

# Praca domowa nr 1

## Rozwiązywanie problemów za pomocą bioinspirowanych algorytmów metaheurystycznych (AG, PSO, ACO)

### Zadanie

Celem pracy domowej jest wybranie ciekawego problemu i rozwiązanie go za pomocą różnych poznanych dotychczas algorytmów (algorytm genetyczny, strategię inteligentnego roju). Całość należy umieścić w notebooku jupyterowym, który w przejrzysty sposób będzie prezentował rozwiązanie. Notebook powinien zawierać wstawki kodu, ich outputy, Twoje komentarze do kodu i wyników. Notebook powinien być sensownie podzielony na rozdziały: wstęp z prezentacją problemu, rozdziały z eksperymentami, podsumowanie i bibliografię.

Co należy zrobić?

- 1) **Wybieramy problem do rozwiązania.** Dobrym pomysłem jest wybranie problemu NP-zupełnego (czyli takiego, dla którego nie ma szybkich algorytmów rozwiązujących). Lista problemów jest dostępna tutaj:

[https://en.wikipedia.org/wiki/List\\_of\\_NP-complete\\_problems](https://en.wikipedia.org/wiki/List_of_NP-complete_problems)

Można wybrać i problem spoza listy. Przed dokonaniem wyboru, warto się zastanowić, czy uda nam w ogóle rozwiązać algorytm za pomocą AG/PSO/ACO. Można przejrzeć źródła internetowe i artykuły naukowe, w których być może ktoś proponował rozwiązania.

W razie niepewności, wybór tematu można skonsultować z prowadzącym zajęcia.

- 2) **Potwierdzenie i rezerwacja wybranego problemu.** W obrębie grupy laboratoryjnej, każdy powinien mieć unikalny temat. Po wyborze tematu, zarezerwuj go publicznie, tak by inni wiedzieli, że temat jest zajęty. Robimy to, dodając komentarz w konwersacji na Teams, założonej przez prowadzącego zajęcia. Komentarz powinien zawierać nazwę problemu. Obowiązuje zasada: kto pierwszy ten lepszy.

W szczególnym przypadku prowadzący może pozwolić na zduplikowany temat, jeśli osoby mają inne pomysły na jego rozwiązanie.

- 3) **Przygotowujemy instancje problemu do rozwiązania.** Nie zależnie od wyboru tematu, należy przygotować kilka inputów problemu do rozwiązania. Proponuję przygotować 9 inputów:
  - Trzy inputy małe (np. grafy o kilku wierzchołkach, łamigłówki o rozmiarze 5x5)
  - Trzy inputy średnie (np. grafy o parunastu wierzchołkach, łamigłówki 10x10)
  - Trzy inputy duże (np. grafy o parudziesięciu wierzchołkach, łamigłówki 15x15 czy 20x20).

Oczywiście powyżej podano tylko przykłady. Twój problem może być zupełnie inny, i wielkości inputów musisz odpowiednio dopasować do jego trudności. Możesz przygotować więcej niż 9 inputów, ale 9 to minimum, poniżej którego nie schodzimy.

Inputy można stworzyć ręcznie lub znaleźć w internecie (podając źródło do strony).

Stworzone inputy kodujemy w Pythonie.

- 4) **Tworzymy model algorytmu genetycznego.** Dla naszego problemu należy stworzyć algorytm genetyczny rozwiązujący go. Korzystamy z paczki pygad (chyba, że z jakichś przyczyn chcemy skorzystać z innego narzędzia). Tutaj trzeba zwrócić uwagę na następujące aspekty:
- Jak kodowane będą rozwiązania jako chromosomy? Należy to objaśnić w notebooku.
  - Jaka będzie funkcja fitness? Jaki ma zakres? Jak działa? Tutaj też dopisujemy komentarz objaśniający. Można ją uruchomić dla kilku sztucznie stworzonych rozwiązań.
  - Czy w moim problemie trzeba wprowadzić niestandardowe krzyżowanie/mutację? Jeśli tak, to opisujemy.
  - Dopasowanie wielkości populacji do skomplikowania problemu. Powinna być na tyle duża, żeby problem się rozwiązywał, i na tyle mała, żeby rozwiązanie było szybko znajdowane.
  - Można też poeksperymentować z innymi parametrami (np. selekcja, mutacja).
  - Podaj lub zilustruj, jak wyglądają rozwiązania algorytmu genetycznego dla różnych inputów.

- 5) **Sprawdź efektywność algorytmu genetycznego.** Chcemy uzyskać odpowiedź na dwa pytania:

- Czy algorytm genetyczny w ogóle rozwiązuje problem?  
(Jeśli mamy problem optymalizacyjny i szukamy najlepszego rozwiązania, to warto wiedzieć, jakie jest najlepsze już w trakcie wyboru inputu problemu).
- Jeśli algorytm genetyczny rozwiązuje problem, to w jakim czasie?

Powyższe dwa pytania można zbadać następująco:

- Uruchom algorytm genetyczny sto razy, wybierając losowo za każdym razem jeden z inputów małych. Zlicz ile spośród tych 100 prób dało idealne rozwiązania (np. 92/100 czyli 92%). Następnie dla prób z idealnym rozwiązaniem (u nas 92), podaj jaki mają średni czas wykonania. Parametry algorytmu genetycznego (np. populacja) muszą być odpowiednio dopasowane do trudności problemu.
- Uruchom 100 razy AG dla inputów średnich. Powtórz eksperyment, jak wyżej. Pamiętaj o odpowiedniej konfiguracji AG (np. wielkość populacji).
- Powtórz 100 razy eksperyment dla inputów dużych.
- Wszystkie wyniki zestaw w tabelce i na sensownym wykresie.
- Zinterpretuj wyniki w komentarzu. Czy AG działa dobrze? Żle?

- 6) **Wykonaj jeszcze jeden eksperyment** (tak jak w powyższych punktach 4 i 5) z innym modelem/algorytmem. Do wyboru (jeden z poniższych):

- Inny model algorytmu genetycznego (inne chromosomy i inna funkcja fitness).
- Particle Swarm Optimizer (Rój cząstek).
- Ant Colony Optimizer (Kolonja mrówek).

Dla wybranego algorytmu/modelu należy powtórzyć badanie efektywności (tak jak w pkt 5). Następnie porównać efektywność drugiego algorytmu z algorytmem genetycznym z poprzedniego eksperymentu. Dodaj interpretację/komentarz.

## Terminy i ocenianie

Czas na zrobienie zadania (termin oddania) zostanie napisany przez prowadzącego na Teams. Z uwagi na skomplikowanie zadania, będzie minimum 2 tygodnie czasu na rozwiązanie go.

Jeśli prowadzący stworzy zadanie na Teams, z możliwością załączenia plików, to należy załączyć notebooki Jupyter w podanym terminie. Jeśli to możliwe, to proszę załączyć też wersję HTML lub PDF notebooka.

Praca domowa oceniana jest na **maks 5 punktów**. Oceniane będą takie aspekty jak:

- Czy problem jest odpowiednio trudny/ciekawy?
- Czy sprawozdanie jest przejrzyste i kompletne?
- Czy przygotowaliśmy odpowiednią liczbę inputów?
- Czy wykonaliśmy kompletny eksperyment (efektywność+czas)?
- Czy AG został zdefiniowany sensownie?
- Czy drugi model/algorytm został uwzględniony w projekcie i czy został zaprojektowany sensownie?

Uwaga! Nawet jeśli stworzyliśmy dość sensowny algorytm genetyczny i testowaliśmy go na różne sposoby, może się zdarzyć, że nie będzie on efektywnie rozwiązywał naszego problemu. Taki model można uwzględnić w sprawozdaniu, ale i tak zachęcam do dość wnikliwej analizy co działa, a co nie, i ewentualnych testów innych modeli/algorytmów.

Praca domowa może być oceniona na trzy sposoby. Dla części osób PD może być sprawdzona jednym sposobem, dla części innym.

- Notebook będzie pobrany przez prowadzącego zajęcia i sprawdzony zdalnie/osobiście.
- Prowadzący poprosi o odpalenie notebooka na zajęciach i krótką prezentację pracy domowej.
- Prowadzący poprosi o prezentację notebooka na projektorze i prezentację rozwiązania przed całą grupą (w przypadku wybranych osób, lub szczególnie ciekawych prac).