Visual Runner

Maciej Rewera

2018-03-07

Abstract

W dokumencie zebrano oraz omówiono wyniki prac realizowanych w trakcie projektu Visual Runner. Projekt polegał na rozwoju narzędzia do wizualizacji przebiegu algorytmu EMAS.

Contents

1 Opis zadania

Narzędzia do prezentacji przebiegu algorytmu EMAS, zwane dalej wizualizatorem.

2 Opis stanu wyjściowego

Wizualizator składa się z zestawu wykresów prezentujących wybrane wskaźniki, jak np. rozkład populacji, HVR, IGDPlus. Za spięcie wszystkich wykresów w całość i umieszczenie ich w jednym oknie odpowiada klasa ChartWrapper. W celu aktualizacji wyświetlanych danych należy wywołać metodę update() na rzecz wybranej klasy, dziedziczącej po BaseChart. W praktyce, wystarczy jednak zarejestrować obiekt klasy wykresu w ChartWrapper.

3 Opis wykonania kolejnych etapów zadania

3.1 Prezentacja ilości spotkań w danej iteracji

Celem pokazania ilości spotkań w danej iteracji została zaimplementowana klasa MeetingsChart, pobierająca parametr MeetingType, celem określenia, czy ma prezentować spotkania typu "jestem lepszy", czy "nikt nie jest lepszy". Aby można było określić ilość spotkań danego typu, wprowadzono zmiany w klasie JMetal5BaseEMAS, celem zliczania spotkań obu typów dla pojedynczej iteracji (liczniki są każdorazowo zerowane).

Tym sposobem, MeetingsChart wykorzystuje referencję do obiektu klasy JMetal5BaseEMAS aby pobrać wartość wybranych liczników, zapisuje historię wyników i aktualizuje wykres (czyli obiekt klasy XYChart). Jego działanie jest analogiczne do działania np. HVRChart. MeetingsChart jest aktualizowany co *Constants.MEETINGS_FREQUENCY* iteracji głównego algorytmu.

3.2 Reorganizacja hierarchi klas typu Chart

Dotychczas, hierarchia np. dla klasy HVRChart wyglądała następująco: *BaseChart - HVRChart*

Celem jest wyodrębnienie części wspólnej z klas: *HVRChart*, *HVChart*, *IGDPlusChart*, *EvaluationHVRChart*, *EvaluationIGDPlusChart* i umieszczenie jej w klasie ProgressBaseChart.

Ponadto, w ramach zadania usunięto zduplikowaną obsługę wyświetlania danych dla wielu algorytmów i dla domyślnego algorytmu.

3.3 Dodanie możliwości obserwowania przebiegu algorytmu dla wielu problemów

Przy pomocy *JMetal5EMASVisualExperimentRunner* jest możliwe uruchomienie wielu algorytmów, dla jednego lub wielu problemów, jednakże wszystkie wyniki są wyświetlane w jednym panelu. Celem tego zadania było utworzenie nowego runner-a i panelu, pozwalającego na wyświetlanie przebiegu algorytmów dla każdego z problemów w oddzielnej zakładce.

W tym celu utworzono klasę *JMetal5EMASMultiProblemExperimentVisualRunner* zawierającą metodę *main()* oraz metody pomocnicze, które odpowiadają za skonstruowanie i odpowiednie połączenie ze sobą wszystkich elementów potrzebnych do uruchomienia.

Dodatkowo, utworzono klasę *MultiTabFrame*, odpowiadającą za utworzenie okna wizualizatora z centralnym elementem będącym panelem z zakładkami, po jednej dla każdego problemu.

Konieczne okazało się również stworzenie nowej klasy *SingleProblemChartWrapper*, ponieważ dotychczasowa implementacja nie pozwalała na stworzenie oddzielnych paneli dla każdego z problemów.

Wynikiem prac jest wizualizator posiadający te same elementy co *JMetal5EMASVisualExperimentRunner*, ale pozwalający dodatkowo na monitorowanie przebiegu algorytmów dla wielu problemów, poprzez wykorzystanie zakładek. Każdy problem posiada dedykowaną zakładkę, w której wyświetlane są dane z jego przebiegu.