Metaopymalizacja algorytmów wielopopulacyjnych

Autor:
Filip Katulski

Etap 1: State of the Art - obecny stan wiedzy

Poniższe artykuły i prace stanowiły źródło informacji dla tego projektu i używane w nich definicje i pojęcia służą za źródło informacji.

1. *What Can We Learn from Multi-Objective Meta-Optimization of Evolutionary Algorithms in Continuous Domains?* - Roberto Ugolotti, Laura Sani and Stefano Cagnoni

Wnioski:

- Algorytmy Ewolucyjne (EA) w swoich podstawowych ustawieniach mogą znajdować optima lokalne, jednak dopiero tuning parametrów może przybliżyć je do optimum globalnego.
- W artykule poruszono zagadnienia odnośnie dobierania parametrów Algorytmów Ewolucyjnych takie jak: natura problemu, jego ograniczenia, czas obliczeniowy.
- Użyty framework: EMOPaT - *Evolutionary Multi-Objective Parameter Tuning*.
- W pracy przedstawiono kilka wersji problemu badawczego w następujących wariantach:
    - Multi-Objective Single-Function Optimization Under Different Constraints - optymalizacja pojedynczej funkcji w ramach różnych warunków ograniczających
    - Multi-Function Optimization - wszystkie funkcje opisane w problemie badawczym są optymalizowane razem.

2. *Meta-optimization of multi-objective population-based algorithms using multi-objective performance metrics* - Krystian Łapa

Wnioski:

- W niniejszym artykule wybrano algorytm genetyczny (GA) do optymalizacji parametrów rozważanego algorytmu wielokryterialnego (metaoptymalizacja)
- Algorytmy populacyjne (population-based algorithms) to algorytmy oparte o populację rozwiązań, zwanych indywiduami (individuals). Mogą (lecz nie muszą) być oparte o działanie populacji prawdziwych, występujących w naturze i środowisku ludzkim.
- Artykuł opisuje nową metodę optymalizacji parametrów algorytmów populacyjnych. W zaproponowanej metodzie parametry algorytmu są wyznaczane przez funkcje zmieniające się od iteracji (iteration-dependent function). Funkcje te zwracają różne wartości parametrów w zależności od iteracji algorytmu i mogą mieć inny kształt zdefiniowany przez ich parametry.
- Optymalizować należy nie tylko parametry całkowite (np. liczba osobników w populacji), ale także parametry wartości rzeczywistej (np. parametry operatorów).

3. *Meta-optimization of massively concurrent optimization algorithms using Elixir technology* - Grzegorz Wcisło, praca dyplomowa

Wnioski:

- Metaoptymalizacja to proces wykorzystywania algorytmu optymalizacji do dostrojenia parametrów innego algorytmu optymalizacji.
- Istniejące metody metaoptymalizacji, np.: CALIBRA*, CMA-ES, BOBYQA, MADS, ParamILS, Irace.
- MEOW jako workbench do uruchamiania algorytmów genetycznych, napisany w Elixirze, LEAP oraz DEAP napisane w Pythonie oraz EASEA napisany w C++ mogą posłużyć do uruchamiania algorytmów genetycznych, których parametry mają być docelowo zoptymalizowane.

4. *A Platform for Testing Multi-Population Evolutionary Algorithms using The BEAM Virtual Machine* - Jonatan Kłosko, Mateusz Benecki

Wnioski:

- Większość frameworków łączy to, że operacje ewolucyjne są stosowane oddzielnie do każdej osoby lub małej grupy osób. Takie podejście ma istotne ograniczenie - uniemożliwia zastosowanie bardziej wydajnej reprezentacji populacji i zastosowanie operacji na populacji w jednym kroku.
- Meow jest deklaratywny - algorytm można opisać w formie serii operacji jaką przechodzi populacja - pipeline.
- Meow może być uruchamiany przy pomocy CPU, GPU, Klastrze

Etap 2 - Analiza i wybór narzędzi

1. Frameworki ewolucyjne, wielopopulacyjne:

LEAP - Najbardziej popularny framework ewolucyjny, ogólne przeznaczenie.
DEAP - W przeciwieństwie do LEAP, algorytmy nie są predefiniowane, należy zaprojektować samemu, wykorzystując framework.

2. Frameworki optymalizacyjne:

Optuna - Framework optymalizacyjny ogólnego zastosowania, wykorzystywany głownie do ML, jednak ma implementację “Nondominated Sorting Genetic Algorythm II” jako algorytm próbkujący (czyli optymalizujący). Dodatkowo ma wbudowany moduł wizualizacji.

pymoo - wieloobiektowy framework optymalizacyjny, zawiera różnorodne algorytmy ewolucyjne (tzw. constrained single-, multi-, and many-objective optimization algorithms).

3. Źródła informacji:

DEAP:
- https://deap.readthedocs.io/en/master/overview.html

LEAP:
- https://leap-gmu.readthedocs.io/en/latest/index.html

Optuna:
- https://optuna.readthedocs.io/en/stable/index.html
- https://www.analyticsvidhya.com/blog/2021/09/optimize-your-optimizations-using-optuna/

pymoo:
- https://www.researchgate.net/publication/340976217_Pymoo_Multi-Objective_Optimization_in_Python
- https://pymoo.org/index.html

4. Dodatkowe frameworki warte uwagi:

- PyGMO: https://esa.github.io/pygmo2
- jMetalPy: https://jmetal.github.io/jMetalPy/index.html
- Platypus: https://platypus.readthedocs.io/en/latest/experimenter.html