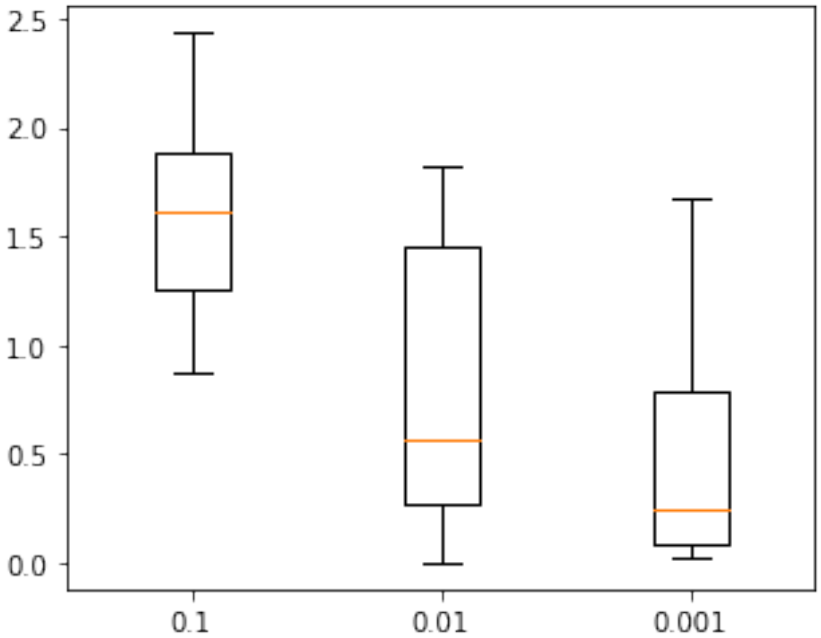
Tabele:

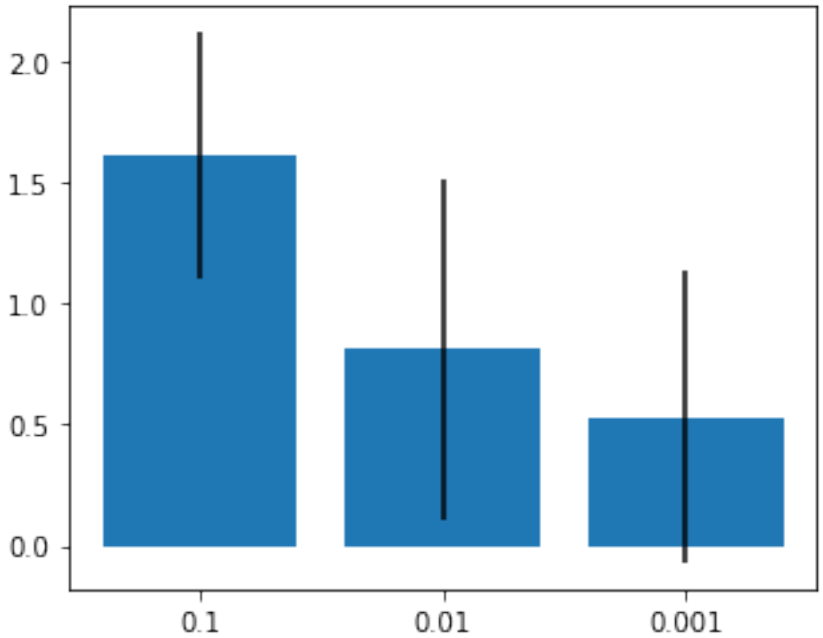
β	wynik	liczba kroków
0.1	0.23 ± 0.05	120 ± 30
...	...	...

$\alpha \backslash \beta$	0.1	0.01
0.1	0.23 ± 0.05	0.32 ± 0.07
0.01	...	...

Przykładowa prezentacja wyników

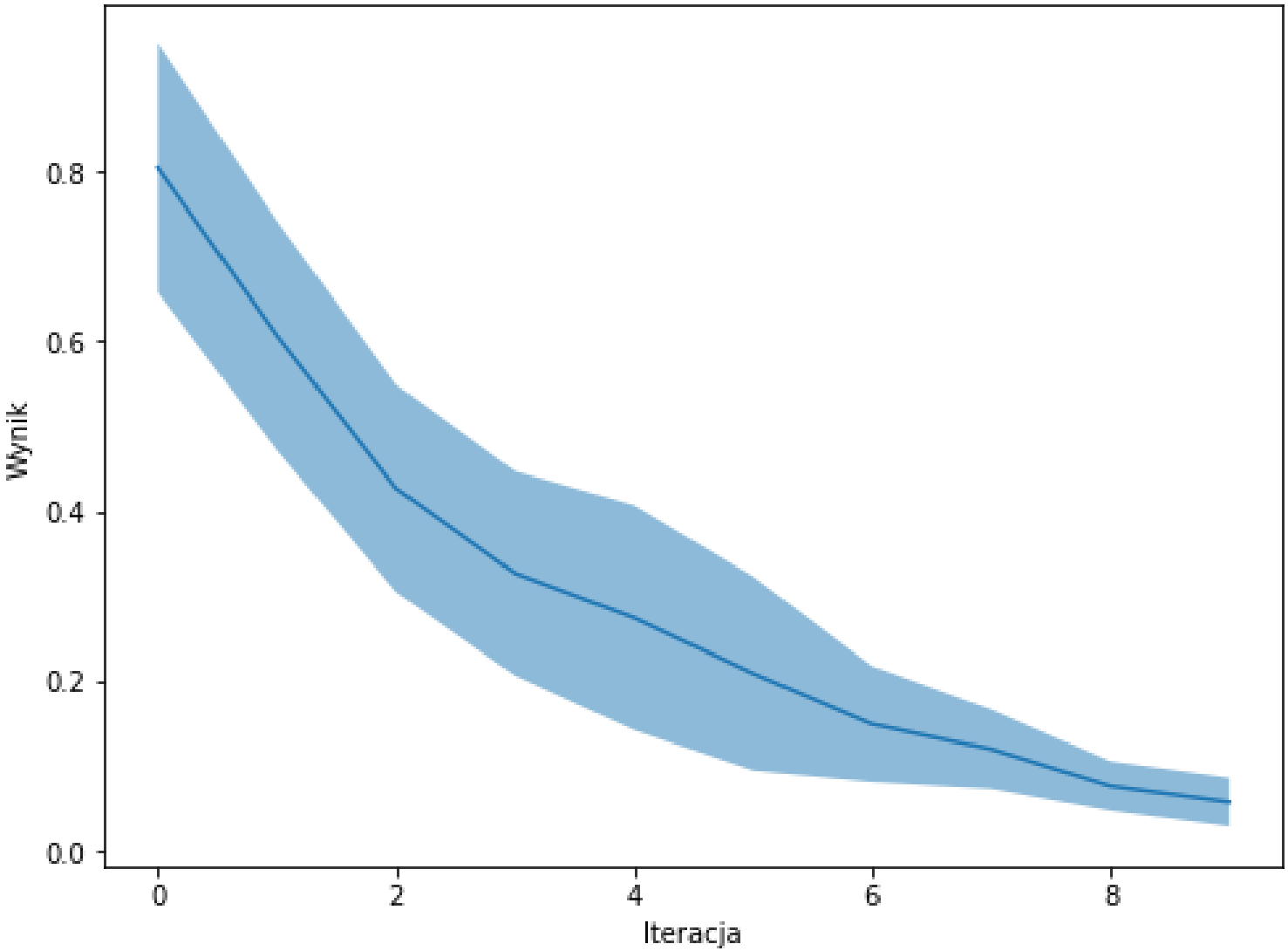
Wykresy:





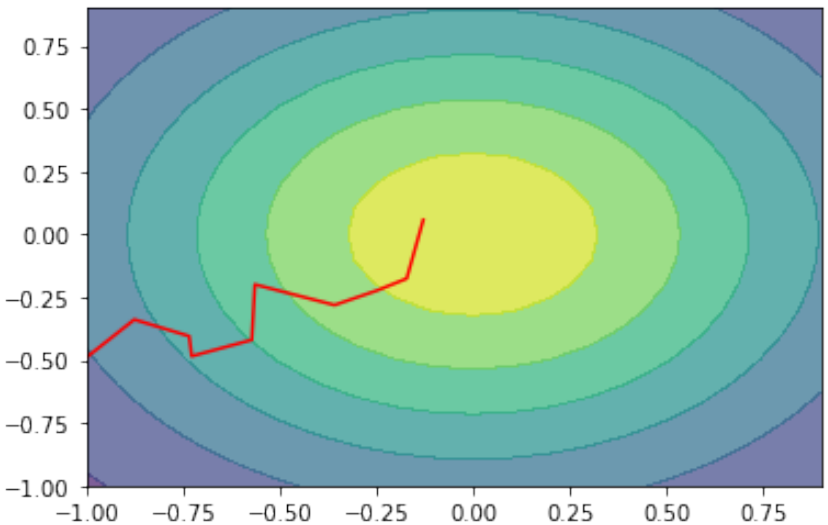
Przykładowa prezentacja wyników

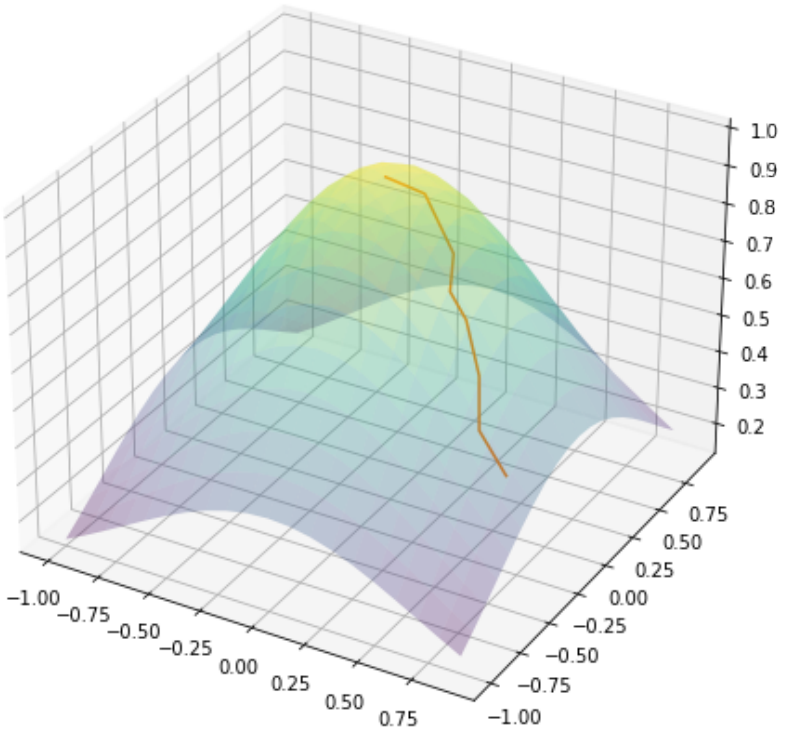
Wykresy w czasie:



Przykładowa prezentacja wyników

Wykresy przedstawiające dane 3D:





Jak przedstawiać wyniki

Jeżeli jest losowość, wyniki powinny dotyczyć wielu uruchomień (raportować np. średnią i odchylenie standardowe).

Jak przedstawiać wyniki

Jeżeli jest losowość, wyniki powinny dotyczyć wielu uruchomień (raportować np. średnią i odchylenie standardowe).

Figury i tabele powinny być podpisane i powinien być do nich odnośnik w tekście raportu.

Jak przedstawiać wyniki

Jeżeli jest losowość, wyniki powinny dotyczyć wielu uruchomień (raportować np. średnią i odchylenie standardowe).

Figury i tabele powinny być podpisane i powinien być do nich odnośnik w tekście raportu.

Przed zbadaniem wpływu parametru należy znaleźć zestaw, który działa.