

## 1. Wstęp

W ramach projektu końcowego z przedmiotu Algorytmy Genetyczne opracowano algorytmy rozmieszczający małe prostokąty w większym w taki sposób aby wzajemnie na siebie nie nachodziły ani też nie wystawały poza obszar dużego prostokąta. Dążono do maksymalizacji powierzchni małych płyt. Poza głównym programem zaimplementowanych w języku C++ z wykorzystaniem biblioteki GaLib. Do projektu dołączony został także mini projekt, pozwalający zweryfikować poprawności otrzymanych wyników. Został on utworzony w technologii html5 i js. Znajduje się on w podkatalogu gui. Aby z niego skorzystać należy plik index.html otworzyć w jednej z nowych przeglądarek np. Chrome i przeciągnąć plik z wynikami nad wskazany obszar.

## 2. Algorytm genetyczny

Biblioteka GaLib oferuje wiele metod selekcji, reprodukcji, mutacji i krzyżowania. W ramach projektu skupiono się głównie jedynie na wyborze metody reprodukcji pozostawiając pozostałe elementy ustawione domyślnie.

- a. Prosta metoda reprodukcji – do przetwarzania przez algorytm trafia populacja wygenerowana losowo bądź zainicjalizowana przez użytkownika. Każde kolejne pokolenie to zupełnie nowy zbiór osobników wybranych z poprzedniego pokolenia
- b. Metoda Steady Stage – przed selekcją następuje przygotowanie, polegające na dodaniu do populacji pewnej ilości nowo wygenerowanych osobników. Następne pokolenie wybierane jest przez odrzucenie najgorszych osobników ze zmieszanej populacji. Selekcja oczywiście zachowuje liczebność populacji na stałym poziomie. Algorytm umożliwia ustalenie ilości osobników które będą dogenerowywane do każdego pokolenia.

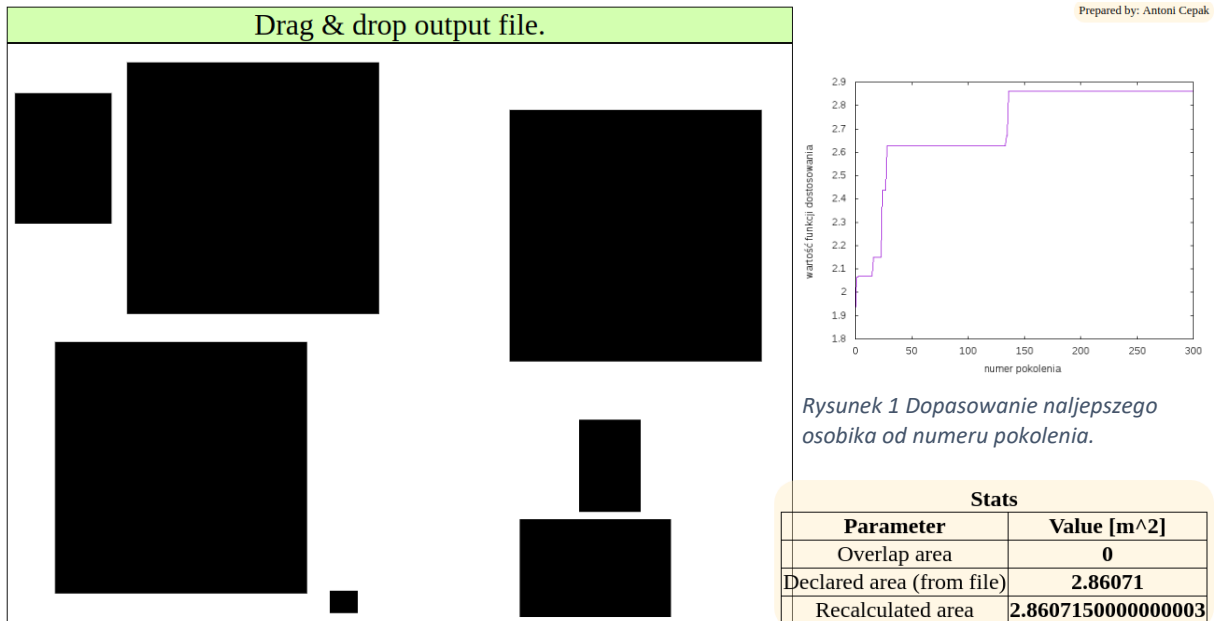
W projekcie do kodowania rozwiązania użyto klas GABin2DecPhenotype i GABin2DecGenome, pozwalają one w łatwy sposób utworzyć schemat rozwiązania przez definiowanie kolejnych części genotypu. Każda z części reprezentuje liczbę z podanego przedziału i jest reprezentowana przez ciąg bitów o zadanej długości. Im dłuższy ciąg tym bardziej szczegółowa kwantyzacja.

### 3. Wyniki i analiza

Podczas rozwiązywania problemu obrano stałe wartości dla prawdopodobieństwa mutacji krzyżowania oraz liczby pokoleń, na poziomie odpowiednio:

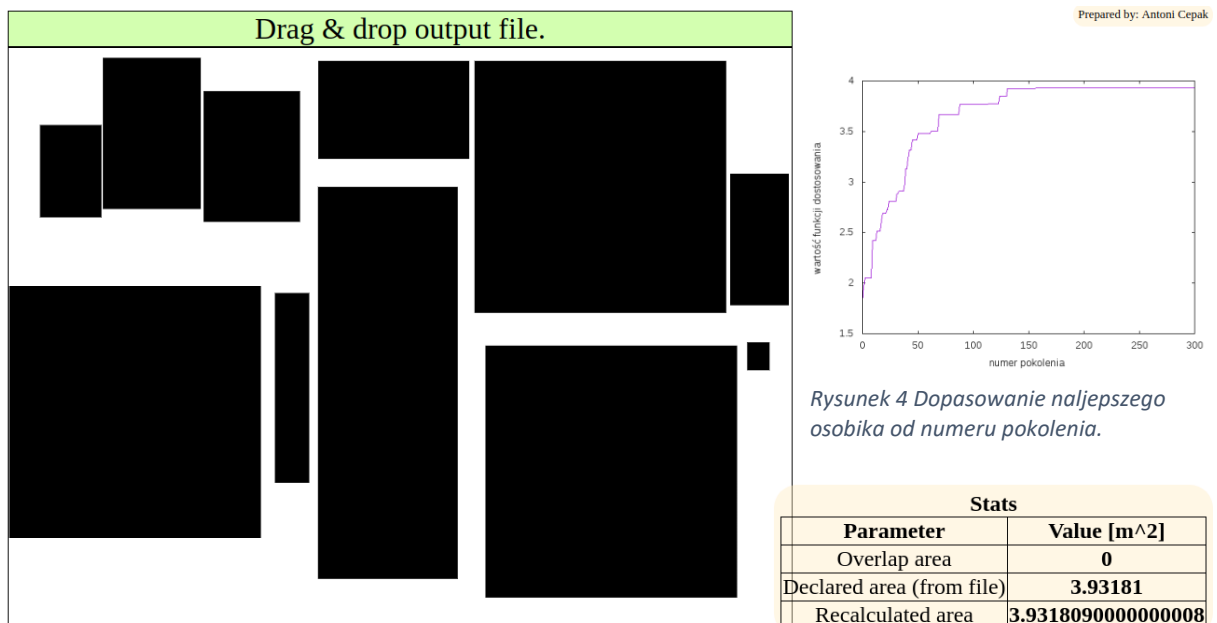
- $MUTATION\_PROB = 0.01$
- $CROSS\_PROB = 0.9$
- $GENERATION_{NUM} = 300$

#### a. prosty algorytm genetyczny - rezultaty



Rysunek 2 Rezultat uzyskany przy użyciu metody prostej metody reprodukcji, dla populacji o wielkości 10 000 osobników.

#### b. Algorytm Steady Stage - rezultaty



Rysunek 3 Wynik działania algorytmu z wykorzystaniem algorytmu GASteadyStage z współczynnikiem nowych osobników w każdym pokoleniu na poziomie 0.9.

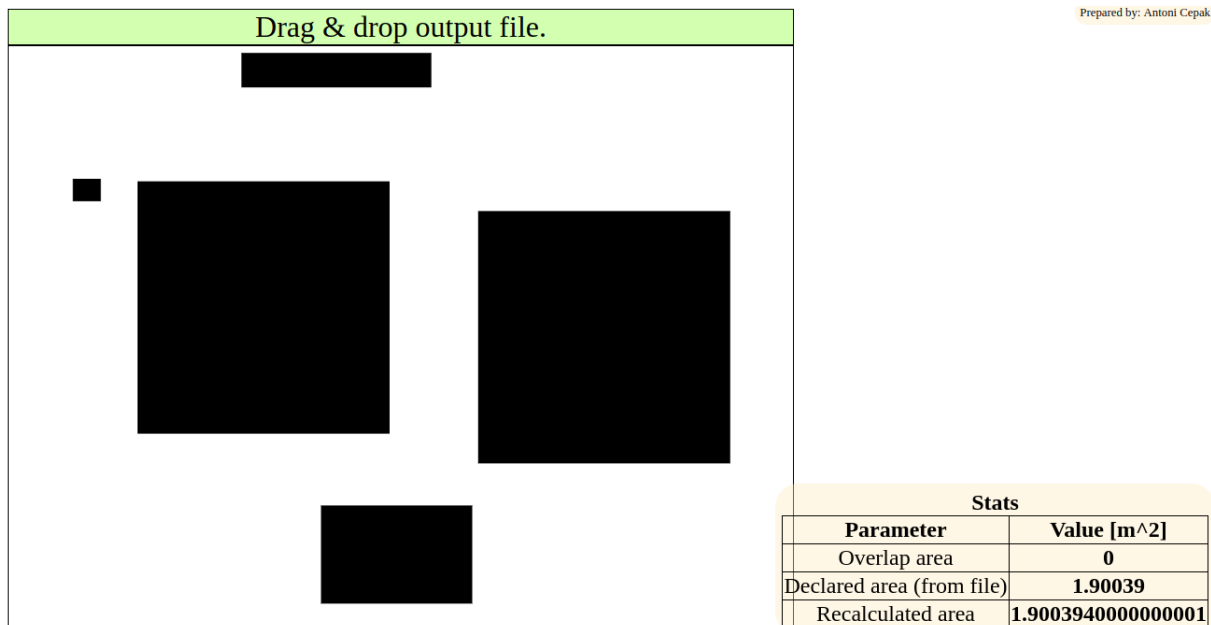
### c. losowanie rozwiązania

Zaobserwowano znaczną różnicę w efektywności powyższych metod selekcji. Znając zasadę ich działania nasuwa się pytanie, czy otrzymanie rozwiązania o wysokim dopasowaniu nie sprowadza się jedynie do wygenerowania odpowiedniej liczby osobników? W celu weryfikacji obliczono liczbę osobników wygenerowanych przy użyciu algorytmu *Steady Stage*

$$individuals = initPopSize * generationsAmount * replacementFactor + initPopSize$$

$$individuals = 10000 * 300 * 0.9 + 10000 = 2710000$$

Ze względu na ograniczone zasoby pamięci komputera zachowano jedynie rząd wielkości populacji początkowej i ustalono ją na 1000000, czyli blisko 3 krotnie mniejszą niż liczba osobników wygenerowana podczas działania algorytmu *Steady Stage*.



Rysunek 5. Rozwiązanie będące najlepszym rozwiązaniem z populacji liczącej 1 000 000.

Na rysunku 5 widzimy, że dopasowanie najlepszego osobnika wygenerowanego bez uruchomienia algorytmu genetycznego jest na bardzo niskim poziomie. Można także przewidywać, że generowanie kolejnych losowych osobników nie prowadziło by szybko do polepszenia tego rezultatu. Wniosek ten można wysnuć analizując maksymalny poziom dopasowania dla populacji o liczebności 10 000 a więc 100 krotnie mniej licznej. Z rysunku 2 i 4 możemy odczytać że plasowała się ona w okolicy 1.9.

### 4. Podsumowanie

W rozwiązaniu poruszonego problemu dużo skuteczniejszym okazał się algorytm *Steady Stage*. Dzięki ciągłemu doładowywaniu rozwiązań i włączaniu ich w kolejnych epokach, zachowano dużą różnorodność w populacji. Zweryfikowano również, że wynik ten nie jest rezultatem wylosowania odpowiednio dużej ilości rozwiązań i doboru najlepszego, ale faktycznie efektem pracy algorytmu genetycznego.