Sprawozdanie 2
Testowanie opracowanej metody heurystycznej

Mateusz Babiaczyk, Bartosz Nawrotek

2018-06-01

1 Zmiany w algorytmie

Po zaimplementowaniu algorytmu i zauważeniu jego słabych osiągnięć, zdecydowano by wprowadzić zmiany w zastosowanym podejściu algorytmicznym. Poniżej w kilku sekcjach omówiono te cześci algorytmu, które zostały zastąpione. Dodatkowo wybrane zostało kryterium stopu jako 200 iteracji algorytmu.

1.1 Kodowanie

Zmiana w kodowaniu polega na traktowaniu osobnika jako cykliczny, spowodowało to że każdy osobnik jest zbiórem rozwiązań z przesuniętym początkiem o jeden oligonukleotyd. Nie pogorszyło to złożoności obliczeniowej wyznaczania funkcji celu, która pozostała równa $O(n)$, gdzie n jest długością osobnika. Pozwoliło natomiast na traktowanie jednego rozwiązania jako zbioru n różnych rozwiązań i korzystania z tej części osobnika o najlepszym przystosowaniu.

1.2 Mutacje

W funkcji odpowiedzialnej za dokonanie mutacji osobnika została zastosowana odrębna idea ze względu na to, że rozwiązania pozostawały w minimach lokalnych. Przyjęto konwencję, w której każdy z oligonukleotydów ma wygenerowane koło fortuny. Dla koła fortuny oligonukleotydu A, każdy z oligonukleotydów otrzymuje wycinek koła w zależności od funkcji kosztu między nim a osobnikiem A (Dla kosztu maksymalnego, nie otrzymuje wycinka). Gdy następuje mutacja osobnika wybierany jest oligoukleotyd a o największym koszcie będącym sumą kosztów z sąsiednimi oligonukleotydami oraz losowany jest oligonukleotyd b przy użyciu wygenerowanego koła fortuny przed którym umieszczany zostaje oligonukleotyd a. Dzięki temu algorytm przemieszcza oligonukleotyd o najgorszym dopasowaniu w co najmniej tak dobre miejsce preferując miejsca lepszego dopasowania.

1.3 Krzyżowanie

Krzyżowanie zaczyna się w dokładnie taki sam sposób jak w pierwotnym algorytmie, a mianowicie od pewnego wylosowanego przedziału przepisuje się oligonukleotydy do nowo tworzonego osobnika (kopiuje wycinek i wkleja go do nowego osobnika) z wybranego osobnika z populacji rodzicielskiej. Następnie uzupełniany jest koniec osobnika wartościami z innego osobnika z populacji rodzicielskiej, uważając oczywiście by dany oligonukleotyd nie został powtórzony. W ten sam sposób zostaje uzupełniony początek osobnika z nowej populacji.
Wszystkie oligonukleotydy które nie zostały dodane wstawiane są, w miejsca w których ich koszt będzie najmniejszy. Motywacją było przyspierzenie zbieżności algorytmu oraz zachowanie losowości dzięki losowemu doborowi punktów krzyżowania. Tym samy dając lepsze rozwiązania w krótszym czasie przetwarzania jak i dając możliwość na opuszczenie minimów lokalnych.

2 Testy

W testach mierzono upływ czasu, jak i zmiany jakości rozwiązań co 10 iteracji algorytmu, aby pokazać poprawę rozwiązań wraz z kolejnymi iteracjami. Jako jakość rozwiązania przyjęto stosunek ilości oligonukleotydów rozwiązania uzyskanego przez algorytm do ilości oligonukleotydów w rozwiązaniu optymalnym. Jakość wyrażamy w procentach.

2.1 Wyniki algorytmu dla błędów negatywnych, losowych

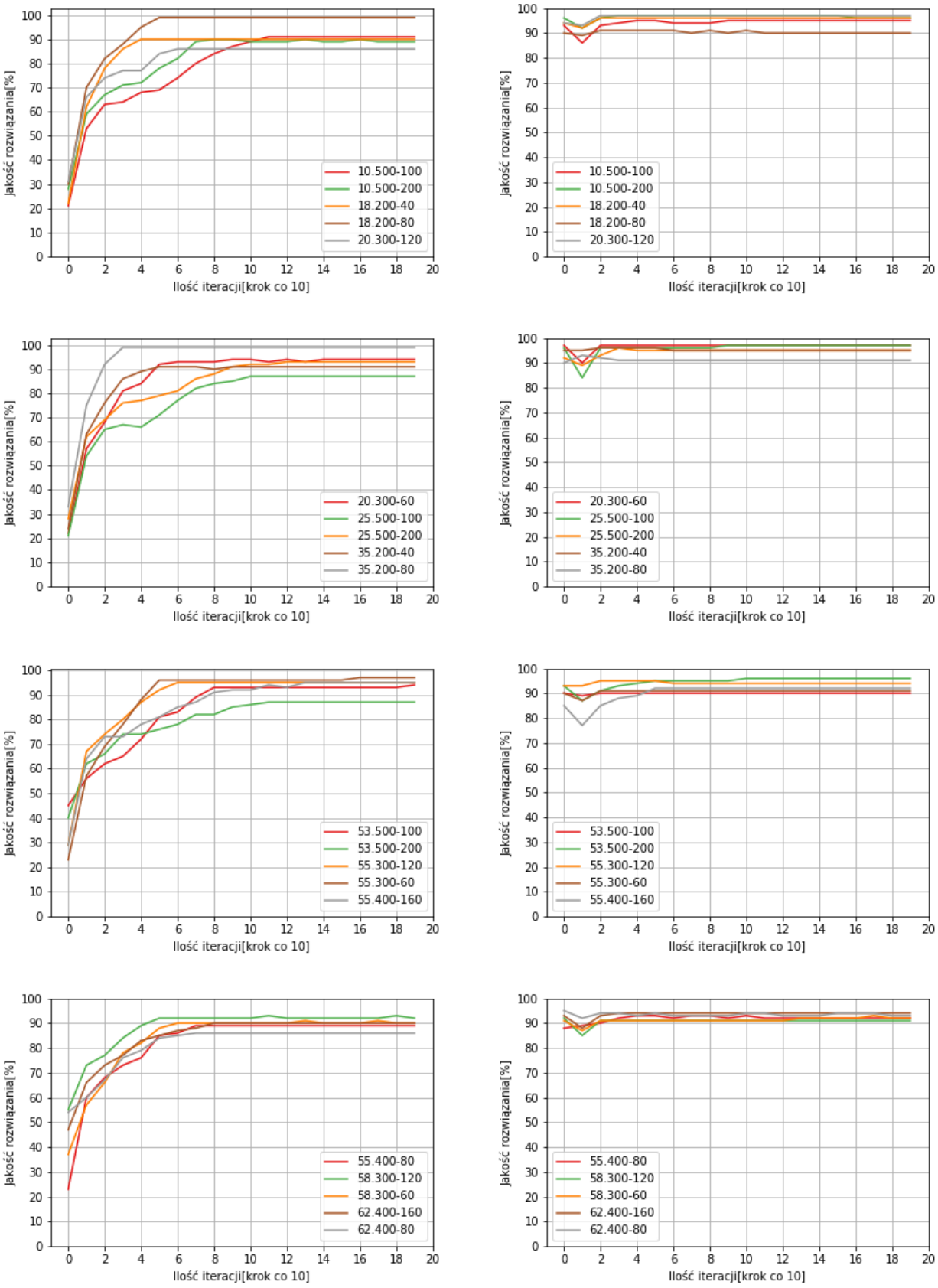

Figure 1: Porównanie algorytmu dla błędów negatywnych, losowych bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od ilości iteracji.

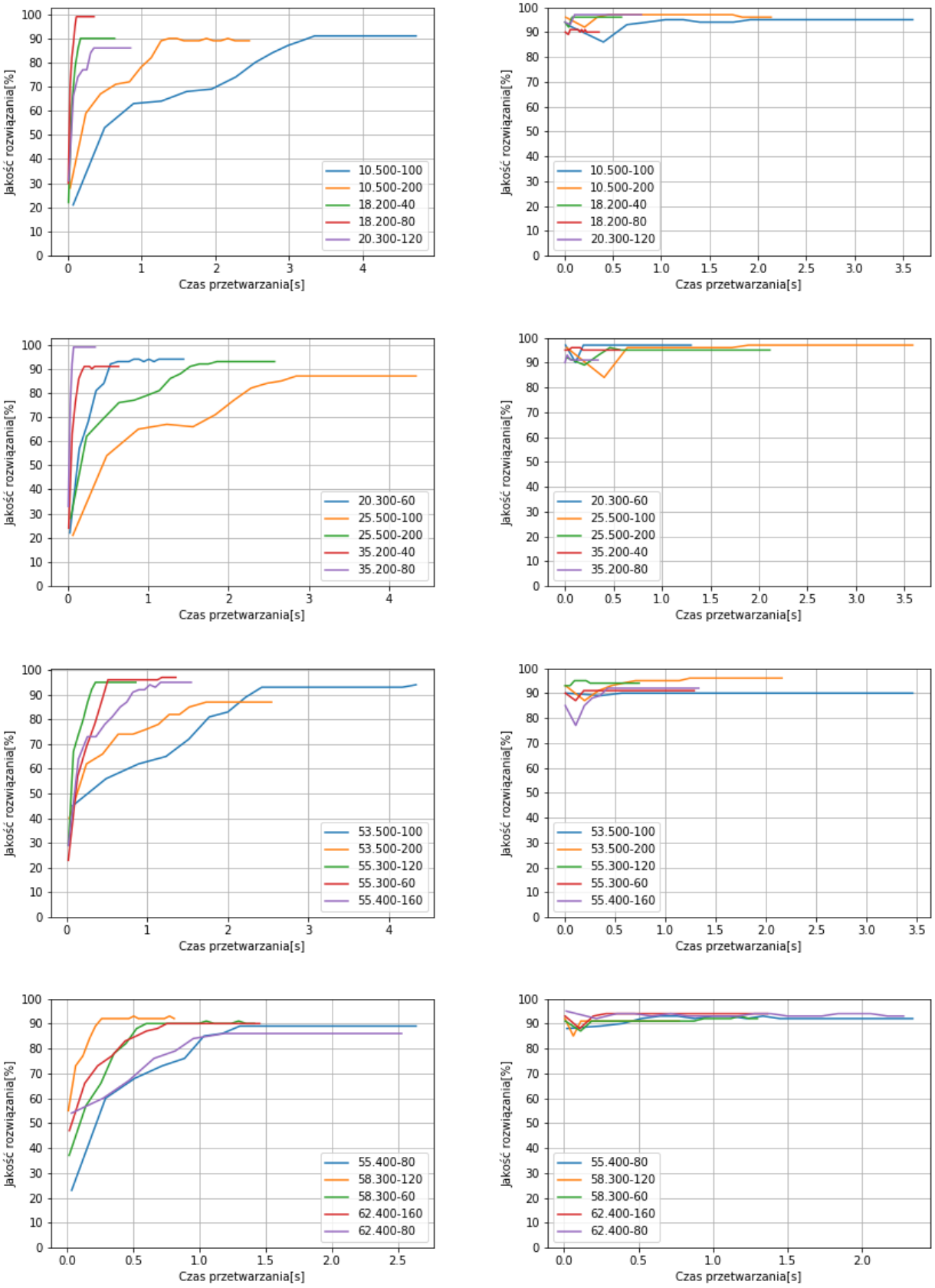

Figure 2: Porównanie algorytmu dla błędów negatywnych, losowych bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od czasu przetwarzania

Jak widać, dla błędów negatywnych, losowych algorytm otrzymuje rozwiązanie o średniej jakości 90%. W algorytmie z wykorzystaniem osobnika wygenerowanego przez algorytm zachłanny częściej uzyskiwane są lepsze wyniki z powodu początkowego uszeregowania które ma bardzo dobrą jakość. Rozwiązania algorytmu bez

wykorzystania algorytmu zachłannego zbiegają do podobnej jakości wyników. Czasem jakość wyników jest nawet lepsza, prawdopodobnie ze względu na to, że osobniki za szybko upodabniają się do wyniku algorytmu zachłannego. Dodatkowo warto zauważyć, że wykresy po prawej stronie wskazują na polepszenie wyniku wygenerowanego przez algorytm zachłanny.

2.2 Wyniki algorytmu dla błędów negatywnych na końcach sekwencji

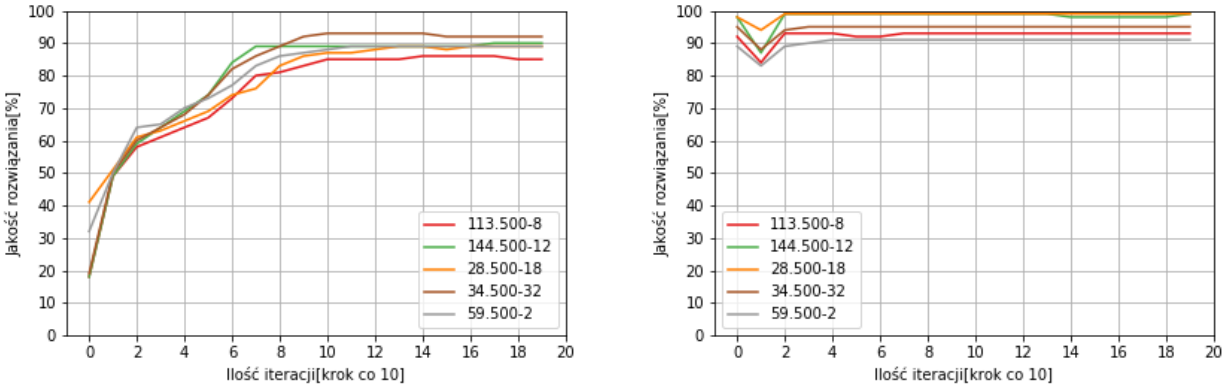

Figure 3: Porównanie algorytmu dla błędów negatywnych na końcach sekwencji bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od ilości iteracji

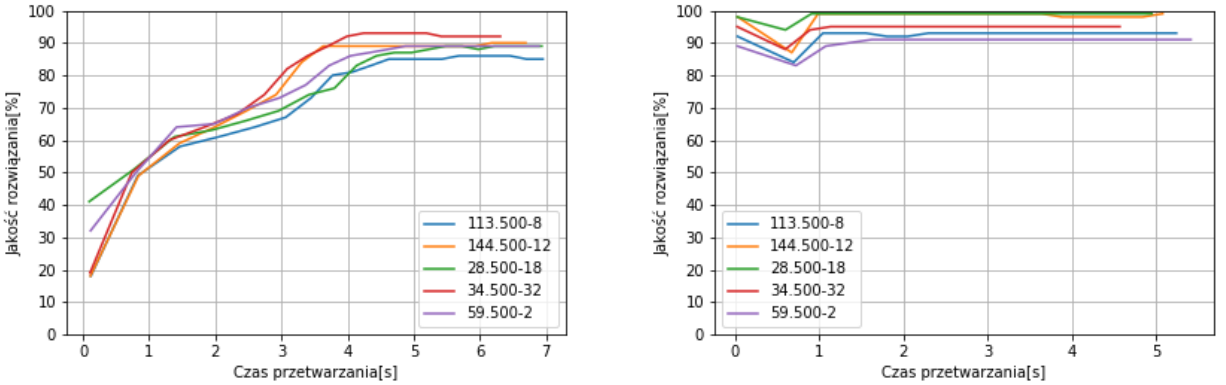

Figure 4: Porównanie algorytmu dla błędów negatywnych na końcach sekwencji bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od czasu przetwarzania

W przypadku błędów negatywnych na końcach sekwencji wyniki są zbliżone do wyników błędów negatywnych losowych. Różnicą jest fakt, że przy rozpoczęciu od wygenerowania losowej populacji początkowej zbieganie trwa dłużej.

2.3 Wyniki algorytmu dla błędów pozytywnych, losowych

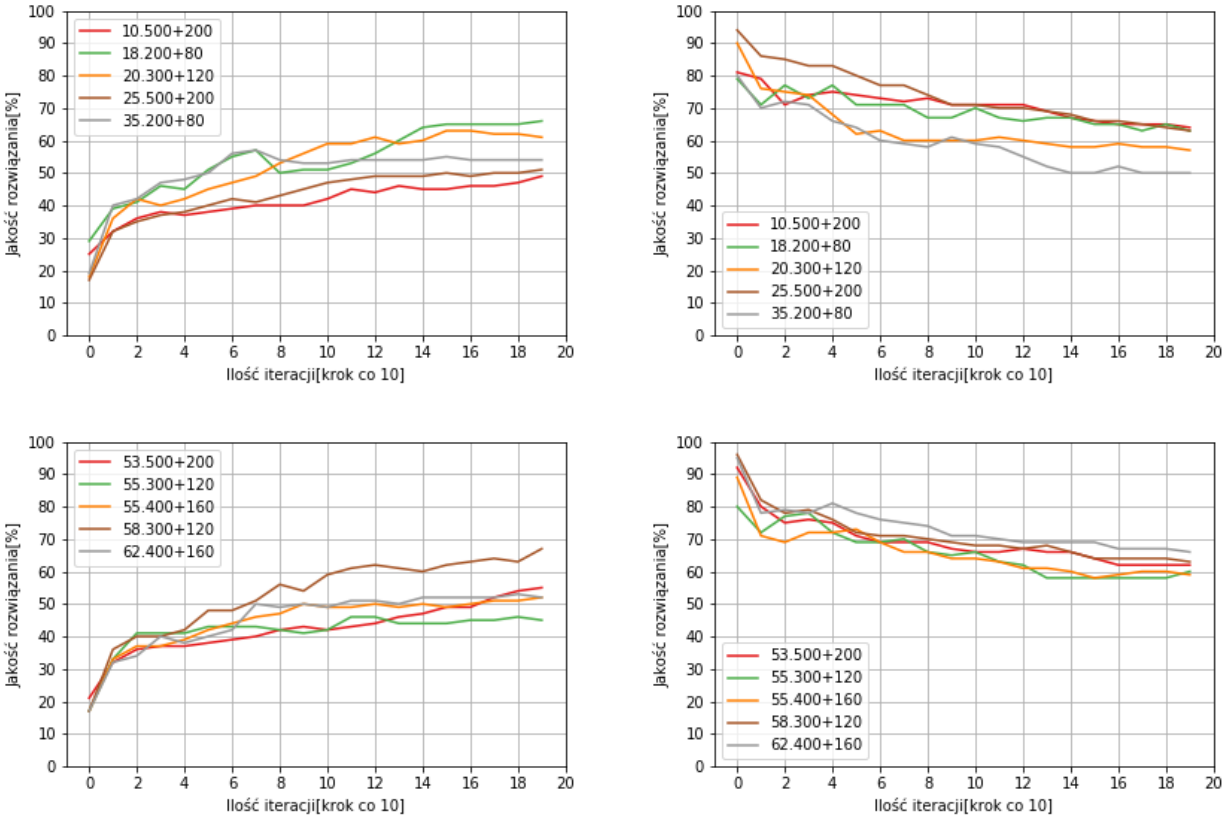

Figure 5: Porównanie algorytmu dla błędów pozytywnych, losowych bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od ilości iteracji

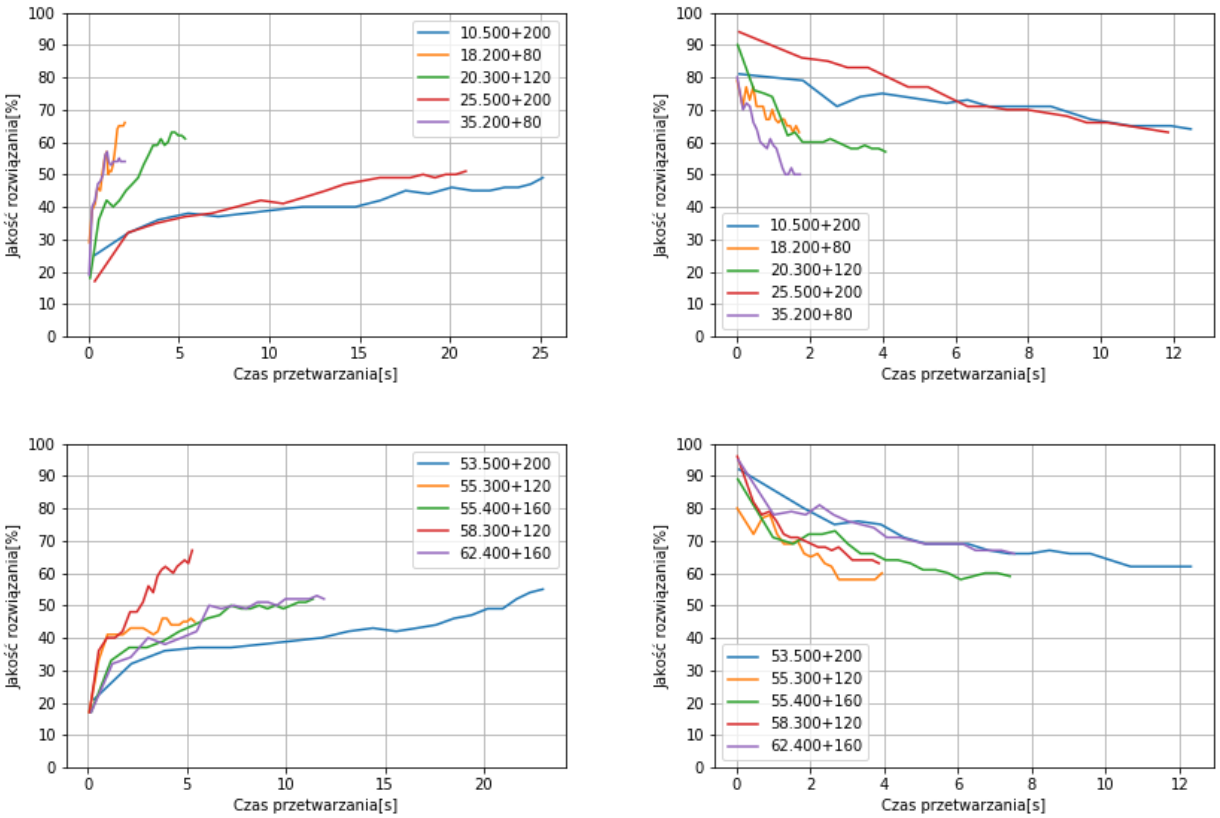

Figure 6: Porównanie algorytmu dla błędów pozytywnych, losowych bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od czasu przetwarzania

Dla błędów pozytywnych losowych algorytm nie działa już tak dobrze. Jak widać algorytm bez początku zachłannego uzyskuje średnie wyniki w okolicach 60%. Dodatkowo algorytm z początkiem zachłannym tylko pogarsza swoje wyniki po lepszym wyniku zachłannym, żeby ostatecznie dojść do prawie takiego samego wyniku jak algorytm bez początku zachłannego. Jest to związane z przyjęciem w metodzie mutacji kryterium wyboru oligonukleotydu na podstawie najsłabszego dopasowania. Powodem pogarszania wyniku jest fakt, że w przypadku błędów pozytywnych często wiąrze się to z wyborem oligonukleotydu będącego tym, nie występującym w oryginalnej sekwencji.

2.4 Wyniki algorytmu dla błędów pozytywnych, na końcach sekwencji

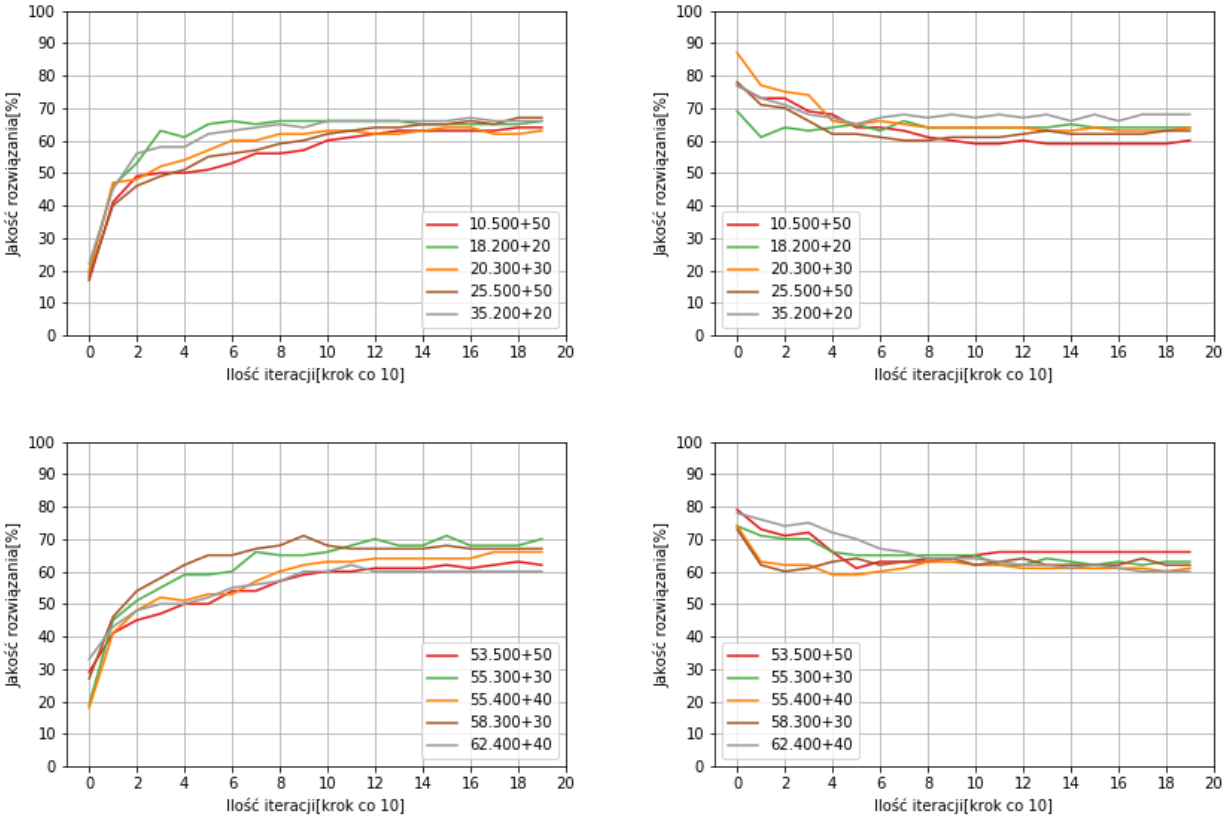

Figure 7: Porównanie algorytmu dla błędów pozytywnych na końcach sekwencji bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od ilości iteracji

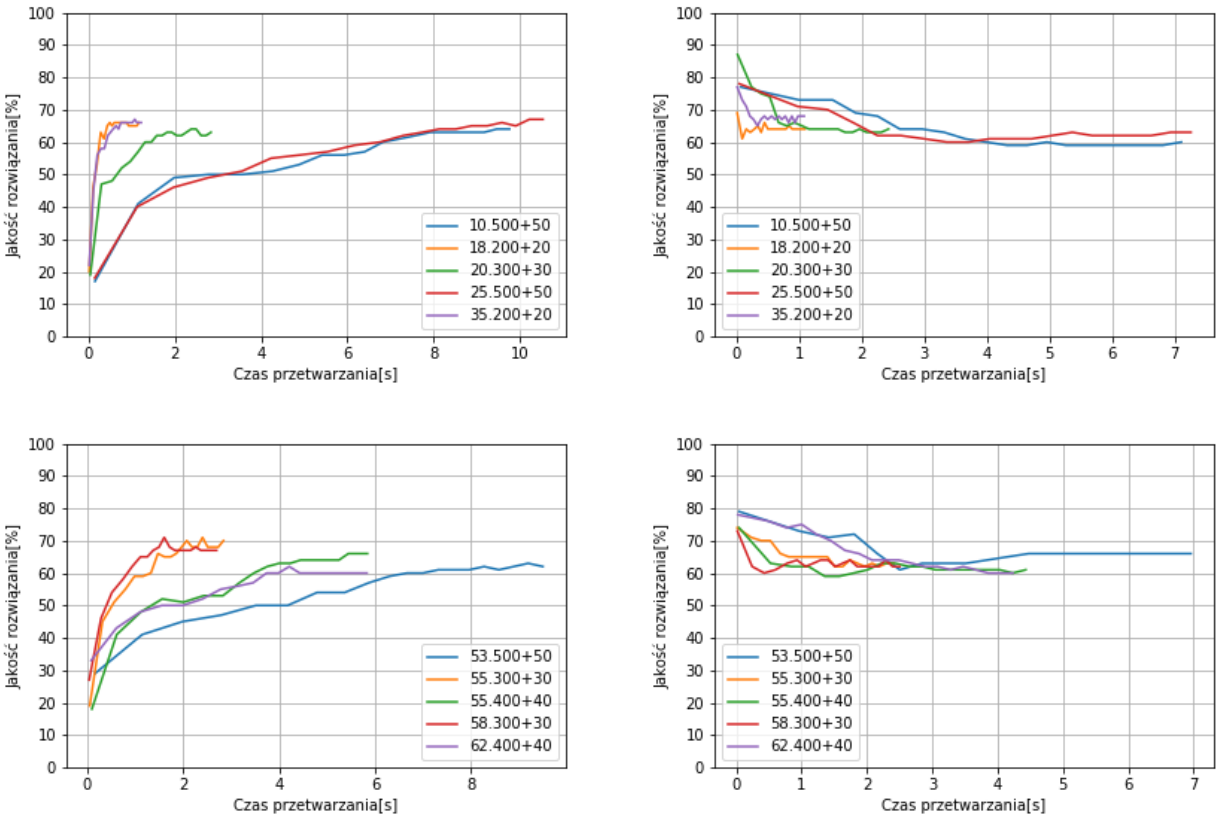

Figure 8: Porównanie algorytmu dla błędów pozytywnych na końcach sekwencji bez (po lewej stronie) oraz z wykorzystanym osobnikiem wygenerownym przez algorytm zachłanny (po prawej stronie) w zależności od czasu przetwarzania

Ostatnią kategorią są błędy pozytywne z przekłamaniami na końcach oligonukleotydów. Jak widać radzi on sobie odrobinę lepiej niż dla błędów pozutywnych losowych, dodatkowo algorytm z rozpoczęciem zachłannym nie maleje aż tak bardzo (algorytm zachłanny ma większe problemy z optymalnym ustawieniem i jego wykonanie nie powoduje od razu ustawienia pierwszej iteracji na wysokości 90-95%, a nieco niżej na 70-80%).

3 Wnioski