Magdalena Majkowska, Bartłomiej Janowski Warszawa, Listopad 2020

PSZT – Przeszukiwanie przestrzeni
Dokumentacja projektu

Link do repozytorium: https://github.com/jabartek/Circle-evolution-PSZT
Link do wyników: https://github.com/jabartek/Circle-evolution-PSZT-appenddix

Treść zadania

Temat zadania: *RB.S11 Najlepiej w środku*

Zaimplementować klasyczny algorytm ewolucyjny. W algorytmie tym dodatkowo, w każdej iteracji, należy wyznaczyć środek populacji i dla niego obliczyć wartość funkcji celu. Czy wartość tej funkcji dla środka populacji jest kiedykolwiek lepsza niż jej wartość dla najlepszego osobnika w danej iteracji?

Interpretacja

W celu realizacji zadania konieczne było wybranie konkretnego badanego problemu. Postanowiliśmy, że będzie nim poszukiwanie koła (reprezentowanego jako współrzędne środka oraz promień), które będzie najlepiej dopasowane do prostokąta/prostokątów znajdujących się w dwuwymiarowej przestrzeni. Wartość funkcji celu jest tym większa im więcej powierzchni koła pokrywa się z powierzchniami prostokątów i tym mniejsza im więcej powierzchni koła „wystaje" poza prostokąty (tutaj mnożnik 2.0).

W celu zbadania „środka populacji" przyjęliśmy 2 możliwe warianty – wyznaczanie koła, którego wymiary x, y, r odpowiadają średniej spośród wszystkich kół oraz, co prawdopodobnie bardziej zasadne, wyznaczanie koła o wymiarach x, y, r odpowiadających medianom spośród populacji. Umożliwia to obserwację, jak pozornie podobne koła w różny sposób reagują na ewolucję populacji.

Można się więc spodziewać, że, poza sytuacjami, gdzie prostokąty znajdują się bardzo blisko siebie, kołem najlepszym będzie takie, którego środek znajdzie się w prostokącie największym pod względem wielkości krótszego z boków.

Podział prac

Bartłomiej Janowski

- Kod związany z obliczeniem wartości funkcji celu
- Graficzne wyświetlanie postępu algorytmu oraz zapis reprezentacji graficznej do plików
- Wyznaczanie środkowych kół.

Magdalena Majkowska

- Implementacja algorytmu ewolucyjnego
- Zapis wyników w formie tekstowej
- Odczyt ustawień z pliku konfiguracyjnego
- Konfiguracja kompilacji
- Analiza danych

Dużą część pracy wykonywaliśmy wspólnie, korzystając z możliwości fizycznej współpracy.

Ważne decyzje projektowe

W kwestii wyboru realizacji algorytmu ewolucyjnego kierowaliśmy się sugestiami z wykładu z PSZT – m.in. właśnie dlatego początkowy program miał następującą strukturę:

- Selekcja: algorytm turniejowy
- Krzyżowanie uśredniające z losowymi wagami
- Sukcesja elitarna

Początkowo mutację realizowaliśmy rozdzielnie dla każdej ze współrzędnych, dopuszczając mutację wszystkich wartości parametrów koła jednocześnie, a także rozdzielnie traktując współrzędne x i y. Realizacja dopuszczała też niemutowanie żadnej z wartości. Pierwotnie źle zinterpretowaliśmy pojęcie elity – sukcesja w naszym algorytmie polegała na wybraniu 20% osobników z pokolenia potomnego i zastąpieniu nimi 20% z najgorszych osobników populacji rodzicielskiej.

Wartości początkowe współrzędnych środka koła były losowane z rozkładu ciągłego z całej badanej powierzchni.

Pomimo błędów merytorycznych algorytm dawał obiecujące wyniki dla początkowych problemów badawczych – pojedynczy prostokąt lub kilka prostokątów znajdujących się względem siebie bezpośrednio w osi x lub y. Skuteczność wynikała z przede wszystkim z bardzo dużych rozmiarów populacji (badaliśmy przypadki nawet po kilka tysięcy osobników) oraz z rozdzielnego mutowania współrzędnej w osi x i osi y, dzięki czemu osobnikom łatwo było trafiać w dogodnie ułożone prostokąty.

Problematycznymi przypadkami były takie w których prostokąty nie znajdywały się względem siebie na osiach i początkowe wartości współrzędnych x i y były ograniczone tak aby nie obejmowały optimum globalnego. Badaliśmy przypadek, kiedy koła pojawiały się początkowo w optimum lokalnym, czyli jednym z jednych z mniejszych prostokątów dla badanej powierzchni – zaobserwowaliśmy problemy eksploracyjne naszego algorytmu przy konieczności ruchu poza pojedynczymi osiami oraz problemy z sukcesją elitarną (błędnie zrealizowaną), co w połączeniu z reprodukcją przez krzyżowanie, uniemożliwiało wytworzenie się populacji w nowo odnalezionym, „lepszym" prostokącie.

Skłoniło nas do zmiany algorytmu mutacji, w taki sposób by mutacja dotyczyła jednocześnie obu współrzędnych tj. losowaliśmy wielkość przemieszczenia oraz kąt pod jakim ma ono zajść. Nie rozwiązało to jednak problemu niemożności wytworzenia się populacji poza obszarem początkowym.

Po konsultacji z prowadzącym projekt, dokonaliśmy zmian w kodzie źródłowym zgodnie z jego wskazówkami – rezygnując z krzyżowania na rzecz reprodukcji przez mutację pojedynczego osobnika; pozostając przy selekcji turniejowej; prawidłowo implementując elitę, przy znacznym zmniejszeniu jej rozmiarów; oraz zmieniając realizację mutacji, w taki sposób, że zachodzi ona zawsze i dotyczy ona tylko jednego parametru osobnika tj., położenie lub promień.

Lista wykorzystanych narzędzi i bibliotek

Staraliśmy się ograniczyć ilość używanych bibliotek. Finalnie skorzystaliśmy z:

- SFML – dla graficznego wyświetlania postępu algorytmu oraz zapisu klatek.
- Scons – dla ułatwienia kompilacji
- ffmpeg – dla eksportu zapisanych klatek w formie animacji
- Rstudio - histogramy

Instrukcja odtworzenia wyników

Kompilacja projektu odbywa się poleceniem `scons`. Parametry symulacji określa się w pliku konfiguracyjnym, ścieżka, do którego stanowi pierwszy i jedyny argument uruchomienia programu zapisywanego do ścieżki `bin/PSZT_SFML`. Szczegółowy opis pliku konfiguracyjnego znajduje się w załączeniu wraz z używanymi przez nas konfiguracjami. Możliwe jest stworzenie zapisu symulacji w postaci filmiku – wymagane jest do tego narzędzie ffmpeg, a skrypt tworzący pojedyncze wideo również znajduje się w załączniku.

Cele i tezy przeprowadzonych badań

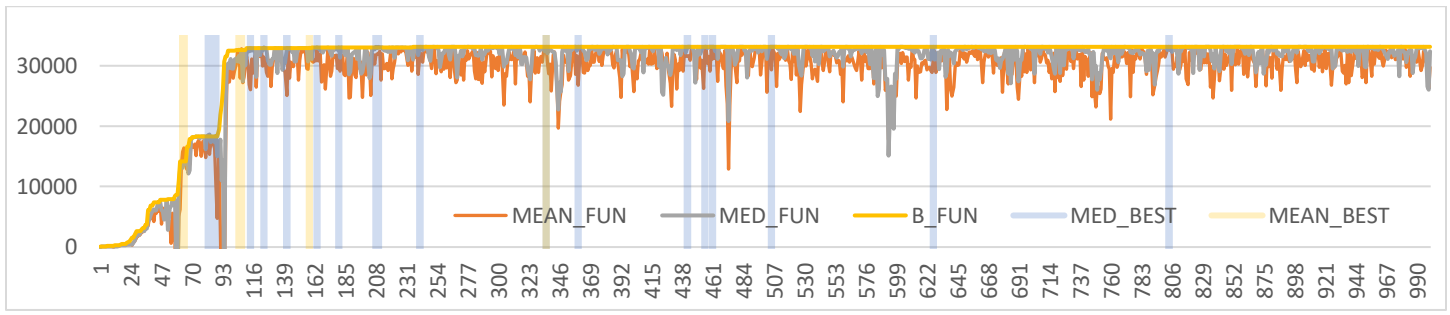
Celem jest zbadanie czy środkowy element populacji może posiadać lepszą wartość funkcji celu niż element najlepszy. Problem dobraliśmy w taki sposób, aby osobniki posiadały łatwo interpretowalne współrzędne, dla których również łatwo jest określić wartości środkowe w populacji.

Początkowo spodziewaliśmy się, że element środkowy może być lepszy pod względem wartości funkcji celu od najlepszego, w momencie, gdy populacja długotrwale „osiada" w danym optimum lokalnym. Element środkowy ma w takiej sytuacji większy potencjał na „mikro-poprawki" niż element najlepszy, jednocześnie też będąc oczywiście mniej stabilnym, jako że mutacji podlega znaczna część populacji (mała elita).

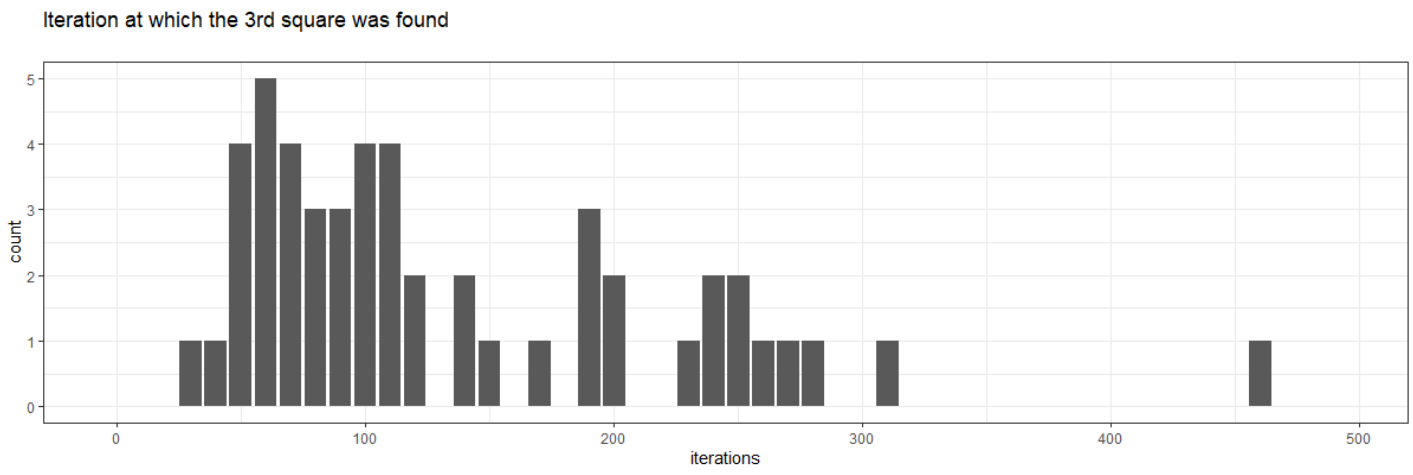
Wyniki eksperymentów

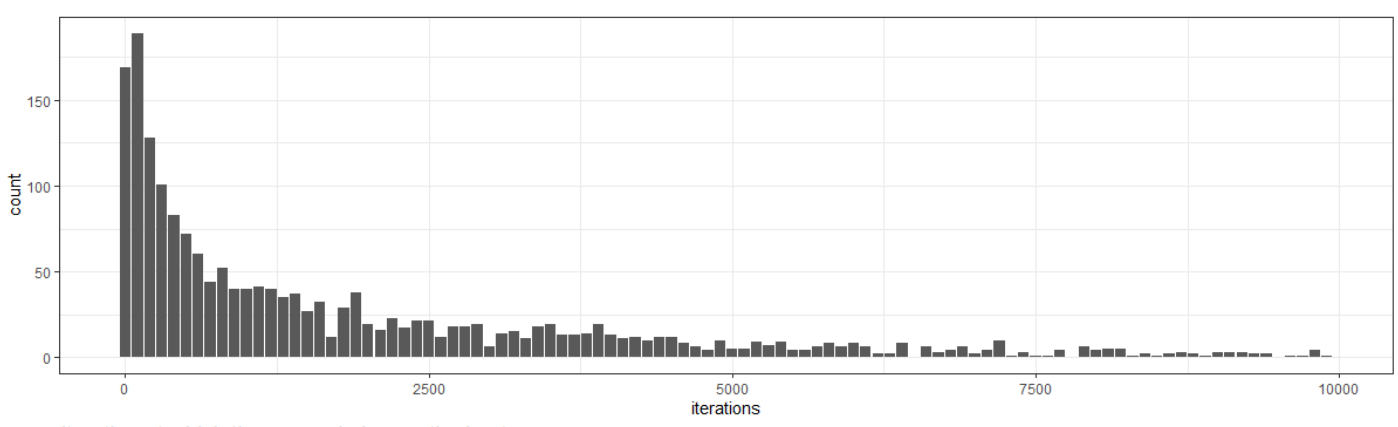
Dla każdego przypadku przygotowaliśmy wykres przedstawiający przykładowy przebieg symulacji z zaznaczonymi momentami, w których odpowiednio mediana oraz średnia są lepsze od najlepszego osobnika oraz histogramy iteracji odnalezienia okolic optimum globalnego oraz iteracji w których odpowiednio mediana i średnia są lepsze od najlepszego osobnika.

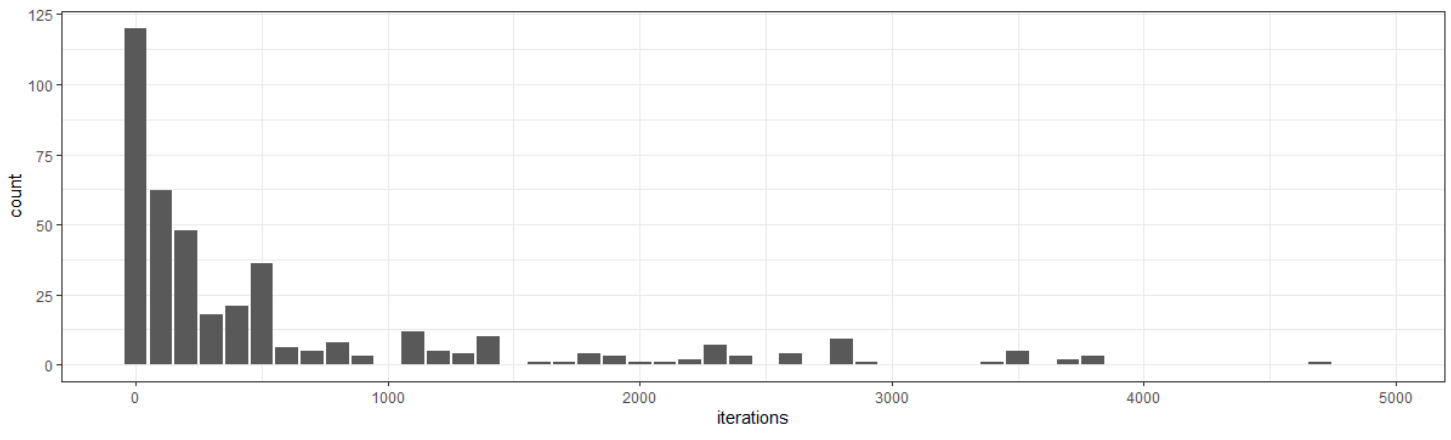
Przypadek 1.

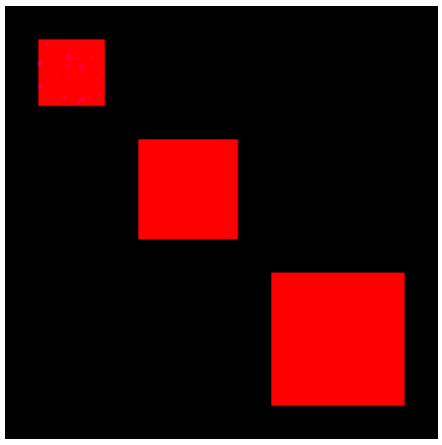


Poniższe wykresy realizowane z danych uzyskanych przy 50 uruchomieniach po 10 000 iteracji.

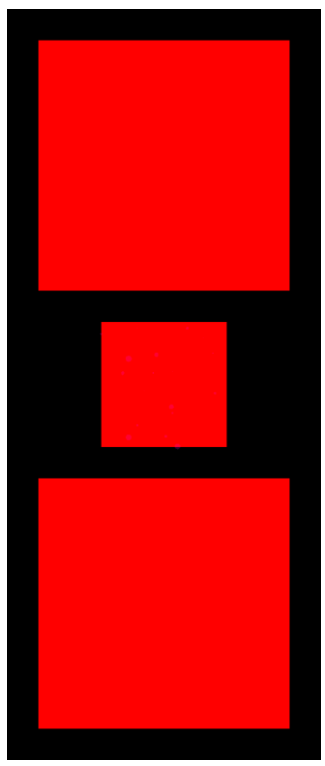




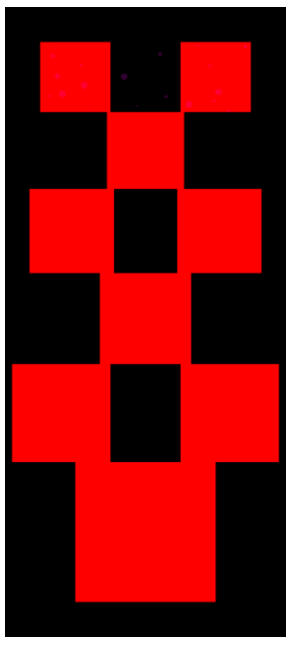




Przypadek 1.
Początkowo generujemy koła o środkach jedynie wewnątrz najmniejszego kwadratu



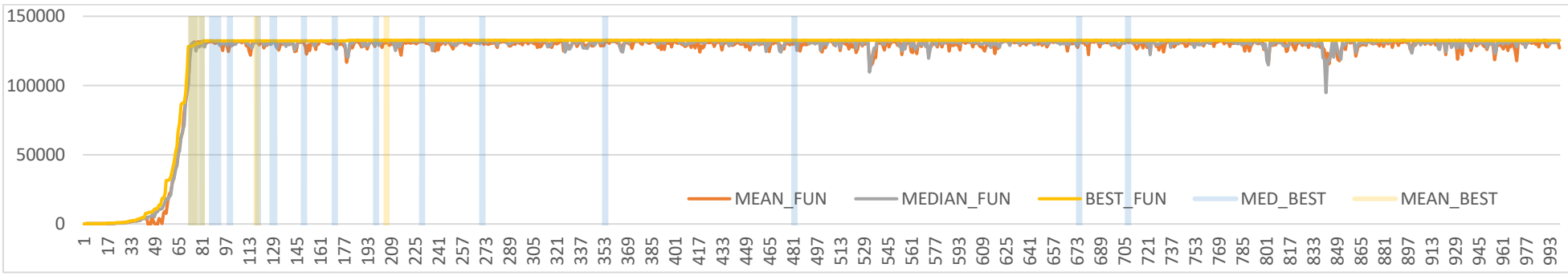
Przypadek 2.
Początkowo generujemy koła w środkowym kwadracie



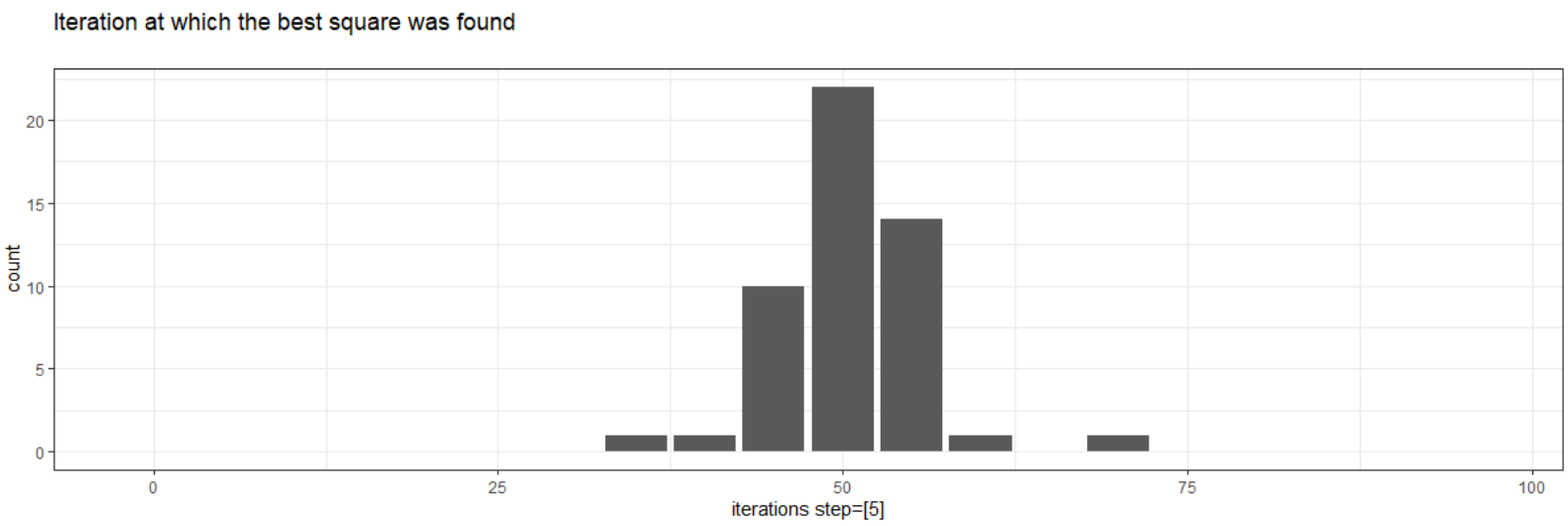
Przypadek 3.
Początkowo generujemy koła na powierzchni 2 górnych kwadratów oraz pomiędzy nimi.

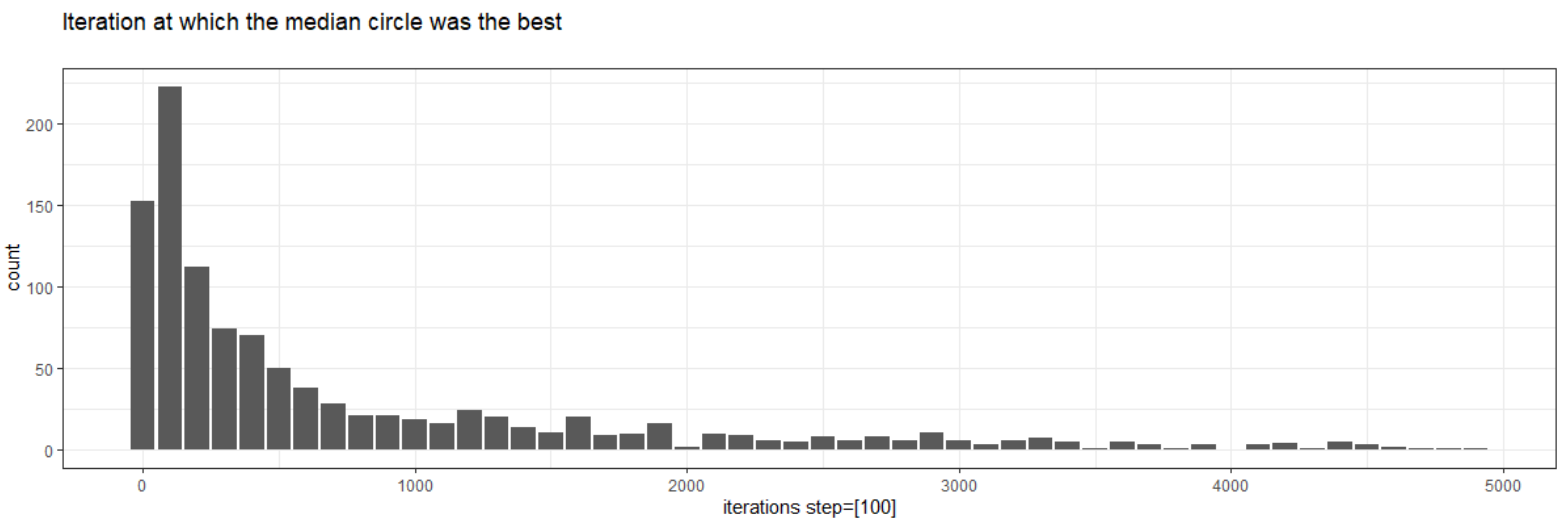
Histogram tylko dla pierwszych 5000 iteracji, bo brak wystąpień średniego koła jako najlepszego po 5000. iteracji.

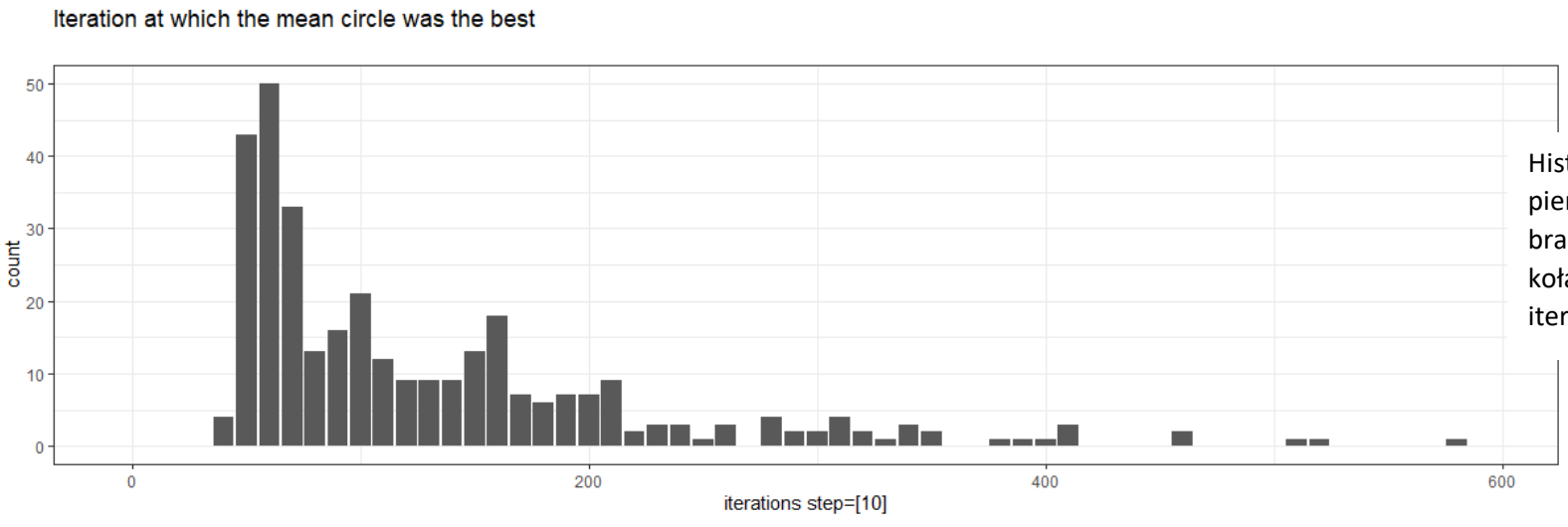
Przypadek 2



Poniższe wykresy realizowane z danych uzyskanych przy 50 uruchomieniach po 5 000 iteracji.

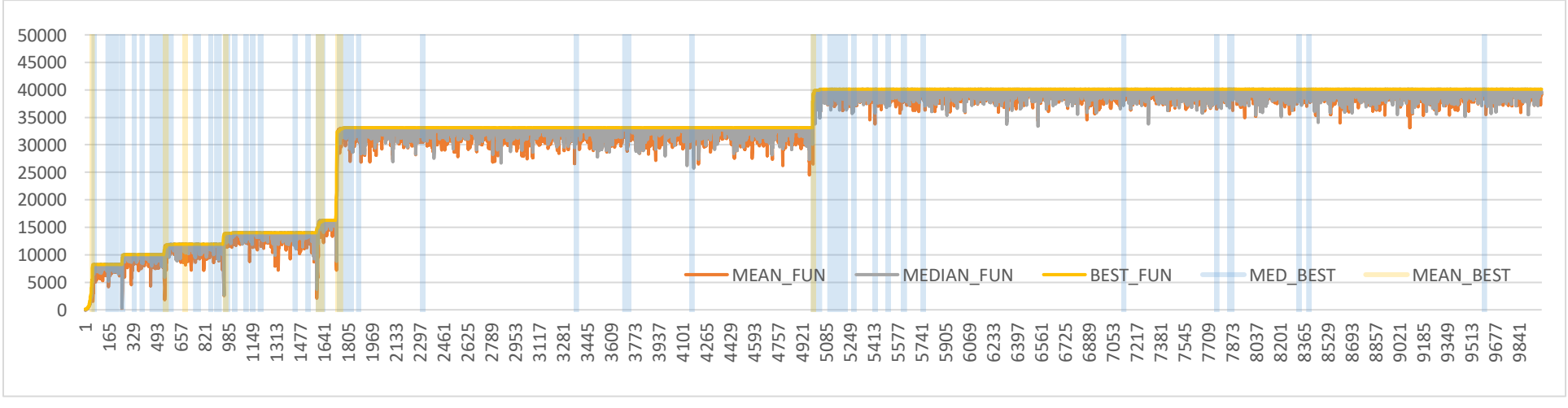


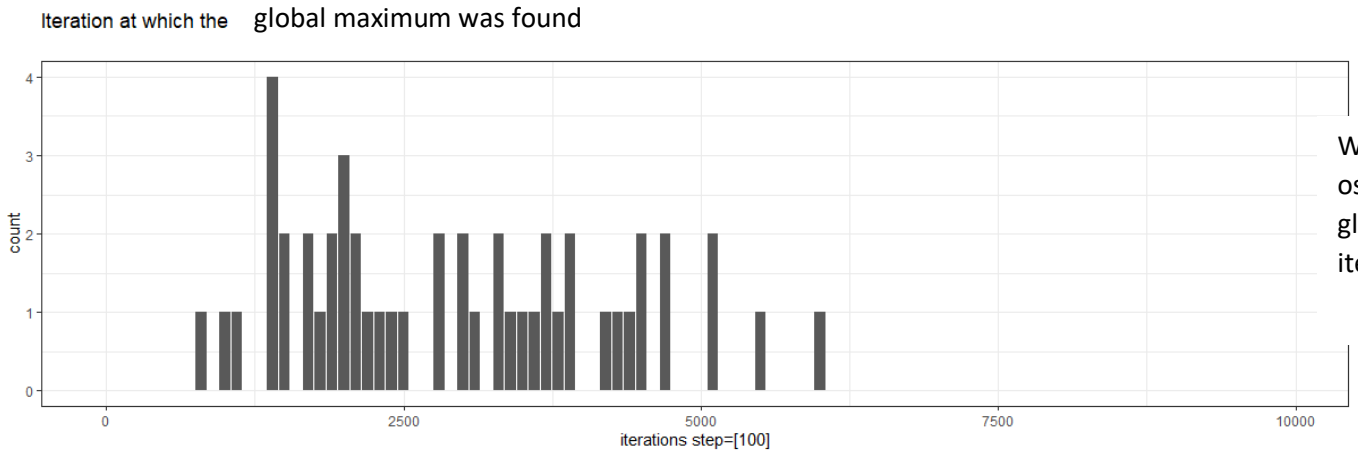




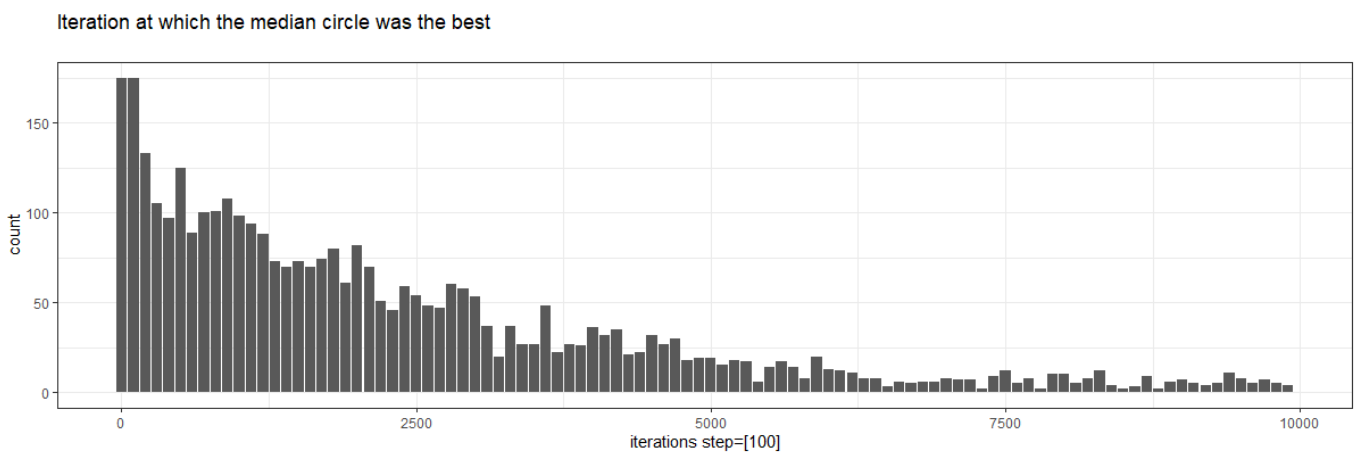
Histogram tylko dla pierwszych 600 iteracji, bo brak wystąpień średniego koła jako najlepszego po 600. iteracji.

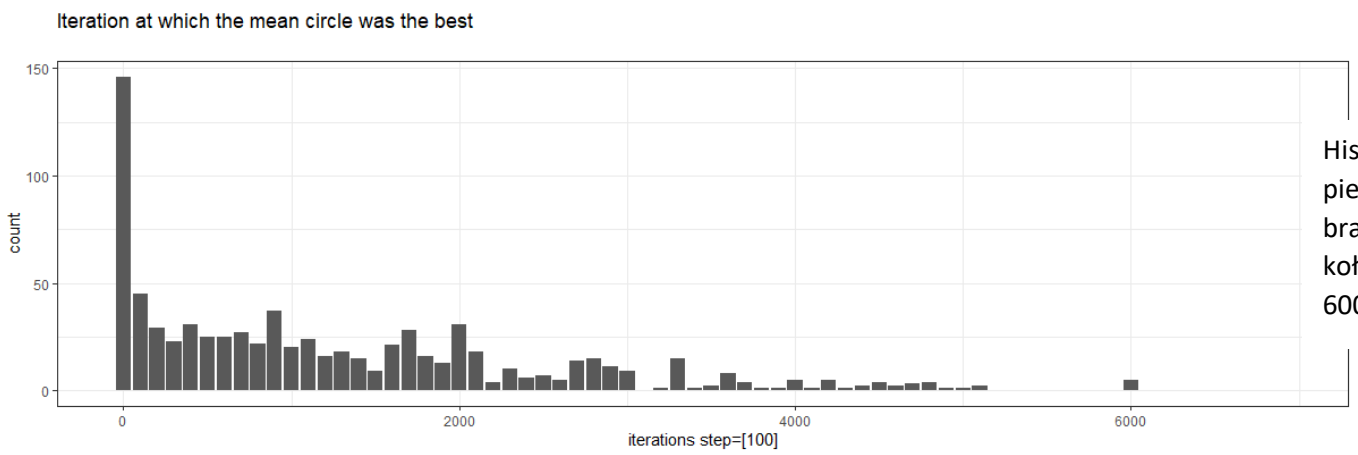
Przypadek 3.





W 1 z 50 przypadków nie osiągnięto okolic maksimum globalnego w ciągu 10000 iteracji





Histogram tylko dla pierwszych 6000 iteracji, bo brak wystąpień średniego koła jako najlepszego po 6000. iteracji.

Omówienie wyników eksperymentów

Przypadki, w których elementy środkowe, i medianowe, i średnie, są lepsze najlepszego osobnika zanikają w czasie – świadczy to o tym, że najlepsze koło wraz z kolejnymi iteracjami coraz lepiej dopasowuje się do obszaru.

Fakt, że szybszym jest zanikanie wybitnych przypadków średnich niż medianowych wiążemy z tym, że dla małej populacji (21 os.) wraz ze zwiększającą się precyzją koła najlepszego, coraz bardziej utrudnione jest mikro-dopasowywanie koła średniego (dla największej skuteczności dokładnie połowa musiałaby przemieścić się w 1. kierunku, a dokładnie połowa w drugim, etc.). W przypadku mediany problem ten jest mniej widoczny, bo dokładne wielkości mutacji wszystkich kół nie wpływają bezpośrednio na wynik.

Przypadek nr 1 był bardzo problematyczny dla pierwotnej wersji naszego algorytmu. Po poprawkach zauważamy znaczącą poprawę – wszystkie 50 uruchomień programu znajduje optimum globalne w mniej niż 500 iteracji. Jest to jednak gorszy wynik niż w przypadku nr 2, ponieważ „skoki" gwarantujące poprawę wartości funkcji celu są znacznie dłuższe oraz kąt pod jakim muszą się przemieścić jest bardziej ograniczony, ze względu na to, że kwadraty są ułożone względem siebie na przekątnej – mutacja musi być dość precyzyjna.

W przypadku 2 widzimy, że bardzo szybko kółka odnalazły optimum globalne – wynika to z tego, że kwadraty, w których znajdowało się optimum globalne zajmowały dużą powierzchnię planszy, oraz początkowa przestrzeń, w której pojawiały się koła była stosunkowo blisko. Z tego wynika, że kąt i przemieszczenie umożliwiające poprawę wartości funkcji celu nie są tak ostre jak w przypadku nr 3.

W przypadku nr 3 kwadraty od góry do dołu rosną bardzo powolnie, w związku z czym, aby osiągnąć poprawę wyniku i przejść do niższego rzędu, koło musi precyzyjnie trafić w środek kwadratu w tejże linii. Zakres kątów i przemieszczeń dających poprawę w tym przypadku jest najmniejszy, przez co jest on najtrudniejszy z tutaj zaprezentowanych, co odbija się na wynikach przedstawionych na histogramie nr. 1. Nie wszystkie uruchomienia programu odnajdują optimum globalne, ponieważ to jest jeszcze większym wyzwaniem dla algorytmu – koło optymalne zawiera w sobie dużo przestrzeni niebędącej prostokątem i skuteczna mutacja (tj. taka dająca wynik lepszy niż osiągany na największym kwadracie) jest wybitnie trudna.

Wnioski

Realizacja projektu pozwoliła nam zapoznać się z działaniem klasycznego algorytmu ewolucyjnego, a także zrozumieć znaczenie nawet niewielkich zmian parametrów na jego działanie. Badany przez nas problem, mimo że miejscami można by go uznać za wadliwy (np. przez brak gradientu funkcji celu poza bliskim sąsiedztwem prostokątów planszy) dał się zbadać algorytmem, a także pozwolił na zaobserwowanie, że, istotnie, najlepszy znajdywany w iteracji element wcale nie musi być „granicą" możliwości obecnej populacji – szczególnie gdy dopiero od niedawna odnalezionym jest nowe optimum lokalne. Spodziewamy się, że istnieją problemy, w których uwzględnienie sprawdzania elementów środkowych mogłoby istotnie wpłynąć na prędkość poszukiwania przez algorytm – w naszym przypadku badawczym jest to mimo wszystko co najwyżej „ciekawostka".