

Model wzrostu guza jako problem odwrotny

Leszek Siwik

2018-02-20

Abstract

Dokument stanowi (wstępny) raport z prac nad zdefiniowaniem problemu wzrostu Tumora jako problemu odwrotnego wraz z próbą jego rozwiązywania z wykorzystaniem ewolucji.

Contents

1	Background i punkt startowy	3
2	Definicja problemu	4
3	Użyty algorytm	4
3.1	Algorytm	4
3.2	Operatory wariacyjne	5
3.3	Zrównoleglenie algorytmu/solwera	5
4	Wstępny eksperyment	5
4.1	Parametry algorytmu	5
4.2	Parametry modelu	5
4.3	Uzyskane wyniki	6
4.4	Napotkane problemy	6
5	Ewolucyjne poszukiwanie wartości referencyjnych wybranych parametrów startując z ‘przypadkowego’ zakresu ich wartości	8
5.1	Seria 1 - zaburzanie pojedynczego parametru	8
5.1.1	Zaburzanie parametru $p_7 = o^{prol}$	10
5.1.2	Zaburzanie parametru $p_8 = o^{death}$	12
5.1.3	Zaburzanie parametru $p_9 = T^{prol}$	16
5.1.4	Zaburzanie parametru $p_{10} = T^{death}$	17
5.2	Kwestie do zweryfikowania	17
6	Eksperyment z fixowaniem wartości parametrów niepodlegających ewolucji/poszukiwaniu	18
6.1	Sposób przeprowadzenia eksperymentów	18
6.2	Problemy na jakie natrafiono	18
6.3	Zmiany wprowadzone do solwera	19
6.4	Aktualna postać funkcji celu	19
6.5	Zaburzanie 4 parametrów	20
6.5.1	Zmiennosc parametrów	20
6.5.2	Rozkład P7	20
6.5.3	Rozkład P8	22
6.5.4	Rozkład P9	25
6.5.5	Rozkład P10	27
6.5.6	Rozkład TumorVolume	29
6.5.7	Wnioski	31
6.6	Zaburzanie 4 parametrów - wydłużone działanie	33
6.6.1	Zmiennosc parametrów	33

6.6.2	Rozkład P7	33
6.6.3	Rozkład P8	35
6.6.4	Rozkład P9	38
6.6.5	Rozkład P10	40
6.6.6	Rozkład TumorVolume	42
6.6.7	Wnioski	44
6.7	Zaburzenie 1 parametru	45
6.7.1	Zaburzenie P7	45

1 Background i punkt startowy

W listopadzie/grudniu 2017 przeprowadzono probe analizy wzrażliwości modelu/solwera wykorzystywanego do symulacji wzrostu Tumora w zależności od przyjętych parametrów.

Po drobnych doustaleniach jako referencyjne przyjęto następujące wywołanie/zestaw parametrów solwera:

```
./tumor 2 80 10000 0.1 1000 0.5 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555  
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

Zas dla poszczególnych parametrów przyjęto w analizie następujący zakres ich zmienności:

- $p_1 = 2$ – stopień splajnow – nie zmieniany
- $p_2 = 80$ – ilość elementów – nie zmieniany
- $p_3 = 10000$ – liczba kroków czasowych – nie zmieniany
- $p_4 = 0.1$ – wielkość kroku czasowego – nie zmieniany
- $p_5 = 1000$ – co ile kroków zapisywać wynik – nie zmieniany
- $p_6 = \tau_B = 0.5 \pm 10\%$
- $p_7 = o^{prol} = 10 \pm 10\%$
- $p_8 = o^{death} = 2 \pm 10\%$
- $p_9 = T^{prol} = 10 \pm 10\%$
- $p_{10} = T^{death} = 100 \pm 10\%$
- $p_{11} = P_b = 0.001 \pm 10\%$
- $p_{12} = r_b$ – zmieniany w zakresie $(0.00001, 3/10]$
- $p_{13} = \beta_M = 0.625 \pm 10\%$
- $p_{14} = \gamma_A = 0.3205 \pm 10\%$
- $p_{15} = \chi_{oA} = 0.0064 \pm 10\%$
- $p_{16} = \gamma_{oA} = 0.0064 \pm 10\%$
- $p_{17} = \chi_c = 0.0000555 \pm 10\%$
- $p_{18} = \gamma_c = 0.01 \pm 10\%$
- $p_{19} = \alpha_0 = 0.0000555 \pm \mathbf{50\%}$
- $p_{20} = \gamma_T = 0.01 \pm \mathbf{50\%}$
- $p_{21} = \alpha_1 = 0.4 \pm \mathbf{50\%}$

Podczas obliczeń każdy z wyżej wymienionych zakresów zmienności dzielono na 20 równych kawałków i startując od wywołania referencyjnego stopniowo podstawiano kolejną z 21 wartości parametru w zdefiniowanym zakresie zmienności.

A zatem kolejne wywołania solwera dla parametru p_6 definiowane były jako:

- ./tumor 2 80 10000 0.1 1000 **0.45** 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
- ./tumor 2 80 10000 0.1 1000 **0.455** 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
- ...
- ./tumor 2 80 10000 0.1 1000 **0.55** 10 2 10 100 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555
0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4

Następnie powtarzano procedurę dla kolejnego parametru powracając z wartością właśnie analizowanego parametru do wartości referencyjnej (0.5 w powyższym przypadku).

Uzyskane wyniki, oraz ich wstępna “obróbka” zebrana została w arkuszu dostępnym **Tutaj**.

Na pierwszy rzut oka, w uzyskanych wynikach w mojej opinii nie widać specjalnie silnych zależności, niemniej daje się zidentyfikować parametry których zmiana wartości zdaje się mieć istotny “kierunkowy” wpływ na zachowanie modelu i wyniki uzyskiwane z implementującego go solwera.

W szczególności jako parametry mające obserwowalny wpływ na wielkość tumora wyselekcjonowano parametry $p_7 = o^{prol}$, $p_8 = o^{death}$, $p_9 = T^{prol}$, $p_{10} = T^{death}$ a zidentyfikowaną zależność pomiędzy wartościami tych parametrów (w przyjętym zakresie zmienności) a objętością guza zobrazowano na poniższym wykresie.

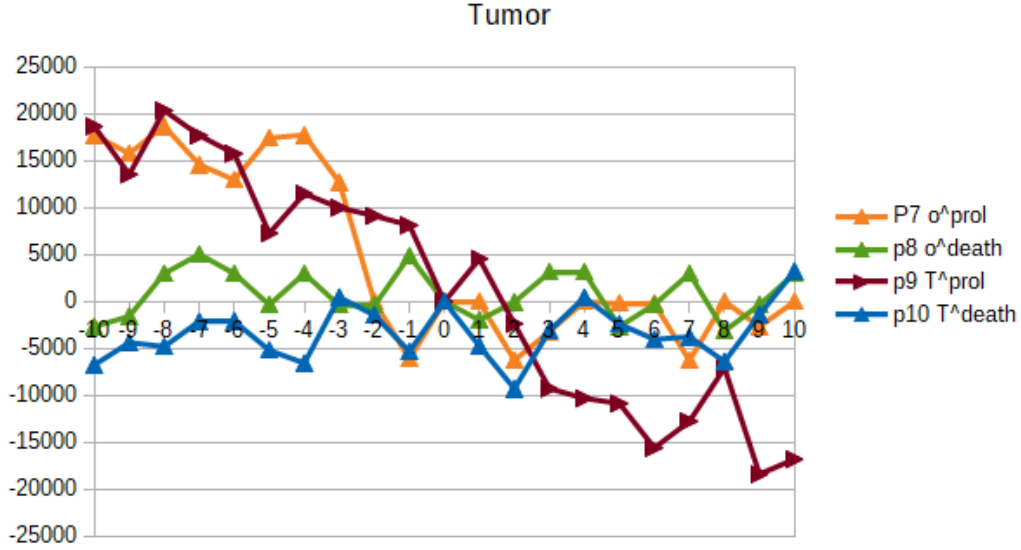


Figure 1: Związek pomiędzy wartościami parametrów p_7 , p_8 , p_9 i p_{10} a objętością guza

2 Definicja problemu

Jako wstępny problem weryfikujący możliwość dalszego researchu bazując na obserwacjach i wynikach skomentowanych w poprzednim paragrafie zdefiniowano problem odwrotny polegający na poszukiwaniu takich wartości parametrów p_7 , p_8 , p_9 i p_{10} aby minimalizować odchyłkę uzyskiwanych wartości objętości guza od objętości guza uzyskanej dla symulacji referencyjnej.

Ta referencyjna wartość objętości guza uzyskana podczas pierwszej fazy eksperymentów to 402000. A zatem funkcja fitness definiowana została jako:

$$abs(402000 - tumor_{volume}) \quad (1)$$

Wartość ta była minimalizowana podczas ewolucji.

3 Użyty algorytm

3.1 Algorytm

Użyty w eksperymentach algorytm to klasyczny algorytm ewolucyjny, spięty z solverem symulującym rozrost guza działający zgodnie z przepisem jak poniżej:

```
public void run() {
    List<S> offspringPopulation;
    List<S> matingPopulation;

    population = createInitialPopulation();
    population = evaluatePopulation(population);
    initProgress();
    while (!isStoppingConditionReached()) {
        matingPopulation = selection(population);
        offspringPopulation = reproduction(matingPopulation);
        offspringPopulation = evaluatePopulation(offspringPopulation);
    }
}
```

```

        population = replacement(population, offspringPopulation);
        updateProgress();
    }
}

```

3.2 Operatory variacyjne

Użyty algorytm wykorzystuje

- **Simulated Binary Crossover (SBXCrossover)**¹ jako operator krzyżowania
`crossoverOperator = new SBXCrossover(0.9,20);`
- **mutacje wielomianowa** jako operator mutacji z następującymi parametrami:
`mutationOperator = new PolynomialMutation(1.0 / problem.getNumberOfVariables(),20);`
- oraz **selekcje turniejowa** jako operator selekcji.

3.3 Zrównoleglenie algorytmu/solwera

Z uwagi na czas wykonania solwera na etapie ewaluacji osobników zrównoleglono wywołania solwera.

Tzn. sam algorytm ewolucyjny nie był zrównoleglany w tym sensie, że nie ma tam wielu przetwarzanych jednocześnie (sub)populacji etc. Mamy tam “klasyczną”, pojedynczą globalnie przetwarzaną populację.

Natomiast zastosowano zrównoleglenie na etapie ewaluacji osobników. Tzn dla każdego osobnika, dla którego potrzebna jest jego ewaluacja, uruchamiany jest osobny proces/watek w którym wykonywany jest kod solwera.

Zrównoleglewnie realizowane jest poprzez wywołania mpi:

```

command = "mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 "
          + p7 + " " + p8 + " " + p9 + " " + p10
          + " 0.001 0.3 0.625 0.3205"
          + " 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333"
          + " 10 0.003 2 5 25 24 0.003 0.4";

p = Runtime.getRuntime().exec(
    command);

```

4 Wstępny eksperyment

4.1 Parametry algorytmu

Ponieważ natknieto się na problemy z czasem wykonania pojedynczej instancji solwera, w pierwszym opisywanym eksperymencie ograniczono obliczenia do 100 ewaluacji (100 uruchomien solwera).

Rozmiar populacji algorytmu przyjęto na 20 (a zatem na etapie ewaluacji uruchamiano max 20 procesów mpi wykonujących kod solwera.)

4.2 Parametry modelu

Zakres zmienności parametrów (ich dopuszczalne granice) zdefiniowane zostały tak samo jak to miało miejsce przy analizie wrażliwości modelu, tzn:

- $p_7 = o^{prol} \in [9.0, 11.0]$ (war ref. 10.0)

¹Szczegoly np. tutaj: <https://pdfs.semanticscholar.org/b8ee/6b68520ae0291075cb1408046a7dff9dd9ad.pdf>

- $p_8 = o^{death} \in [1.8, 2.2]$ (war ref. 2.0)
- $p_9 = T^{prol} \in [9.0, 11.0]$ (war ref. 10.0)
- $p_{10} = T^{death} \in [90.0, 110.0]$ (war ref. 100.0)

4.3 Uzyskane wyniki

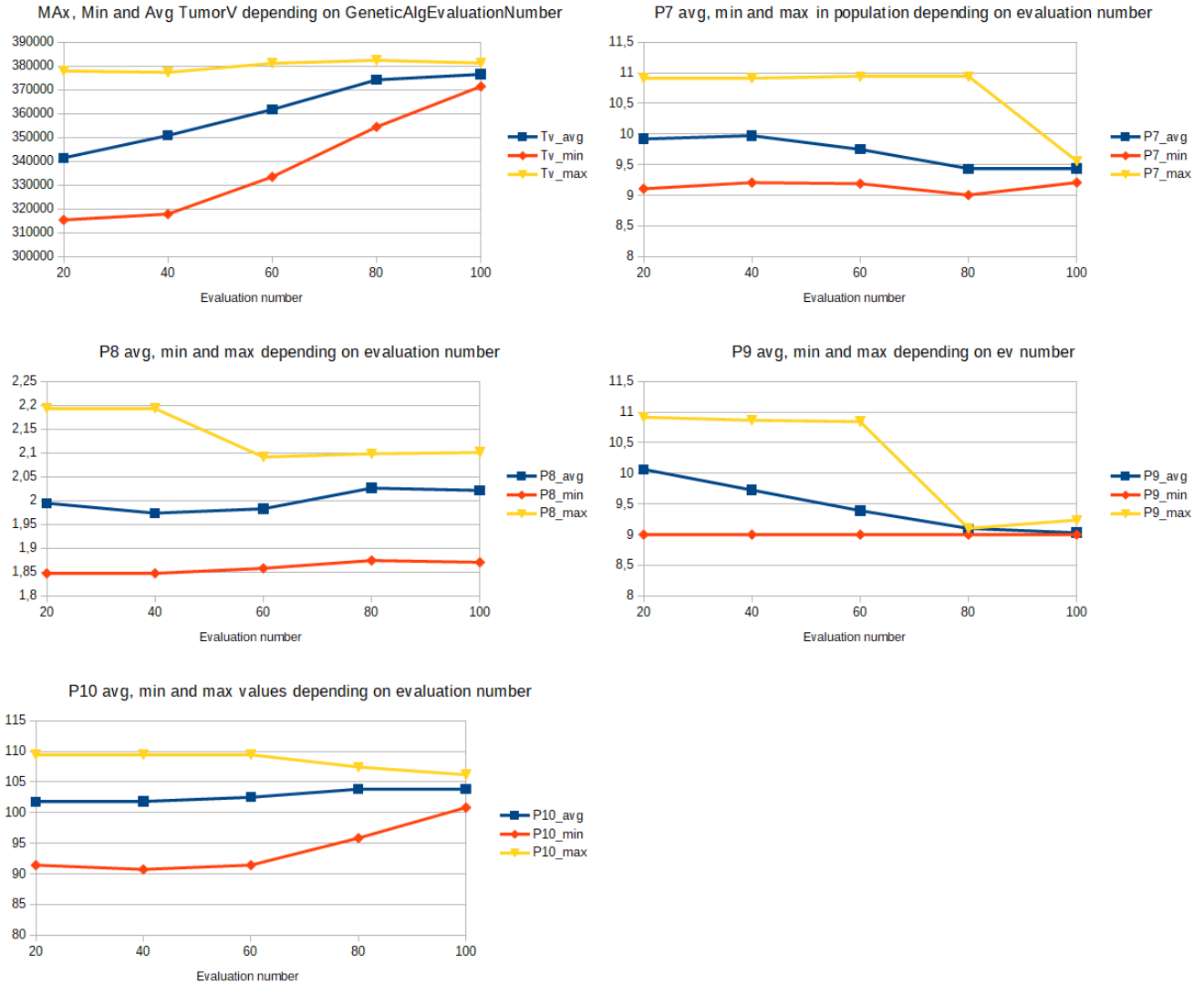


Figure 2: Results in first experiment

4.4 Napotkane problemy

Poniżej opisano problem na który nadziano się podczas wywoływania solvera wraz z próbą wyjaśnienia jego przyczyn. Być może przyda się w analizie/poprawkach modelu/solvera ew. przyszłym użytkownikom zaoszczędzi trochę czasu :)

Kłopot na jaki natrafiono to potężny wzrost czasu pojedynczego wykonania solvera przy niektórych zestawach parametrów wejściowych.

Jak bardzo to wydłużało obliczenia: np w “normalnych” warunkach pojedyncze wywołanie solwera trwało ok 100s o tyle takie “nienormalne” wydłużało się np do 2500s albo w ogóle się nie kończyło bo powodowało tak dużą alokację pamięci że było ubijane przez mpiruna.

M.in to było powodem wyczerpywania zasobów na węzle atari. Natrafiono na to ponieważ początkowa konfiguracja alg ewolucyjnego zakładała większą zmienność / szersze zakresy wartości dopuszczalnych poszczególnych parametrów (założono np wstępnie zmienność $p_7 \in [5, 15]$, $p_8 \in [1, 3]$, $p_9 \in [5, 15]$ i $p_{10} \in [50, 150]$)

Przykładowe wywołanie które bardzo długo się liczyło to np:

```
../TumorGenetyka/tumor 2 80 10000 0.1 1000 0.5 5.819810396051385 3.4400377040438777 11.112855386932164
125.35805356351595 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5
0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4
```

Po przyjrzeniu się temu przypadkowi diagnoza jest taka że to niefortunne złożenie wartości parametrów. Parametry 7 - 10 to odpowiednio: 7 - ile co najmniej tlenu potrzeba, żeby komórki się mnożyły 8 - ile co najwyżej musi tlenu być, żeby żyły 9 - jak wolno się mnożyć (większe - wolniej) 10 - jak wolno umierają

Uruchomienie które się nie doliczyło ma bardzo niską w porównaniu z resztą wartość 7, czyli rak nie potrzebuje bardzo dużo tlenu żeby żyć, więc będzie się rozrastał swobodnie. Jednocześnie ma bardzo niską wartość 8, czyli musi mieć dość dużo tlenu żeby nie umierać - tam gdzie jest za mało, jest tworzona substancja powodująca rozrost żył.

Różnica między tymi dwoma wartościami w tym uruchomieniu jest bardzo mała (5.8 - 3.4 vs np. 14 - 1 w drugim), więc jest spora szansa że jak rak będzie wołał o tlen to dostanie go od razu na tyle dużo, żeby móc się rozrastać. Jednocześnie parametr 10 jest stosunkowo duży, więc nawet jak nie dostanie tlenu od razu to nie będą komórki umierać bardzo szybko.

Czyli podsumowując:

- rak będzie rosł jak dostanie nawet niewiele tlenu - rozejdzie się łatwo na większość dziedziny
- jest “wymagający”, będzie krzyczał o tlen (= rozrost żył) dopóki nie będzie miał prawie tyle żeby móc rosnąć, więc spowoduje gęste żyły
- nie umiera szybko bez tlenu, więc może poczekać aż te żyły się utworzą

Teraz jakie z tego praktyczne wnioski... trudno powiedzieć coś o konkretnych zakresach parametrów. Być może należałoby je nieco zawęzić (7 i 8 raczej będą najważniejsze), być może jakieś ograniczenia zakresu na podstawie wartości jednego z nich (np. żeby różnica między 7 a 8 nie była za duża) acz to chyba ciężko byłoby zrobić od strony algorytmu genetycznego. Ale może to nie będzie potrzebne jak się zrobi to co jest opisane dalej.

W przypadkach które się doliczyły też dość sporo się tych naczyń tworzy, dużo więcej niż się spodziewałem. Jedną rzecz jaką na szybko można zmienić która na pewno przyspieszy znacznie symulację w tej fazie z gęstymi naczyniami to usunięcie łączenia naczyń które rosną tak, że przez siebie przechodzą, bo to jest zaimplementowane z myślą o małych ilościach naczyń i poszukuje bliskich węzłów brutalnie kwadratowo. Nie powinno to raczej mieć specjalnie dużego wpływu na wyniki (tak mi się wydaje).

Żeby to zrobić wystarczy zakomentować linijki 187-192 w tumor/vasculature/vasculature.hpp:

```
auto neighbor = find_neighbor(new_tip, tip, cfg.segment_length);
if (neighbor != nullptr) {
    connect(neighbor, new_tip);
    removed = true;
    it = sprouts.erase(it);
}
```

Co zostało zrobione i wyniki z tego pierwszego eksperymentu miały to w solwerze zakomentowane.

Żeby zobrazować problem (być może naprowadzić na rozwiązanie) poniżej czasy wykonania solwera, ilość segmentów i węzłów żył, objętość/masa guza dla:

- przypadku normalnego (brak nadmiernego rozrostu żył) ../TumorGenetyka/tumor 2 80 10000 0.1 1000 0.5 14.449578938363844 1.3853073609726758 7.860871658614398 63.030234570317525 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4

- przypadku pesymistycznego (nadmierny rozrost) ../TumorGenetyka/tumor 2 80 10000 0.1 1000 0.5 5.819810396051385 3.4400377040438777 11.112855386932164 125.35805356351595 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4

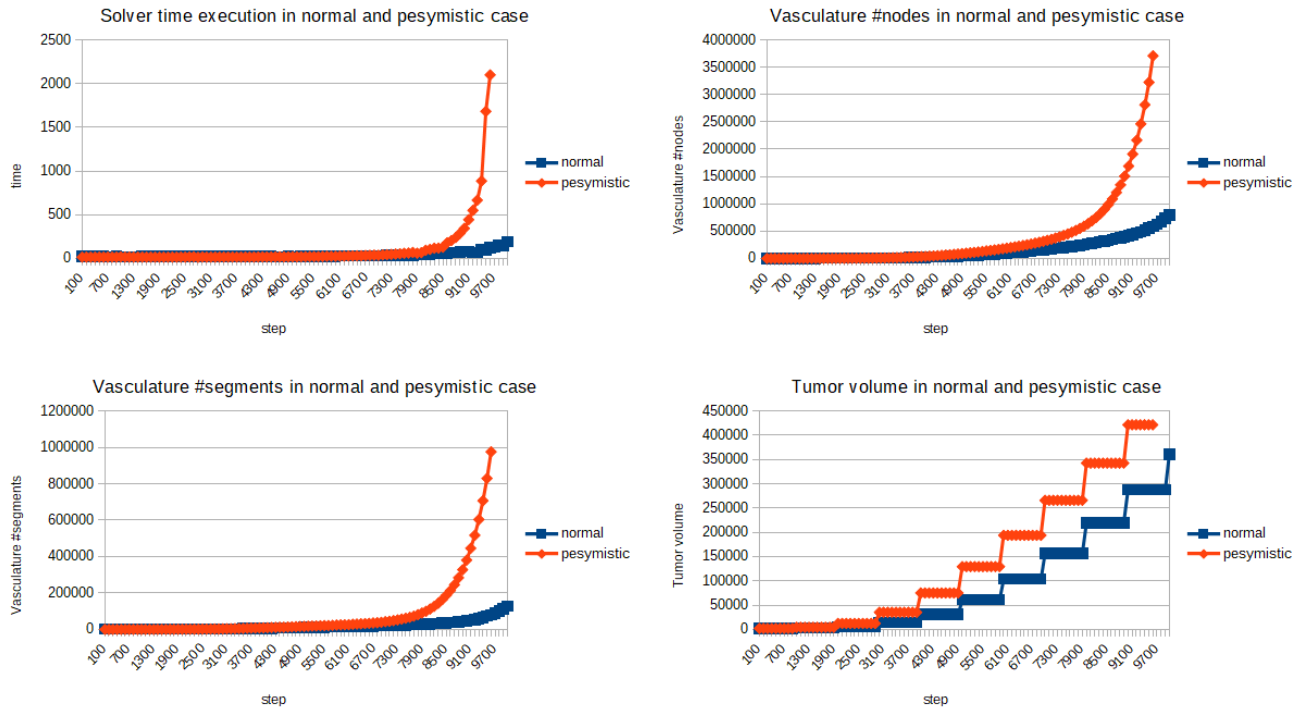


Figure 3: Czas obliczen, siatka zyl, masa guza w “normalnym” i “pesymistycznym” przebiegu symulacji

5 Ewolucyjne poszukiwanie wartości referencyjnych wybranych parametrow startując z ‘przypadkowego’ zakresu ich wartosci

Po opisanym wcześniej wstępnym zweryfikowaniu działania narzędzi - przystapiono do przeprowadzenia eksperymentu polegającego w uproszczeniu na zweryfikowaniu zdolności odnalezienia przez wcześniej opisany solver (hybryda alg. ewolucyjnego z solverem modelującym wzrost guza)

referencyjnych wartości wybranych parametrów startując z innego zakresu ich wartości.

Podczas eksperymentu stopniowo zwiększamy ‘stopień trudności’ tzn zaburzeniu ulega stopniowo (w stosunku do zakresu referencyjnego) rosnąca ilość parametrów (tych przyjętych na wstępnym etapie weryfikacji jako istotne czyli odpowiednio: $p_7 = o^{prol}$, $p_8 = o^{death}$, $p_9 = T^{prol}$ oraz $p_{10} = T^{death}$) czyli:

- zaburzeniu podlega 1 parameter (4 symulacje po jednej dla ‘indywidualnie’ zaburzonych parametrów $p_7 \dots p_{10}$),
- zaburzeniu podlegają 2 parametry (6 symulacji dla różnych par zaburzonych parametrów)
- zaburzeniu podlegają 3 parametry (4 symulacje dla jednego parametru niezaburzonego)
- zaburzeniu podlegają 4 parametry (1 symulacja dla wszystkich zaburzonych)

5.1 Seria 1 - zaburzanie pojedynczego parametru

Zgodnie z wcześniejszym opisem pierwsza seria eksperymentów polega na pojedynczym zaburzaniu parametrów $p_7 \dots p_{10}$ i weryfikacji możliwości odnalezienia przez hybrydowy solver wartości referencyjnych tych parametrów Eksperymenty przeprowadzano w ten sposób, że na poziomie alg ewolucyjnego zmieniano zakres dopuszczalnych

wartości zaburzanego parametru dla populacji początkowej , a następnie rozszerzano zakres dopuszczalny tego parametru w taki sposób aby obejmował on zakres referencyjny oraz ten wynikający z zaburzenia.

Innymi słowy populacja początkowa zawierała osobniki z zaburzonymi wartościami wybranego parametru (wybranych parametrów) natomiast wartości tego parametru dla osobników tworzonych w trakcie działania alg. ewolucyjnego generowane były z szerszego zakresu tj. $zakres_{referencyjny} \uplus zakres_{zaburzony} \uplus zakres_{dodatkowy}$

Wszystkie pozostałe parametry zarówno alg ewolucyjnego jak i wywołań symulatora wzrostu guza pozostawały niezmienione i były takie same jak opisane w sekcjach 3 oraz 1 zwiększając ilość ewaluacji ewolucji do 200.

Planowane do przyjęcia podczas eksperymentów zakresy zmienności poszczególnych analizowanych parametrów:

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_7	$p_7 = 10 \pm 10\%$	$p_7 \in [11.0, 15.0]$	$p_7 \in [5.0, 15.0]$
p_8	$p_8 = 2 \pm 10\%$	$p_8 \in [2.1, 3.0]$	$p_8 \in [1.0, 3.0]$
p_9	$p_9 = 10 \pm 10\%$	$p_9 \in [11.0, 15.0]$	$p_9 \in [5.0, 15.0]$
p_{10}	$p_{10} = 100 \pm 10\%$	$p_{10} \in [110.0, 150.0]$	$p_{10} \in [50.0, 150.0]$

5.1.1 Zaburzenie parametru $p_7 = o^{prol}$

5.1.1.1 Parametry eksperymentu

Dla przypomnienia parametr $p_7 = o^{prol}$ solwera mówi ile co najmniej tlenu potrzeba, żeby komórki się mnożyły. Na wstępnym etapie jako wartość referencyjną tego parametru przyjęto jako $p_7 = o^{prol} \in [9.0, 11.0]$ (z war ref. 10.0).

W pierwszym eksperymencie przyjęto zatem, iż początkowa seria wartości tego parametru (wartości tego parametru dla populacji początkowej alg ewolucyjnego) losowana będzie z przedziału $p_7' \in [11.0, 15.0]$ Natomiast dla osobników generowanych już w trakcie działania alg. ewolucyjnego jako dopuszczalny zakres wartości parametru p_7 przyjęto zakres $p_7 \in [5.0, 15.0]$

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_7	$p_7 = 10 \pm 10\%$	$p_7 \in [11.0, 15.0]$	$p_7 \in [5.0, 15.0]$

5.1.1.2 Otrzymane wyniki

Ponizej odpowiednio: rozkład p_7 w osobnikach na kolejnych etapach działania alg ewolucyjnego, jego wartości usrednione, maksymalne i minimalne w populacji, oraz odchyłka sredniej wartosci p_7 w populacji od wartosci referencyjnej w kolejnych iteracjach.

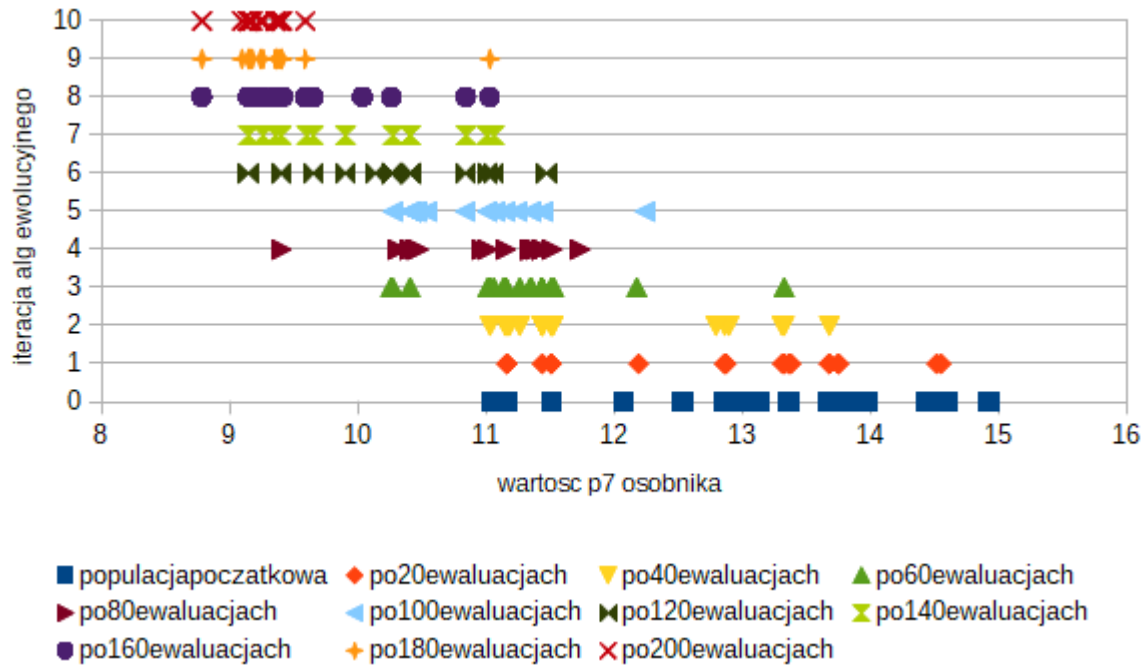


Figure 4: Rozkład p_7 w populacji

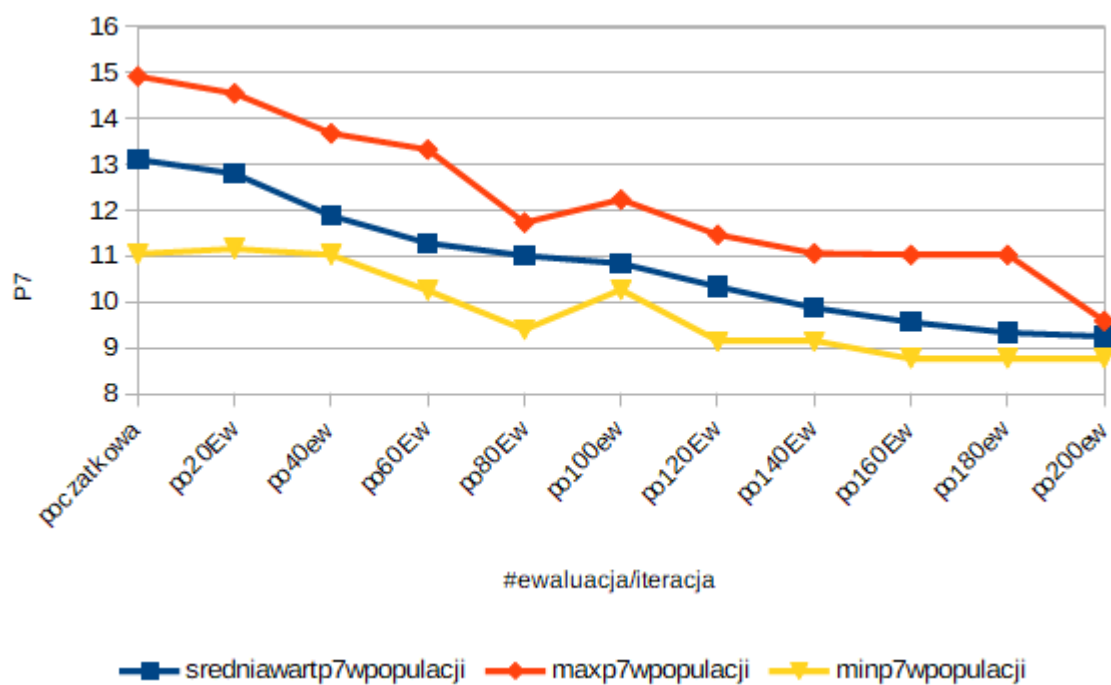


Figure 5: Wartości średnie, maksymalne oraz minimalne p7 na kolejnych etapach działania alg ewolucyjnego

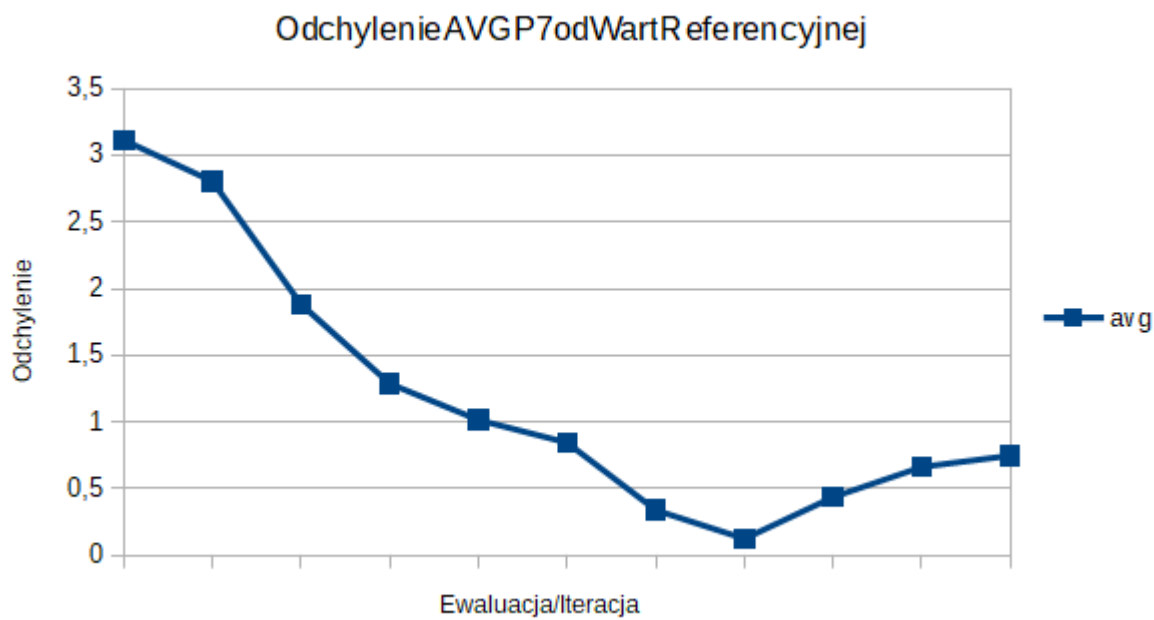


Figure 6: Odchyłka średniej wartości p7 w populacji od wartości referencyjnej

5.1.2 Zaburzenie parametru $p_8 = o^{death}$

5.1.2.1 Parametry eksperymentu

Dla przypomnienia parametr $p_8 = o^{death}$ solwera mówi ile ile co najwyżej musi być tlenu, żeby żyły. Na wstępnym etapie jako wartość referencyjną tego parametru przyjęto jako $p_8 = o^{death} \in [1.8, 2.2]$ (war ref. 2.0).

W niniejszym eksperymencie przyjęto zatem, iż początkowa seria wartości tego parametru (wartości tego parametru dla populacji początkowej alg ewolucyjnego) losowana będzie z przedziału $p_7' \in [2.1, 3.0]$ Natomiast dla osobników generowanych już w trakcie działania alg. ewolucyjnego jako dopuszczalny zakres wartości parametru p_8 przyjęto zakres $p_8 \in [1.0, 3.0]$

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_8	$p_8 = 2 \pm 10\%$	$p_8 \in [2.1, 3.0]$	$p_8 \in [1.0, 3.0]$

5.1.2.2 Otrzymane wyniki

Ponizej odpowiednio: rozkład p_8 w osobnikach na kolejnych etapach działania alg ewolucyjnego, jego wartości usrednione, maksymalne i minimalne w populacji, oraz odchyłka sredniej wartosci p_7 w populacji od wartosci referencyjnej w kolejnych iteracjach.

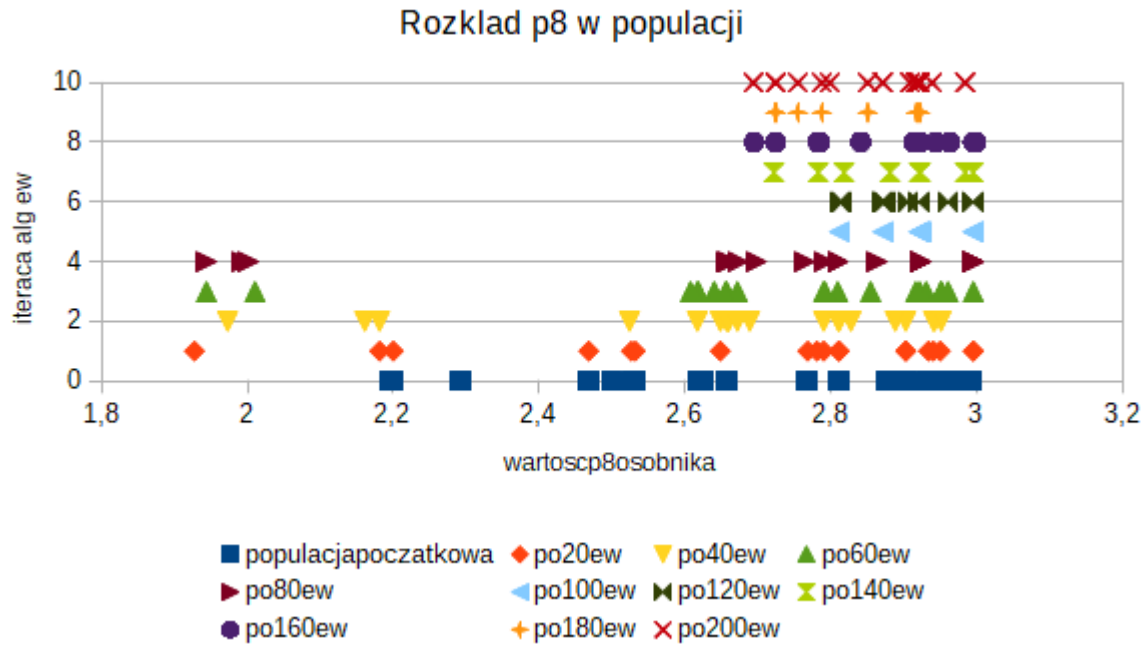


Figure 7: Rozkład p_8 w populacji

5.1.2.3 Napotkane problemy

Przy eksperymentowaniu z zaburzonym p_8 nadziano sie na klopociki prawdopodobnie bedace pochodna/kontynuacja problemów opisywanych wczesniej polegajacych na “niestabilnym” działaniu solwera przy niektórych wartosciach/kombinacjach parametrow

Ponizej wywolania solwera ktore sie nie powiodly (moze przyda sie kiedys do poprawek w solwerze)

Unable to run solver: java.lang.NumberFormatException: For input string: “=====“
 mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454590847505983

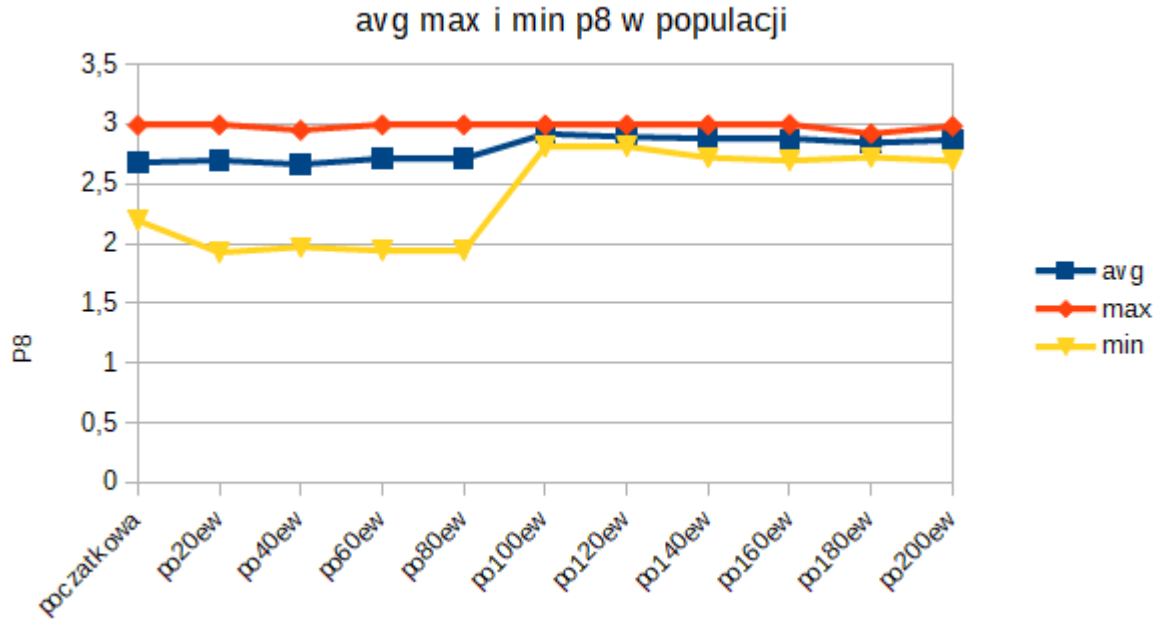


Figure 8: Wartości średnie, maksymalne oraz minimalne p8 na kolejnych etapach działania alg ewolucyjnego

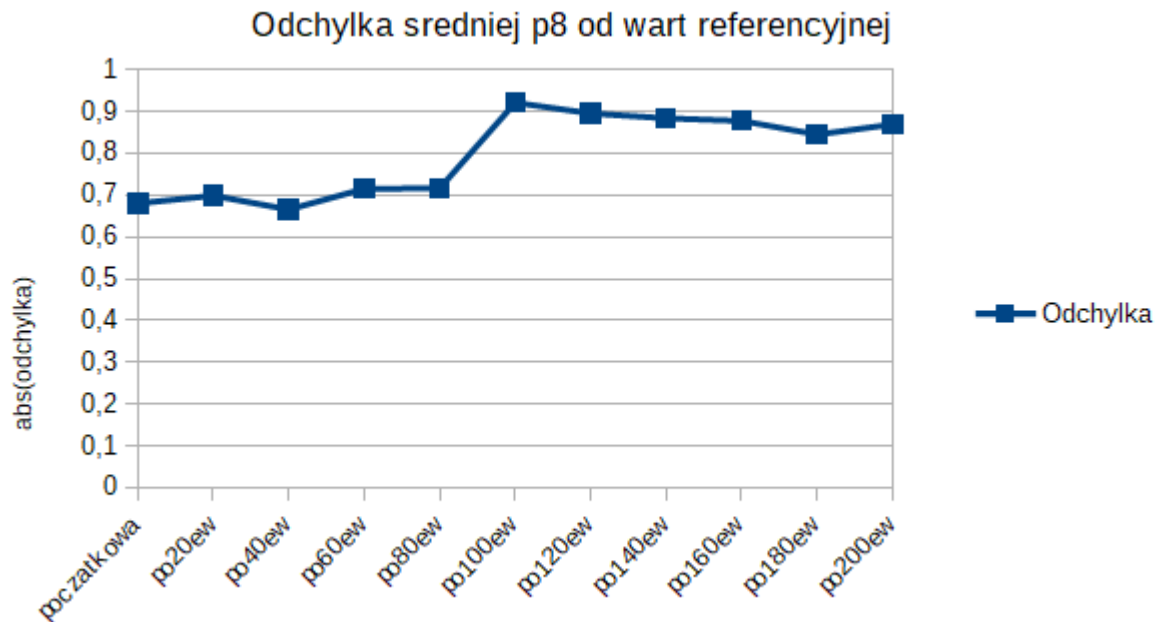


Figure 9: Odchylka średniej wartości p8 w populacji od wartości referencyjnej

```

2.9222689984674073 9.063008773943382 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.430679318110801
2.9613064430274214 9.001731455509889 94.84556414555985 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.312816143866291
2.8373159607698737 9.068744087595107 109.4082352110131 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.45412400070996 2.9512270566632774
9.134113928012875 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05
0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input string:
"=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.506700280337986
2.6729585594183844 9.127711756586537 98.28336497095415 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.428205108285592
2.6603929941110316 9.0677395656585 97.18615629884131 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.477124985184965
2.7878387123218205 9.071758596198237 108.77922017134355 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Read from err: termi-
nate Read from err: called Read from err: after Read from err: throwing Read from err: an Read
from err: instance Read from err: of Read from err: 'std::bad_alloc' Read from err: what(): Read
from err: std::bad_alloc Unable to run solver: java.lang.NumberFormatException: For input string:
"=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.45300605302644 2.991957323550271
9.06939941685454 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05
0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input string:
"=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.424280747231732 2.816132673178266
9.080270122482826 109.95693142895279 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5
0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input
string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454330657752786
2.6073509233841365 9.065938202205565 109.70008853564136 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.729087704398712
2.9222689984674073 9.146183592522865 98.30290819602372 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01
0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4

Unable to run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.312825087735012
2.9227547222493135 9.24885580468986 90.5577187348602 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.45412400070996 2.9512270566632774
9.11652843068546 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05
0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input string:
"=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454330657752786

```

```

2.7220574749335147 9.065931652601307 109.70008853564136 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.455134787600183 2.922261983693214
9.063918179902238 109.55449860456025 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5
0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input
string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.006641892977663
2.8373206073973387 9.002174195946964 93.65986847646002 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454160715570499
2.9956881333657854 9.001542159250945 94.68929923984707 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.401906819035798 2.996356833670716
9.254686609769006 90.49447838526919 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05
0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver: java.lang.NumberFormatException: For input string:
"=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.453536622315424
2.9424569354815837 9.249968498181236 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454590847505983
2.9222689984674073 9.25062940767948 96.92921554534925 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555
0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to run solver:
java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.422952034739607 2.923597621538141
9.005725604299714 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064 0.0000555 0.01 0.0000555 0.01 0.4 0.5
0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4

```

5.1.3 Zaburzenie parametru $p_9 = T^{prol}$

5.1.3.1 Parametry eksperymentu

Dla przypomnienia parametr $p_9 = T^{prol}$ solwera mówi jak wolno komórki się mnożą (większe - wolniej) Na wstępnym etapie jako wartość referencyjną tego parametru przyjęto jako $p_9 = T^{prol} \in [9.0, 11.0]$ (war ref. 10.0)

W niniejszym eksperymencie przyjęto zatem, iż początkowa seria wartości tego parametru (wartości tego parametru dla populacji początkowej alg ewolucyjnego) losowana będzie z przedziału $p'_9 \in [11.0, 15.0]$ Natomiast dla osobników generowanych już w trakcie działania alg. ewolucyjnego jako dopuszczalny zakres wartości parametru p_9 przyjęto zakres $p_9 \in [5.0, 15.0]$

5.1.3.2 Otrzymane wyniki

Eksperyment w trakcie realizacji. . .

5.1.4 Zaburzenie parametru $p_{10} = T^{death}$

5.1.4.1 Parametry eksperymentu

Dla przypomnienia parametr $p_{10} = T^{death}$ solwera mówi jak wolno umierają komórki. Na wstępnym etapie jako wartość referencyjną tego parametru przyjęto jako $p_{10} = T^{death} \in [90.0, 110.0]$ (war ref. 100.0)

W niniejszym eksperymencie przyjęto zatem, iż początkowa seria wartości tego parametru (wartości tego parametru dla populacji początkowej alg ewolucyjnego) losowana będzie z przedziału $p'_{10} \in [110.0, 150.0]$ Natomiast dla osobników generowanych już w trakcie działania alg. ewolucyjnego jako dopuszczalny zakres wartości parametru p_{10} przyjęto zakres $p_{10} \in [50.0, 150.0]$

5.1.4.2 Otrzymane wyniki

Eksperyment w trakcie realizacji

Eksperymentu ostatecznie nie wykonano

5.2 Kwestie do zweryfikowania

- czy zaburzanie realizowane jak powyżej - ok?
- czy przyjmowana siła zaburzenia / zakres zmian parametru w pop początkowej ok/wystarczający?
- czy parametry ewolucji (ilość ewaluacji / długość trwania ok)?
- czy zakres gromadzonych / wizualizowanych wyników wystarczający? (może potrzebne coś jeszcze? czas obliczeń? masa guza? . . .)

6 Eksperyment z fixowaniem wartosci parametrow niepodlegajacych ewolucji/poszukiwaniu

6.1 Sposób przeprowadzenia eksperymentow

W ponizszej serii eksperymentow podobnie jak wczesniej stopniowo zwiększamy ‘stopień trudności’ tzn zaburzeniu ulega stopniowo (w stosunku do zakresu referencyjnego) rosnąca ilość parametrów (tych przyjętych na wstępnym etapie weryfikacji jako istotne czyli odpowiednio: $p_7 = o^{prol}$, $p_8 = o^{death}$, $p_9 = T^{prol}$ oraz $p_{10} = T^{death}$) czyli:

- zaburzeniu podlega 1 parameter (4 symulacje po jednej dla ‘indywidualnie’ zaburzonych parametrów $p_7 \dots p_{10}$),
- zaburzeniu podlegaja 2 parametry (6 symulacji dla roznych par zaburzonych parametrów)
- zaburzeniu podlegaja 3 parametry (4 symulacje dla jednego parametru niezaburzonego)
- zaburzeniu podlegaja 4 parametry (1 symulacja dla wszystkich zaburzonych)

Z tym ze tym razem poza zaburzanymi i ewoluowanymi parametrami wszystkie pozostale byly “fixed”. Czyli jesli zaburzamy pojedynczy parametr p_7 to wszystkie pozostale ($p_8 \dots p_{10}$ powinny zostac “zafiksowane” na swoich wartosciach referencyjnych i nie byc poddawane ewolucji/poszukiwaniom.)

Podobnie jesli zaburzaniu ma podleac para (np p_8 i p_{10}) to pozostale dwa bedace przedmiotem zainteresowania powinny zostac zafiksowane na swoich wartosciach referencyjnych.

6.2 Problemy na jakie natrafiono

Niestety po wprowadzeniu do definicji solwera ewolucyjnego modyfikacji pozwalajacych prowadzic eksperymenty zgodnie z powyzzszym opisem nadziano sie na problemy z dzialaniem solwera. Wydaje sie ze problemy mialy podobna nature jak poprzednio - przy jakiejjs kombinacji parametrow nastepowal zbyt duzy rozrost (mapy) zyl co powodowalo, albo bardzo znaczne wydłużenie obliczen albo “wywrocenie” sie solwera z uwagi na brak mozliwosci dalszego alokowania pamieci etc.

```
Unable to run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454590847505983
2.9222689984674073 9.063008773943382 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.430679318110801
2.9613064430274214 9.001731455509889 94.84556414555985 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.312816143866291
2.8373159607698737 9.068744087595107 109.4082352110131 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.45412400070996
2.9512270566632774 9.134113928012875 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.506700280337986
2.6729585594183844 9.127711756586537 98.28336497095415 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.428205108285592
2.6603929941110316 9.0677395656585 97.18615629884131 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.477124985184965
2.7878387123218205 9.071758596198237 108.77922017134355 0.001 0.3 0.625 0.3205 0.0064 0.0064
```

```

0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Read from
err: terminate Read from err: called Read from err: after Read from err: throwing Read from
err: an Read from err: instance Read from err: of Read from err: 'std::bad_alloc' Read from
err: what(): Read from err: std::bad_alloc Unable to run solver: java.lang.NumberFormatException:
For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.45300605302644
2.991957323550271 9.06939941685454 96.1524747998914 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.424280747231732
2.816132673178266 9.080270122482826 109.95693142895279 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.454330657752786
2.6073509233841365 9.065938202205565 109.70008853564136 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.4 Unable to
run solver: java.lang.NumberFormatException: For input string: "=====
mpirun -np 1 -host node13:1 ../TumorGenetyka/tumor 2 80 9900 0.1 1000 0.5 9.729087704398712
2.9222689984674073 9.146183592522865 98.30290819602372 0.001 0.3 0.625 0.3205 0.0064 0.0064
0.0000555 0.01 0.0000555 0.01 0.4 0.5 0.05 0.3 0.01333 10 0.003 2 5 25 24 0.003 0.

```

6.3 Zmiany wprowadzone do solwera

Aby uniknąć tego typu problemów, wprowadzono w solwerze następującą zmianę

Dodano kod który uzależnia wzrost żył od aktualnego rozmiaru już istniejącego układu żył. Model żył działa w ten sposób, że w określonych warunkach jest pewne prawdopodobieństwo że coś nowego się utworzy, zmiana polega na tym że to prawdopodobieństwo jest mnożone przez liczbę `inhibit_factor` z przedziału $[0, 1]$, która jest równa 1 dopóki ilość wierzchołków + żył jest mniejsza niż 10,000 (czyli wtedy prawdopodobieństwo utworzenia nowych jest takie samo jak przed tą zmianą), a jak jest ta ilość większa to odpowiednio `inhibit_factor` jest mniejszy, dla układu z milionem+ staje się zerem. To powinno dość sztywno ograniczyć rozrost niezależnie od parametrów, jednocześnie nie zmieniając wiele dla “rozsądnych” symulacji gdzie tych żył się nie robią miliony.

6.4 Aktualna postać funkcji celu

Po powyższych zmianach, ponieważ zmienia to model, przeliczono ponownie wartości uzyskiwane przez solwer dla referencyjnych wartości parametrów. Wartość ta aktualnie to 397321

A zatem funkcja fitness to aktualnie:

$$abs(397321 - tumor_{volume}) \quad (2)$$

Wartość ta była minimalizowana podczas ewolucji.

6.5 Zaburzenie 4 parametrów

6.5.1 Zmienność parametrów

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_7	$p_7 = 10 \pm 10\%$	$p_7 \in [11.0, 15.0]$	$p_7 \in [5.0, 15.0]$
p_8	$p_8 = 2 \pm 10\%$	$p_8 \in [2.1, 3.0]$	$p_8 \in [1.0, 3.0]$
p_9	$p_9 = 10 \pm 10\%$	$p_9 \in [11.0, 15.0]$	$p_9 \in [5.0, 15.0]$
p_{10}	$p_{10} = 100 \pm 10\%$	$p_{10} \in [110.0, 150.0]$	$p_{10} \in [50.0, 150.0]$

6.5.2 Rozkład P7

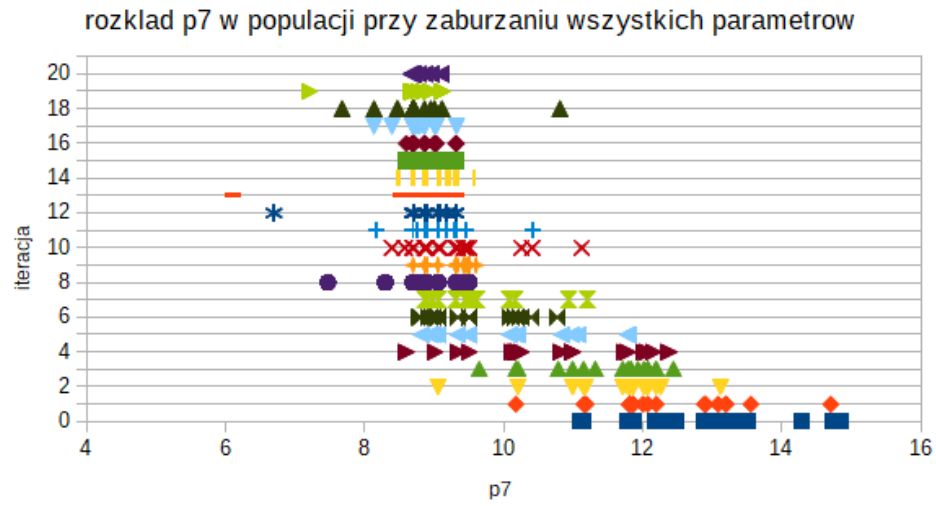


Figure 10: Rozkład p_7 w populacji przy zaburzaniu 4 parametrów

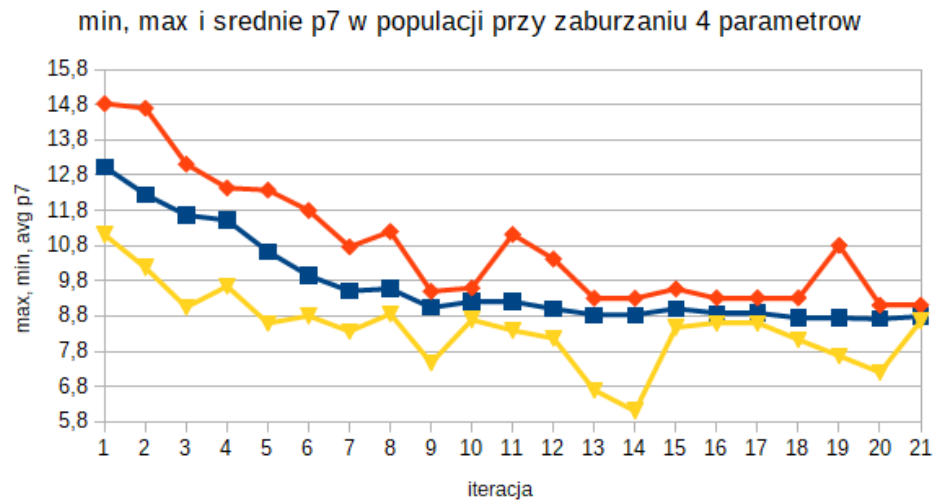


Figure 11: Wartości średnie, maksymalne oraz minimalne p7 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow

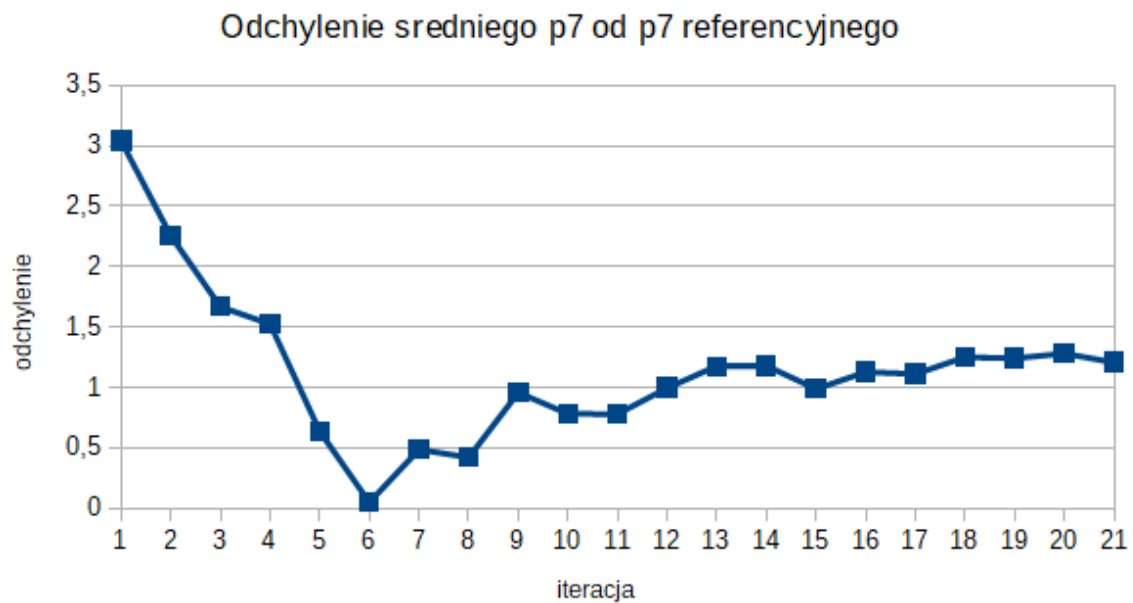


Figure 12: Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu 4 parametrow

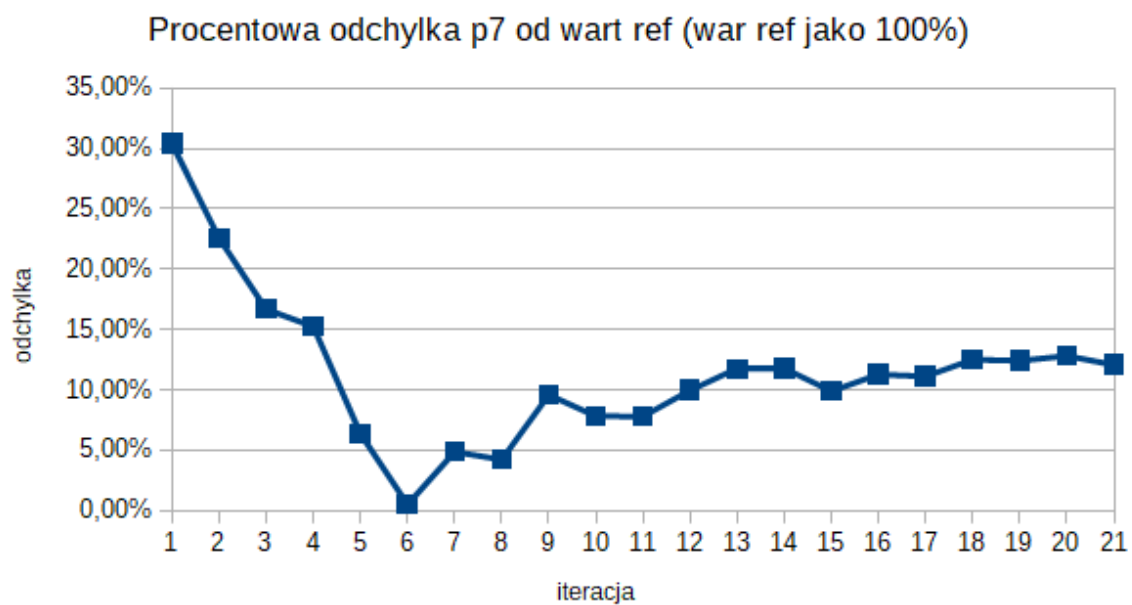


Figure 13: Procentowa Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu 4 parametrów

6.5.3 Rozkład P8

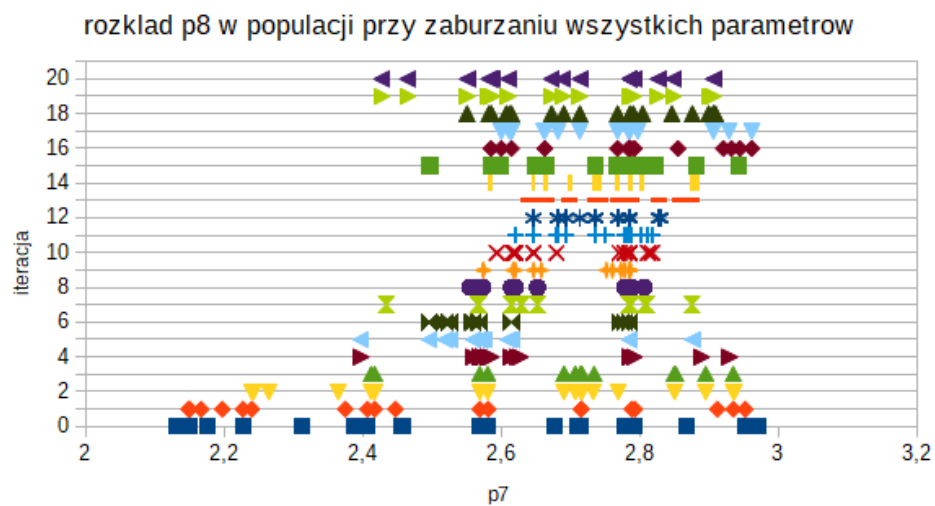


Figure 14: Rozkład p_8 w populacji przy zaburzeniu 4 parametrow

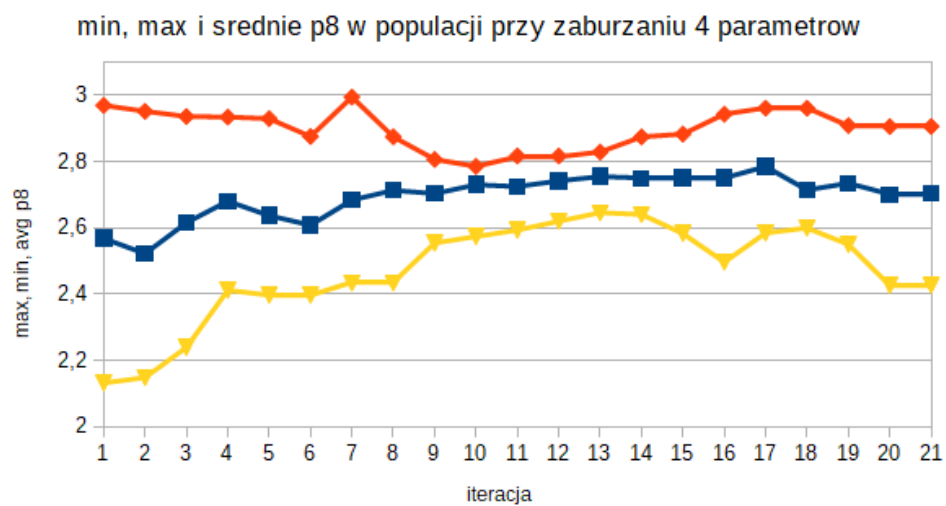


Figure 15: Wartości średnie, maksymalne oraz minimalne p_8 na kolejnych etapach działania alg ewolucyjnego przy zaburzeniu 4 parametrow

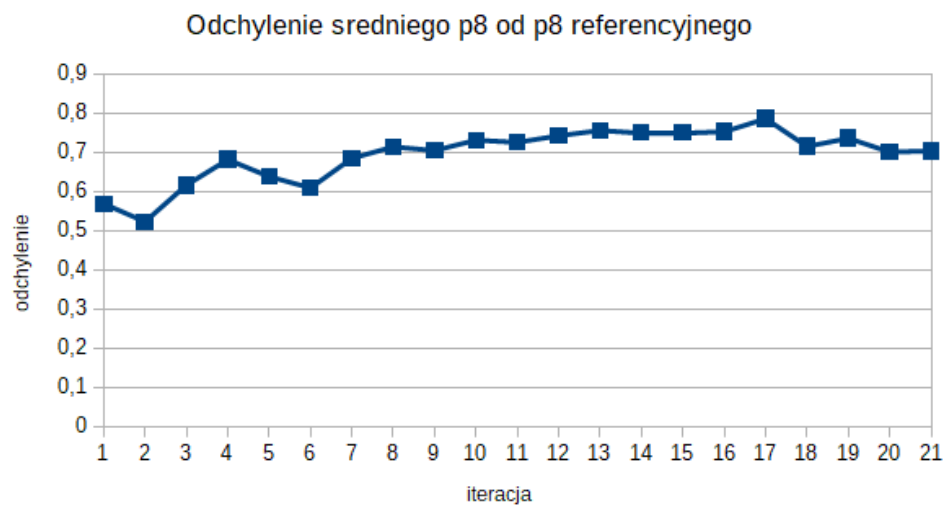


Figure 16: Odchyłka sredniej wartosci p8 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

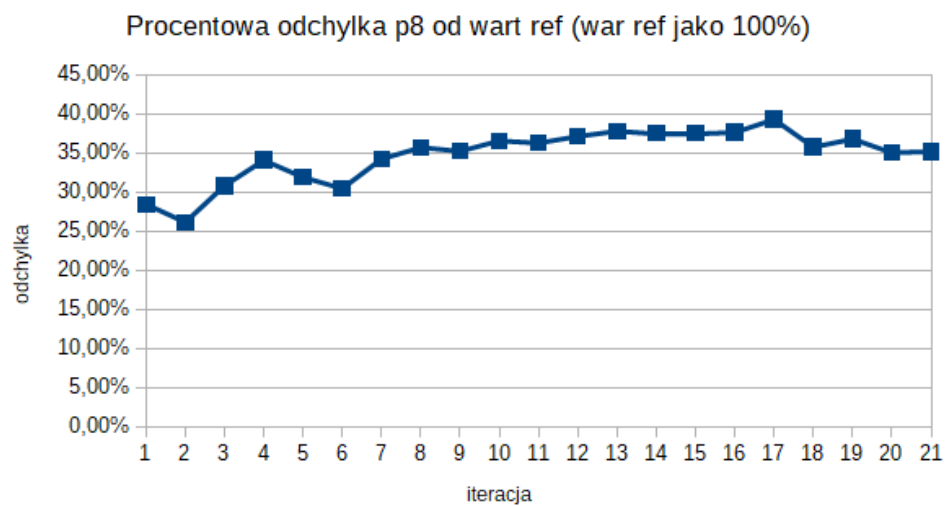


Figure 17: Procentowa Odchyłka sredniej wartosci p8 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

6.5.4 Rozkład P9

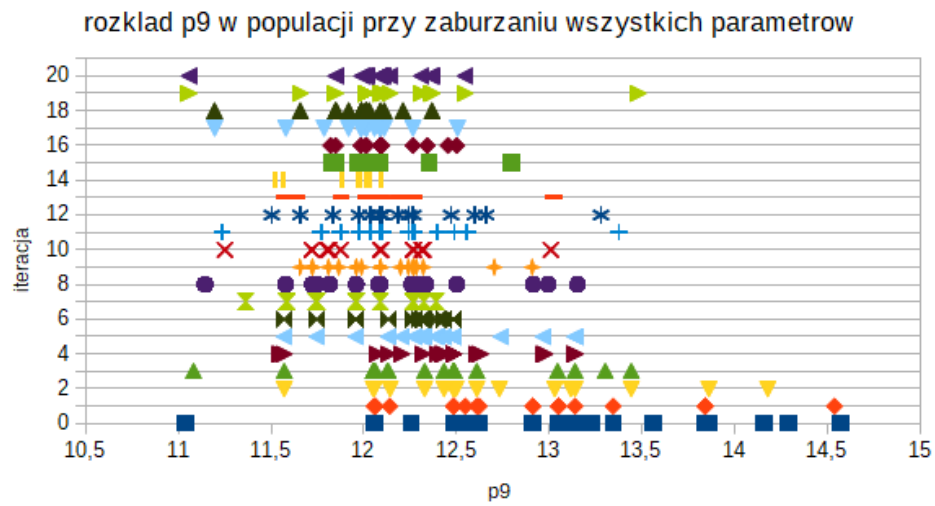


Figure 18: Rozkład p_9 w populacji przy zaburzaniu 4 parametrow

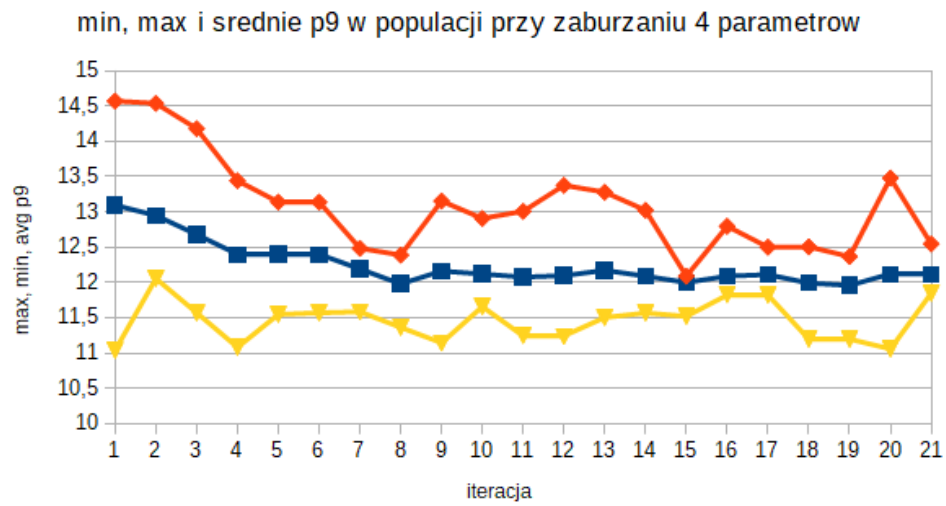


Figure 19: Wartości średnie, maksymalne oraz minimalne p_9 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow

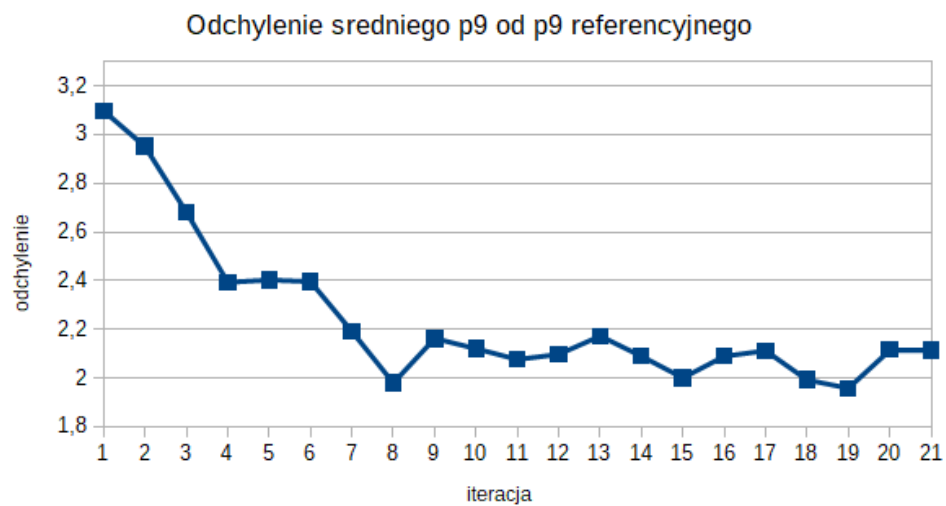


Figure 20: Odchyłka sredniej wartosci p9 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

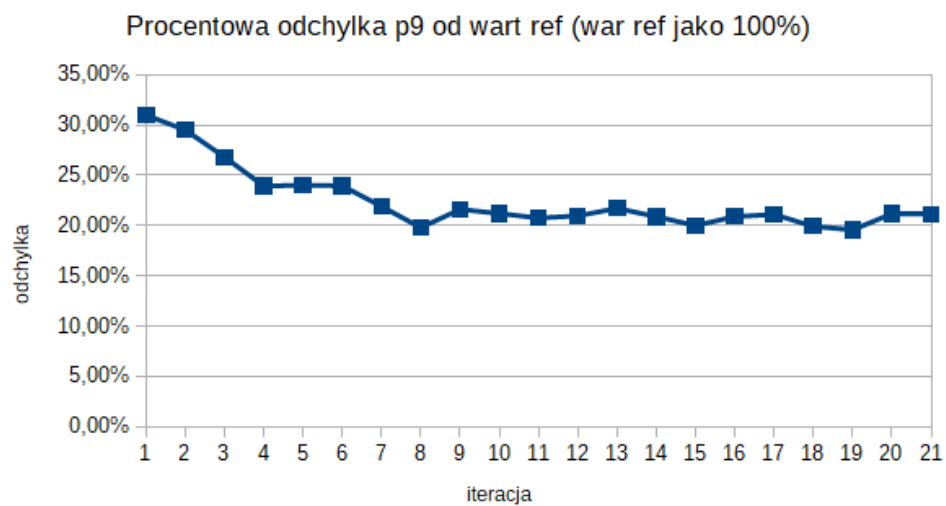


Figure 21: Procentowa Odchyłka sredniej wartosci p9 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

6.5.5 Rozkład P10

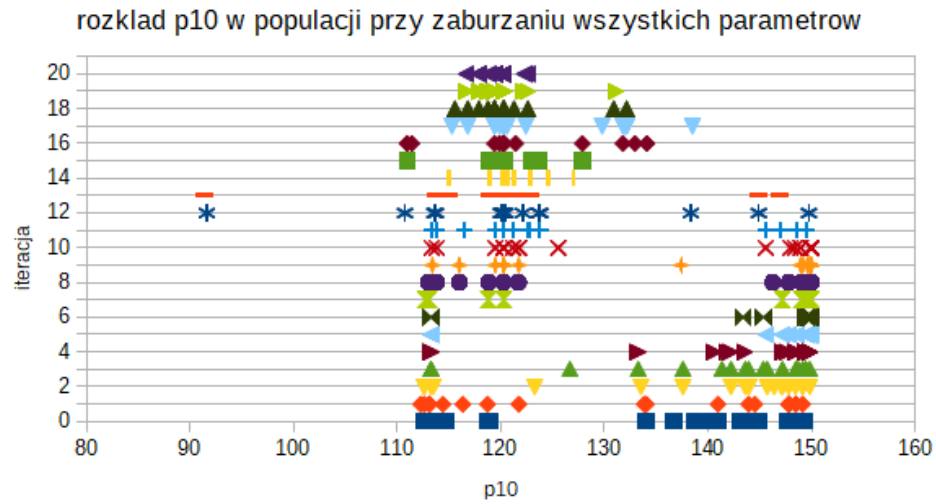


Figure 22: Rozkład p_{10} w populacji przy zaburzaniu 4 parametrow

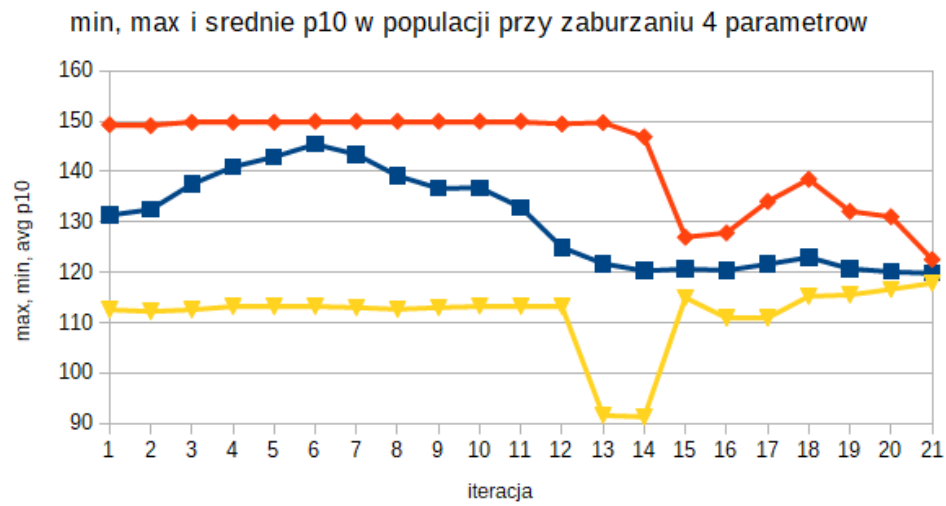


Figure 23: Wartości średnie, maksymalne oraz minimalne p_{10} na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow

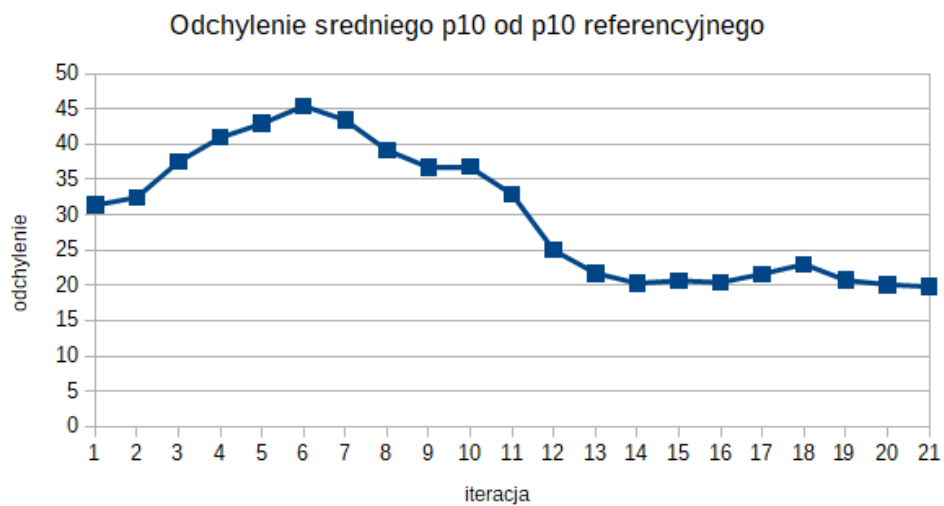


Figure 24: Odchyłka sredniej wartosci p10 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

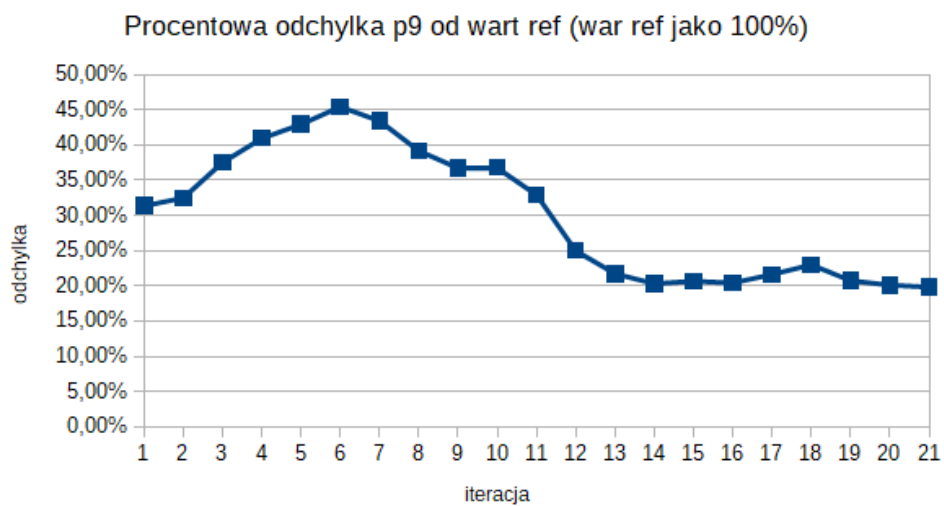


Figure 25: Procentowa Odchyłka sredniej wartosci p10 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

6.5.6 Rozkład TumorVolume

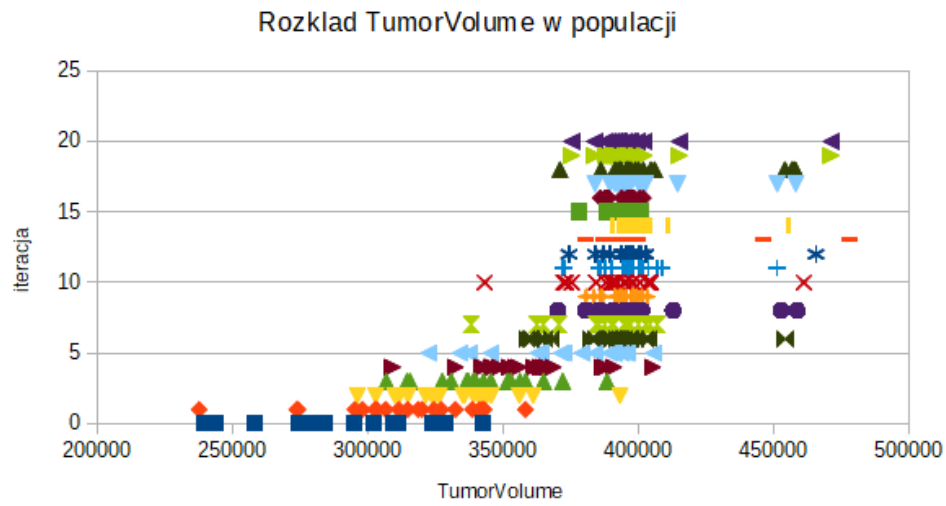


Figure 26: Rozkład *TumorVolume* w populacji przy zaburzaniu 4 parametrow

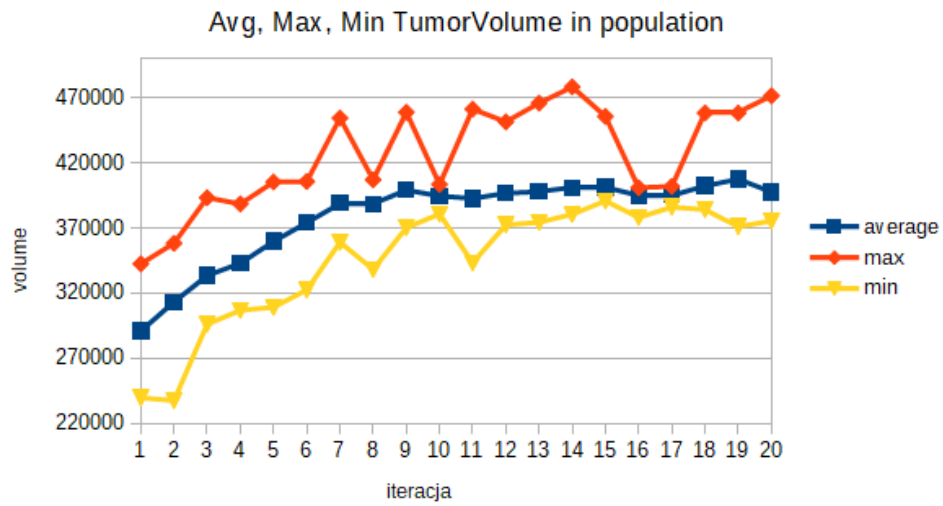


Figure 27: Wartości średnie, maksymalne oraz minimalne *TumorVolume* na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow

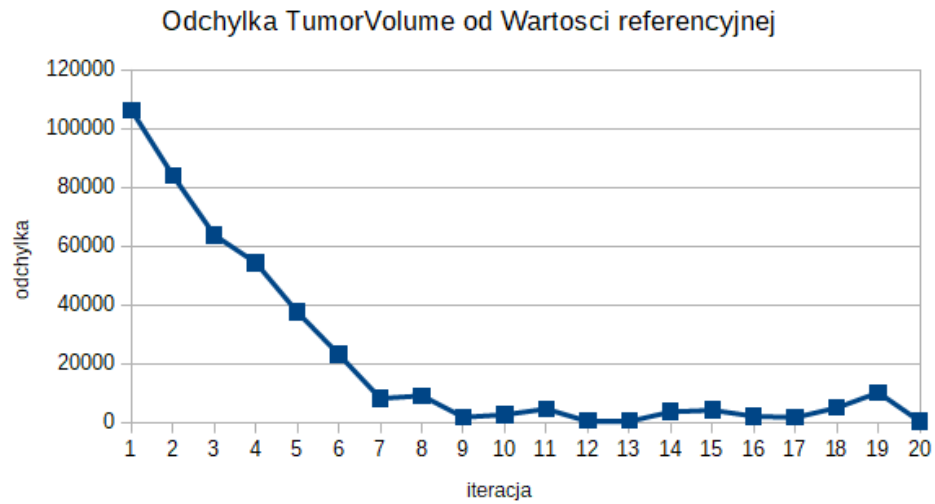


Figure 28: Odchylka sredniej wartosci TumorVolume w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

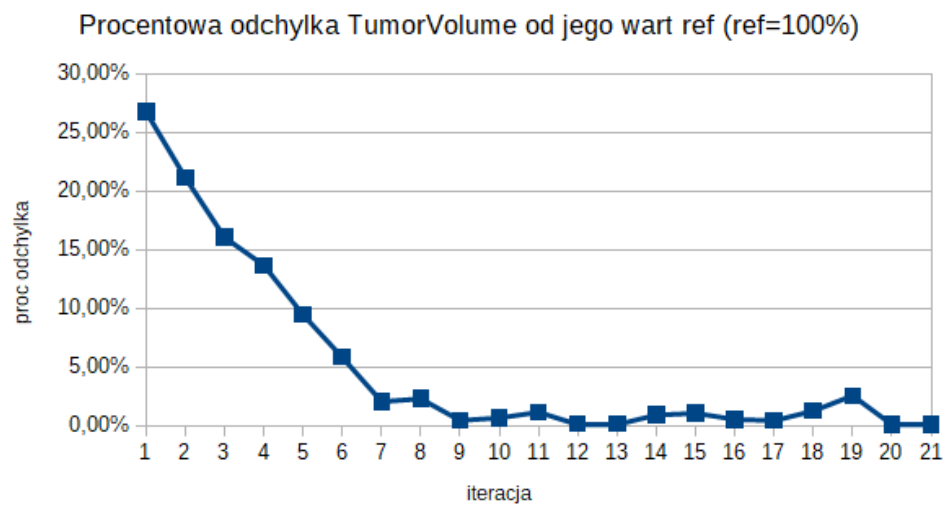


Figure 29: Procentowa Odchylka sredniej wartosci TumorVolume w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow

6.5.7 Wnioski

Ponizej odchyłki wszystkich 4 parametrow i tumorVolume od ich wartosci referencyjnych.

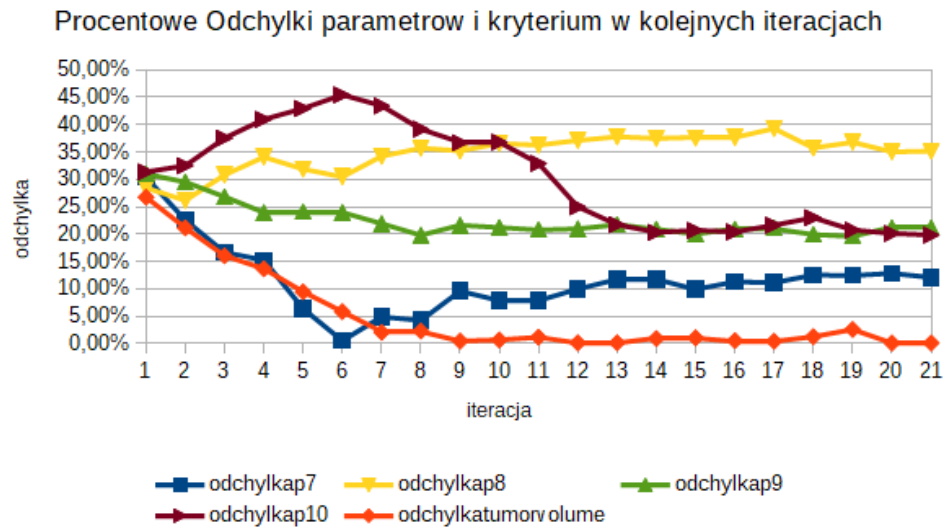


Figure 30: Procentowa Odchyłka poszczególnych parametrow i TumorVolume od wart referencyjnych

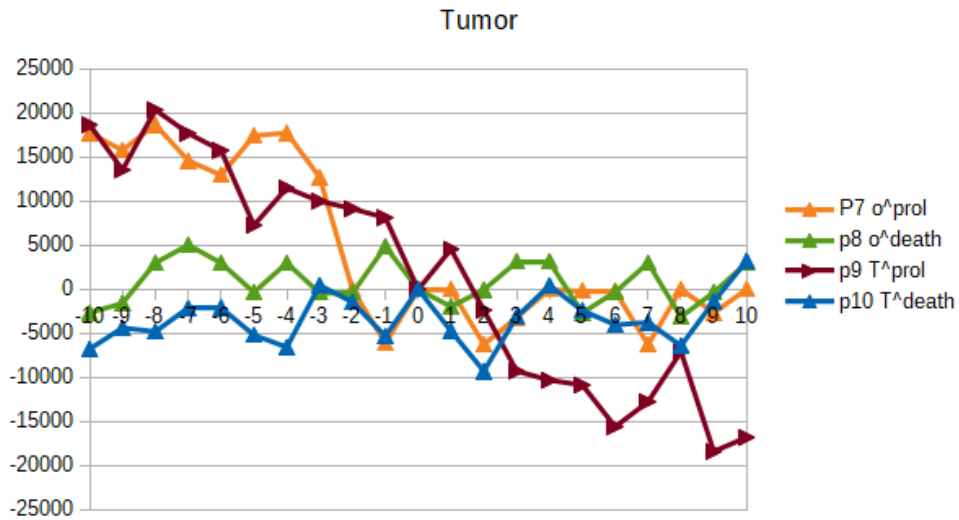


Figure 31: Związek pomiędzy wartościami parametrow p_7 , p_8 , p_9 i p_{10} a objetoscia guza

Zbieznosc niby jest, ale nie wszedzie i niekoniecznie do wartosci referencyjnych

O ile odchyłka rozmiaru guza od wartosci referencyjnej zmierza do 0 to juz od wartosciach poszczegolnych parametrow powiedziec tego nie mozna.

Co wiecej w przypadku p8 ta odchyłka wrecz rosnie :) ale - brak zbieznosci w przypadku p8 jest w zasadzie oczekiwany bo jak sie popatrzy na wykres zaleznosci tumorvolume z wartoscia tego parametru to nie widac tam trendu. Wraz ze zmianami p8 wartosci tumorvolume owszem zmienia sie ale bez jakiegos specjalnego (a w zasadzie zadnego) trendu - wiec i ewolucji trudno bylo podazac pewnie z p8 w jakims konkretnym kierunku.

- Troche zastanawiajace jest dlaczego efekt jest nieco inny w przypadku p10 skoro zaleznosc pomiedzy tumorVolume a tym parametrem byla podobna do zaleznosci od p8
- to be continued....

6.6 Zaburzenie 4 parametrów - wydłużone działanie

6.6.1 Zmienność parametrów

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_7	$p_7 = 10 \pm 10\%$	$p_7 \in [11.0, 15.0]$	$p_7 \in [5.0, 15.0]$
p_8	$p_8 = 2 \pm 10\%$	$p_8 \in [2.1, 3.0]$	$p_8 \in [1.0, 3.0]$
p_9	$p_9 = 10 \pm 10\%$	$p_9 \in [11.0, 15.0]$	$p_9 \in [5.0, 15.0]$
p_{10}	$p_{10} = 100 \pm 10\%$	$p_{10} \in [110.0, 150.0]$	$p_{10} \in [50.0, 150.0]$

Zmiana w tym eksperymencie w stosunku do poprzednio opisywanego polegała na wydłużeniu działania alg ewolucyjnego do 600 ewaluacji (600 wywołań solwera)

6.6.2 Rozkład P7

rozkład p_7 w populacji przy zaburzaniu wszystkich parametrów 600 ewaluacji

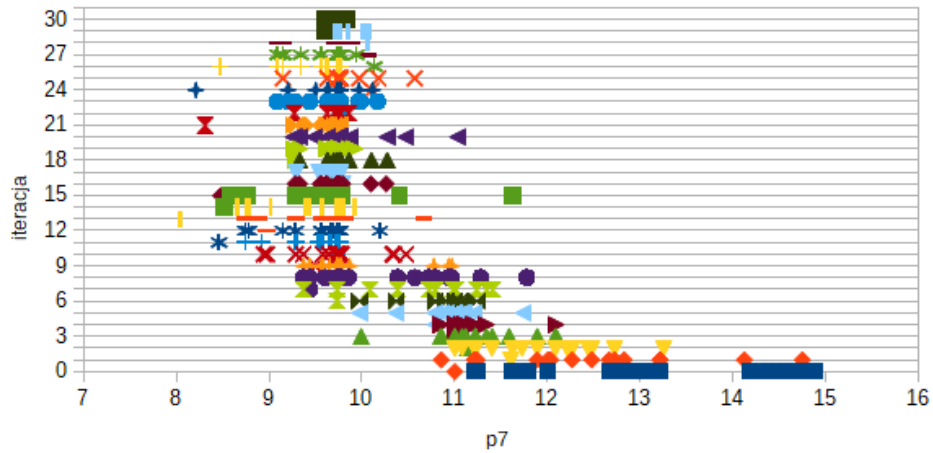


Figure 32: Rozkład p_7 w populacji przy zaburzaniu 4 parametrów dla 600 ewaluacji

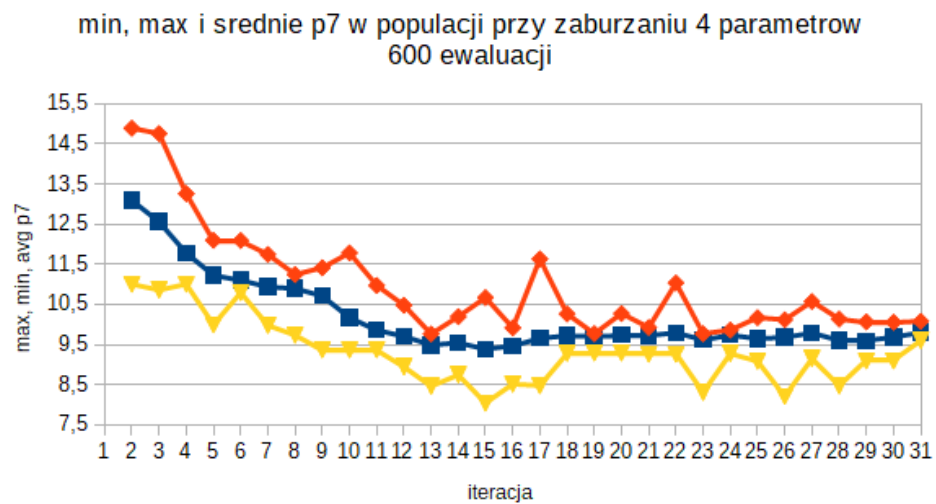


Figure 33: Wartości średnie, maksymalne oraz minimalne p7 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow dla 600 ewaluacji

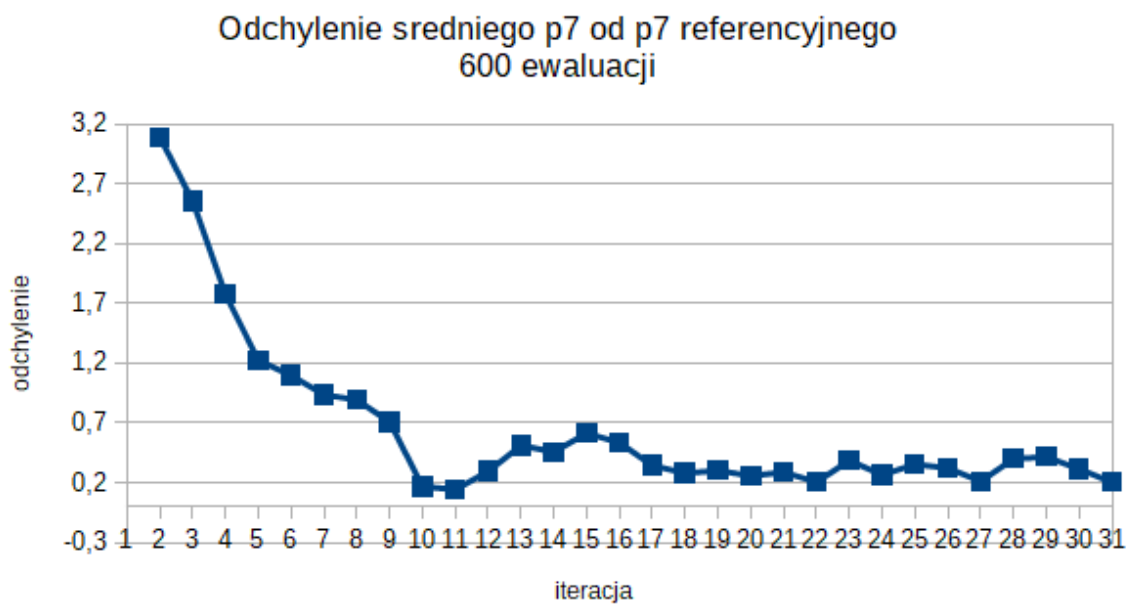


Figure 34: Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

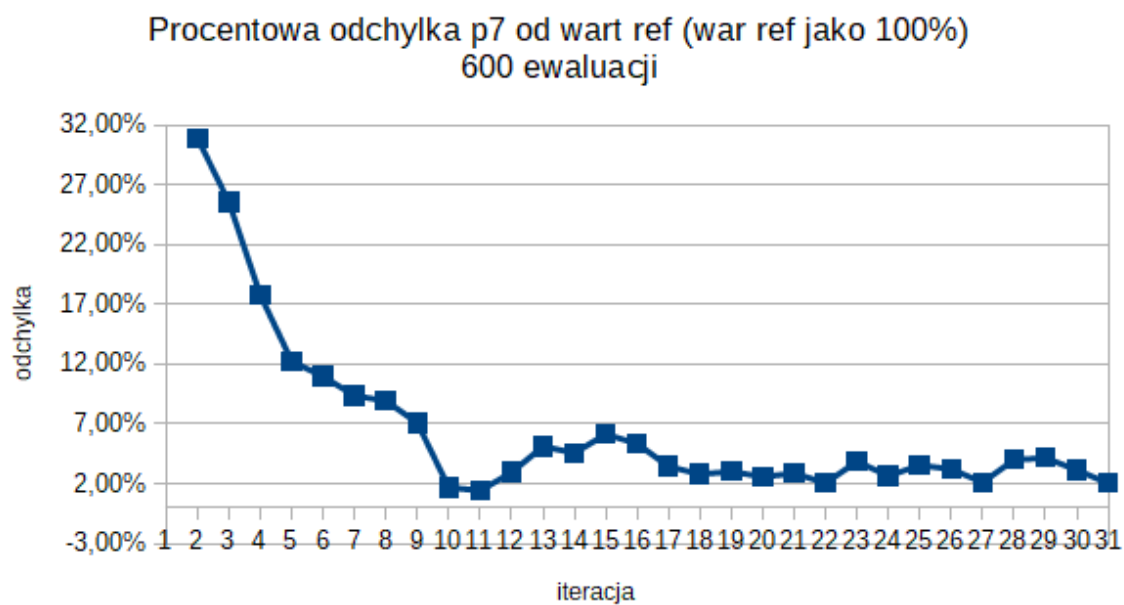


Figure 35: Procentowa Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu 4 parametrów

6.6.3 Rozkład P8

rozklad p7 w populacji przy zaburzaniu wszystkich parametrow 600 ewaluacji

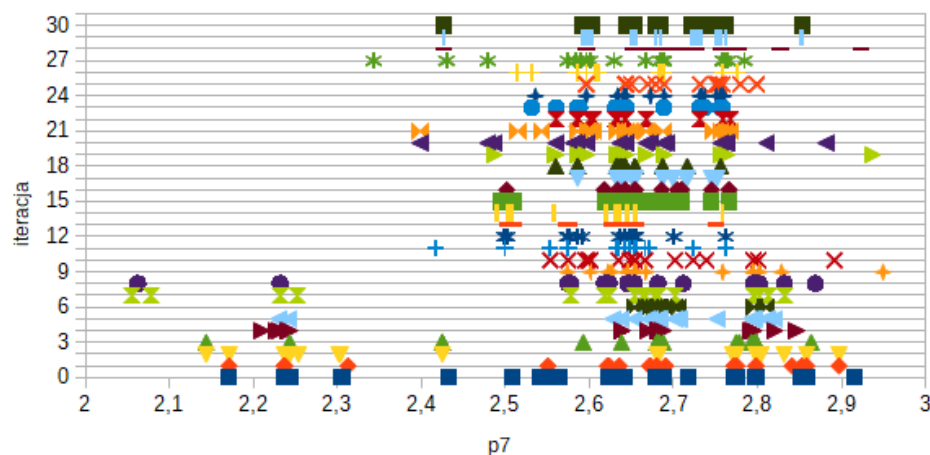


Figure 36: Rozklad p_8 w populacji przy zaburzaniu 4 parametrow dla 600 ewaluacji

min, max i srednie p7 w populacji przy zaburzaniu 4 parametrow
600 ewaluacji

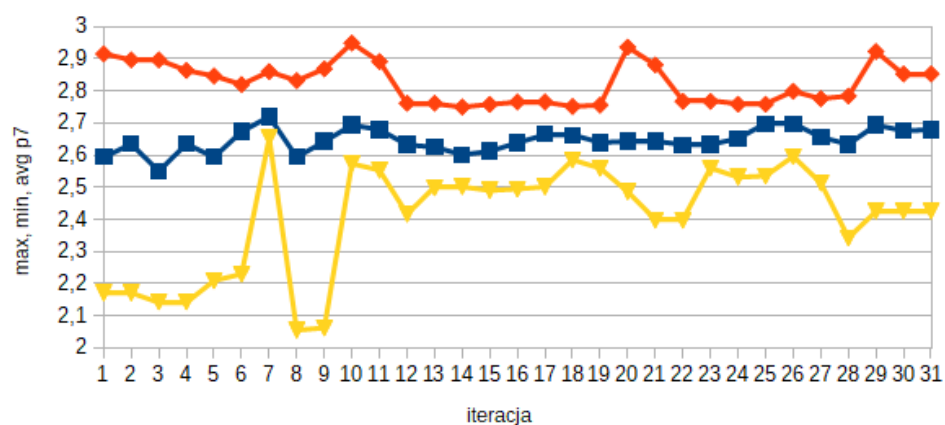


Figure 37: Wartości średnie, maksymalne oraz minimalne p_8 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow dla 600 ewaluacji

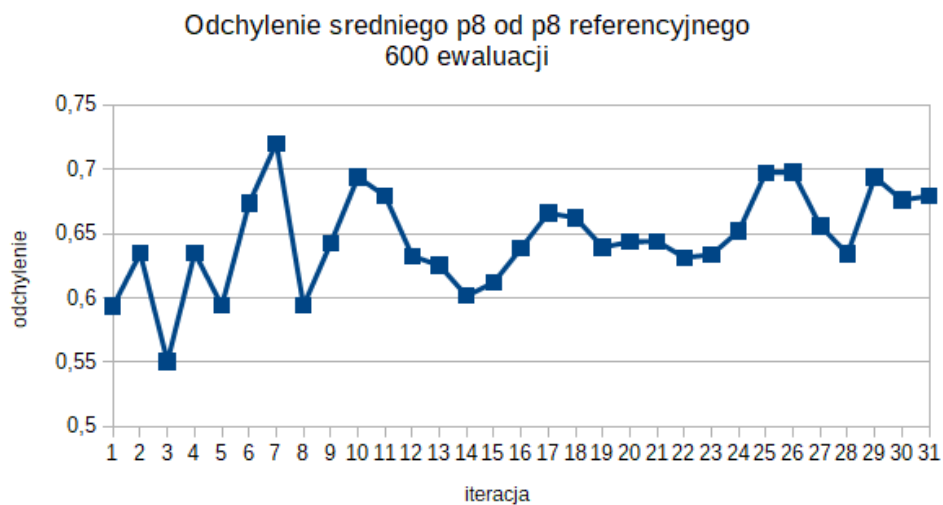


Figure 38: Odchyłka sredniej wartosci p8 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

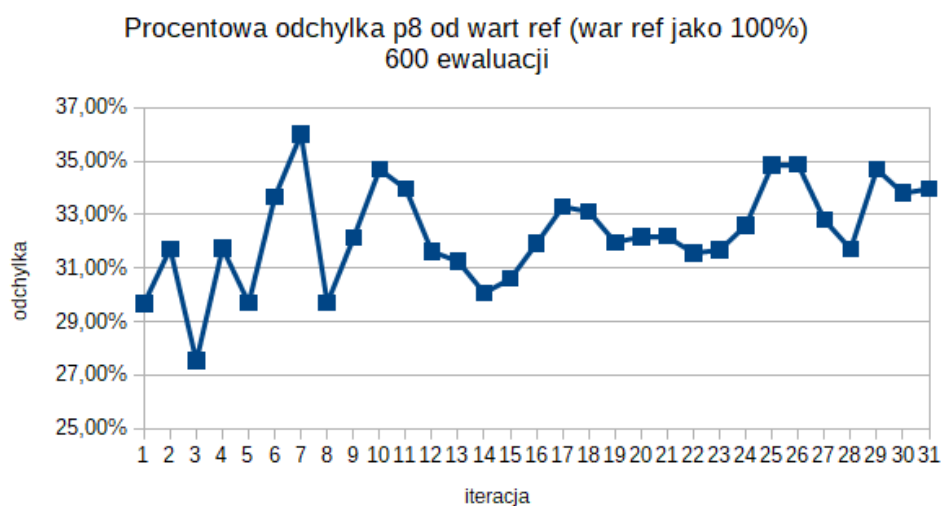


Figure 39: Procentowa Odchylka sredniej wartosci p8 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

6.6.4 Rozkład p_9

rozkład p_9 w populacji przy zaburzaniu wszystkich parametrow 600 ewaluacji

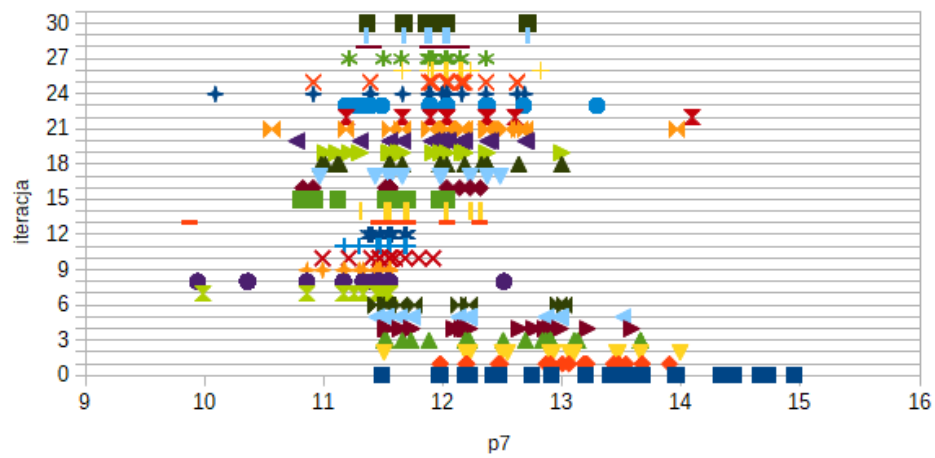


Figure 40: Rozkład p_9 w populacji przy zaburzaniu 4 parametrow dla 600 ewaluacji

min, max i srednie p_9 w populacji przy zaburzaniu 4 parametrow
600 ewaluacji

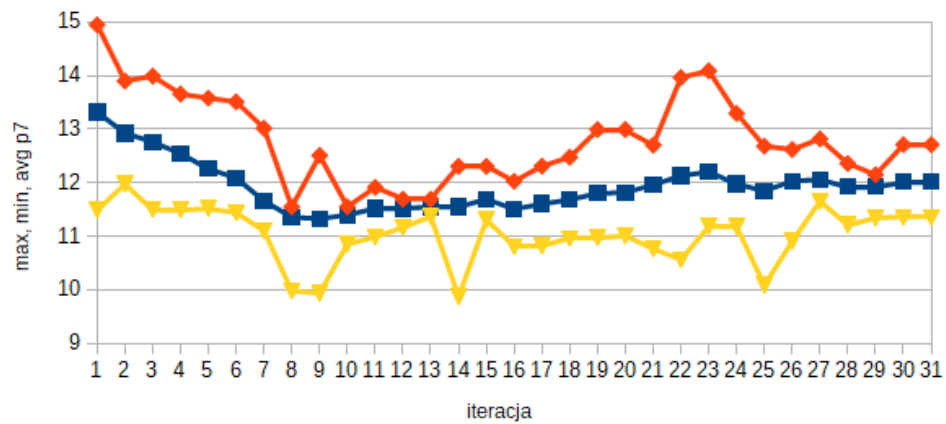


Figure 41: Wartości średnie, maksymalne oraz minimalne p_9 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow dla 600 ewaluacji

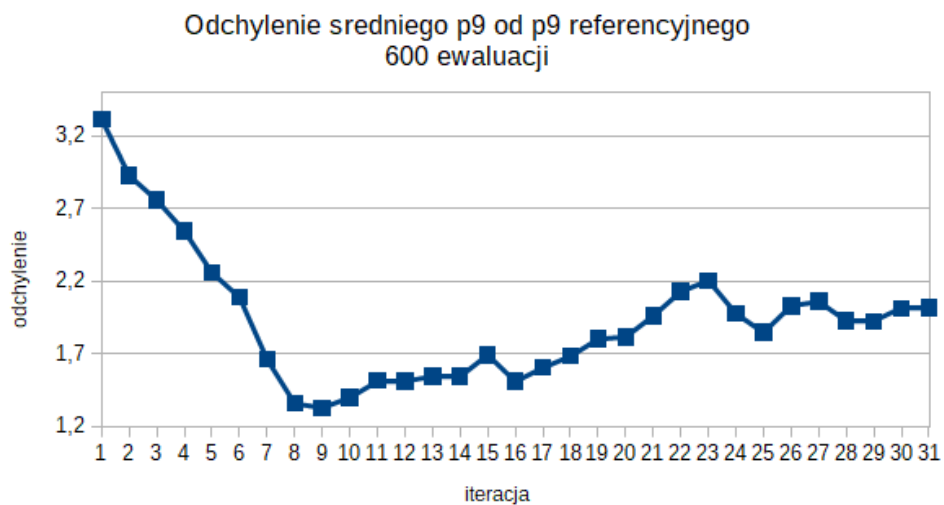


Figure 42: Odchyłka sredniej wartosci p9 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

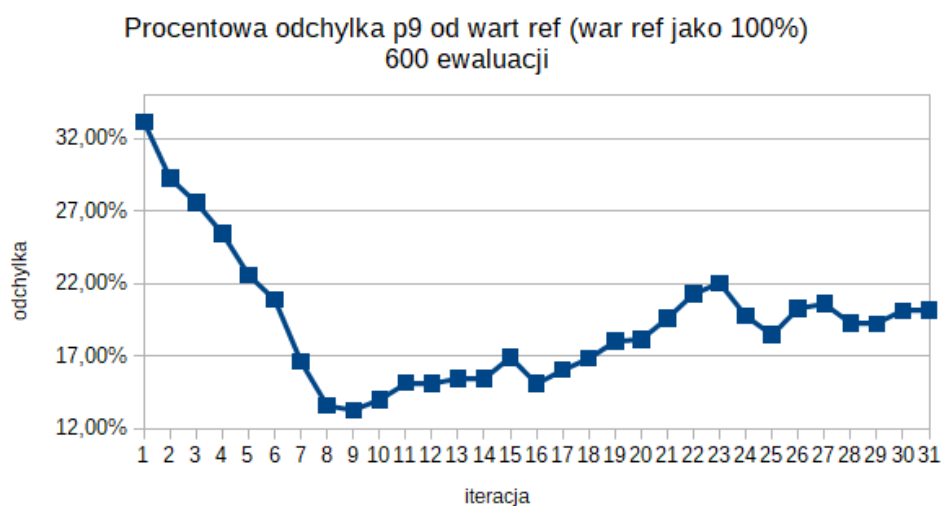


Figure 43: Procentowa Odchylka sredniej wartosci p9 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

6.6.5 Rozkład P10

rozkład p10 w populacji przy zaburzaniu wszystkich parametrow 600 ewaluacji

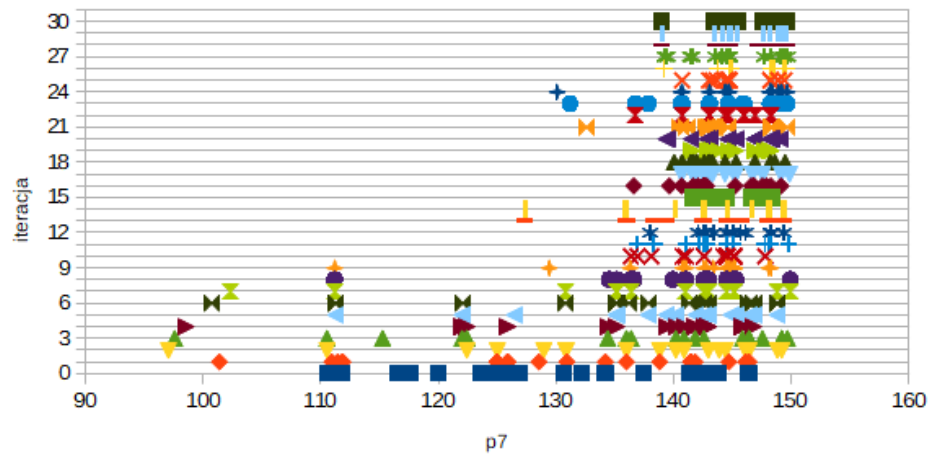


Figure 44: Rozkład p_{10} w populacji przy zaburzaniu 4 parametrow dla 600 ewaluacji

min, max i srednie p10 w populacji przy zaburzaniu 4 parametrow
600 ewaluacji

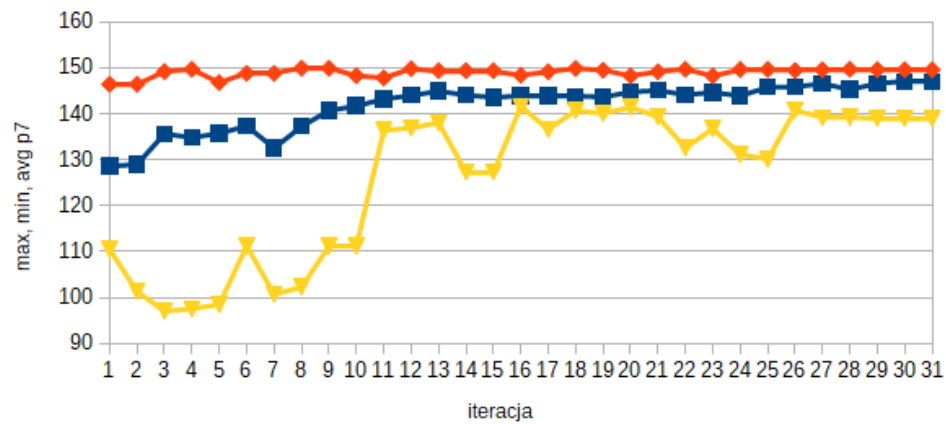


Figure 45: Wartości średnie, maksymalne oraz minimalne p_{10} na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow dla 600 ewaluacji

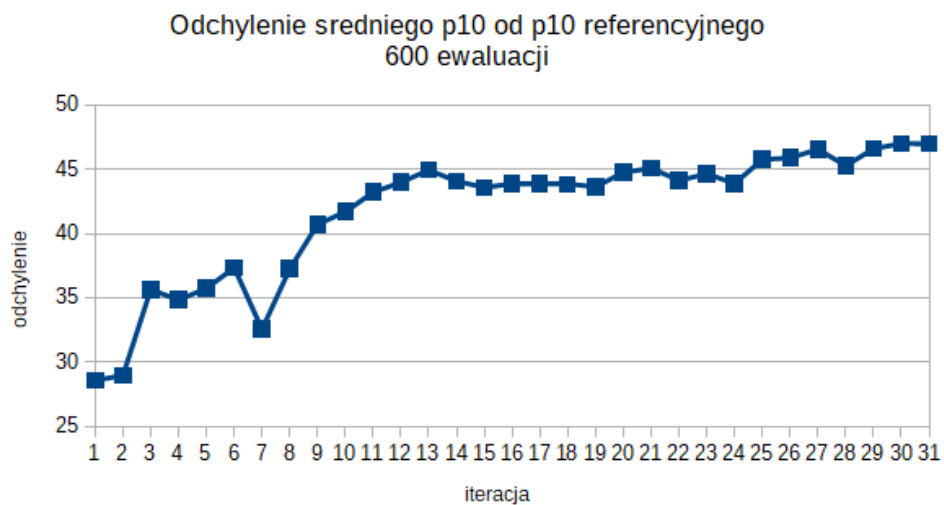


Figure 46: Odchyłka sredniej wartosci p10 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

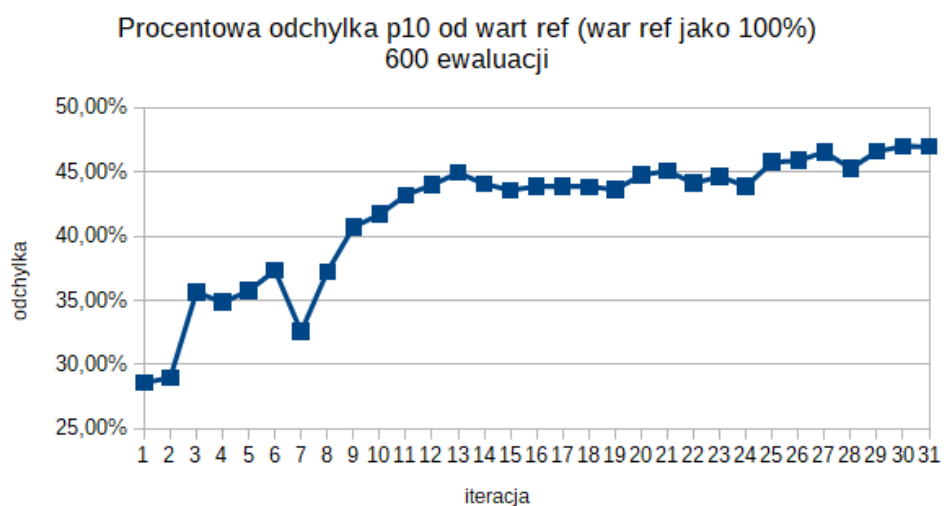


Figure 47: Procentowa Odchyłka sredniej wartosci p10 w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

6.6.6 Rozkład TumorVolume

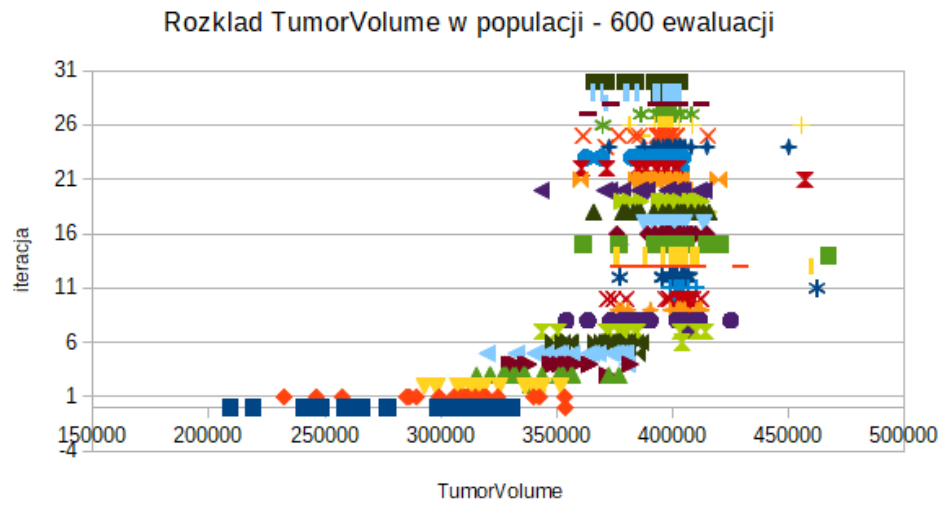


Figure 48: Rozkład *TumorVolume* w populacji przy zaburzaniu 4 parametrow dla 600 ewaluacji

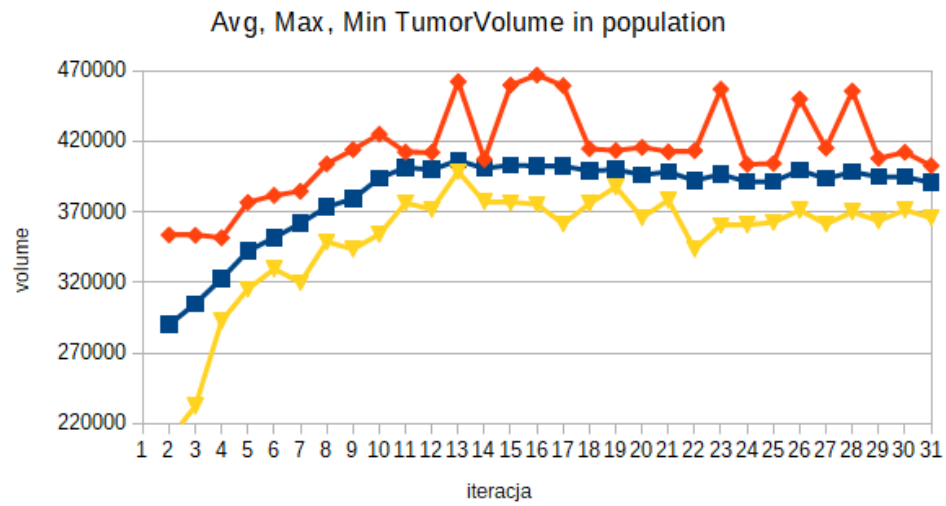


Figure 49: Wartości średnie, maksymalne oraz minimalne *TumorVolume* na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu 4 parametrow dla 600 ewaluacji

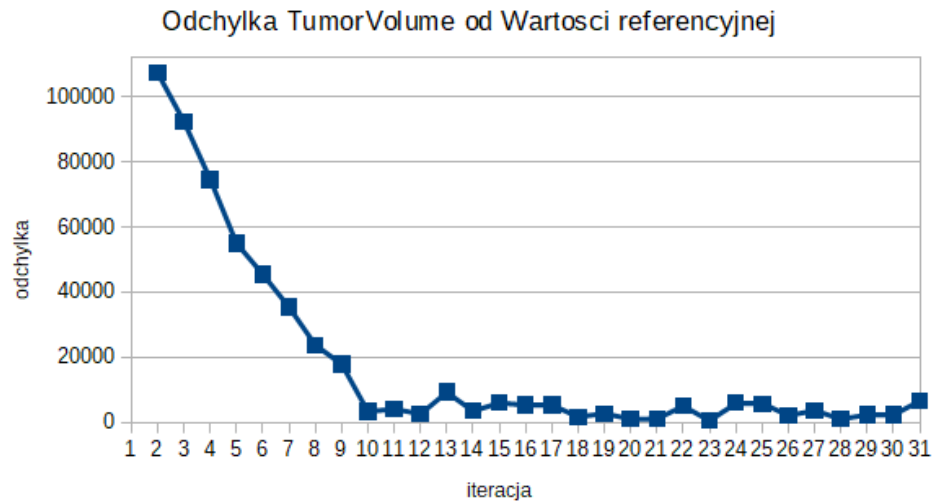


Figure 50: Odchylka sredniej wartosci TumorVolume w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

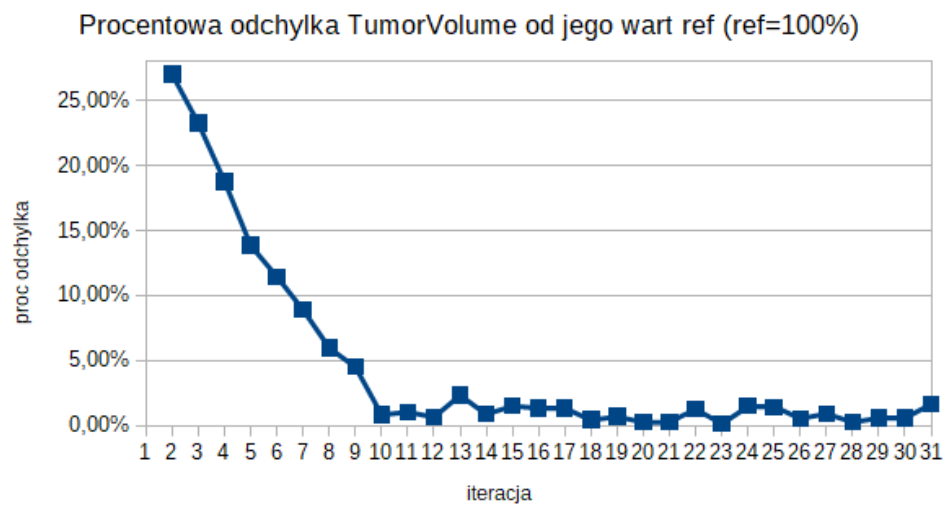


Figure 51: Procentowa Odchylka sredniej wartosci TumorVolume w populacji od wartosci referencyjnej przy zaburzaniu 4 parametrow dla 600 ewaluacji

6.6.7 Wnioski

Ponizej odchyłki wszystkich 4 parametrow i tumorVolume od ich wartosci referencyjnych.

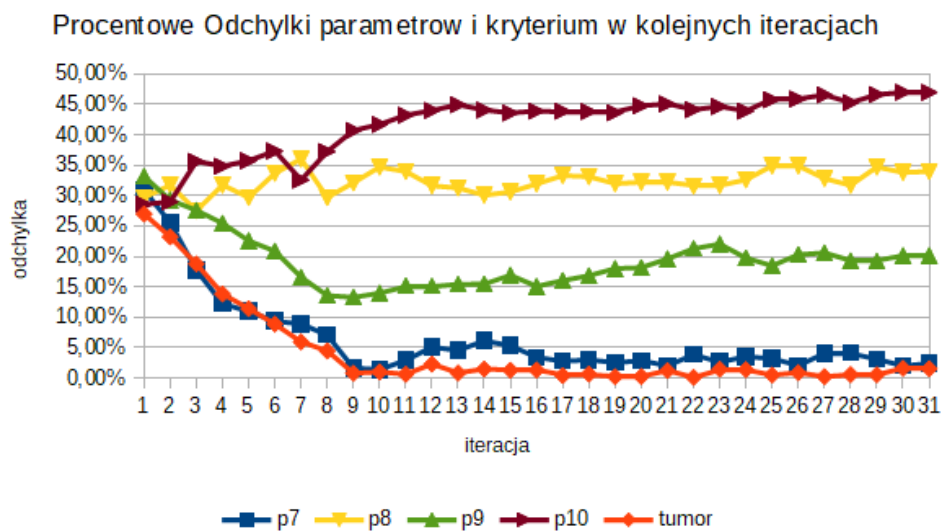


Figure 52: Procentowa Odchyłka poszczegolnych parametrow i TumorVolume od wart referencyjnych dla 600 ewaluacji

6.7 Zaburzanie 1 parametru

6.7.1 Zaburzanie P7

6.7.1.1 Zmienność parametru

Parametr	Zakres zmienności podczas wstępnej ewaluacji	Zakres zmienności dla populacji początkowej	Zakres zmienności w trakcie trwania ewolucji
p_7	$p_7 = 10 \pm 10\%$	$p_7 \in [11.0, 15.0]$	$p_7 \in [5.0, 15.0]$
p_8	$p_8 = 2 \pm 10\%$	$p_8 = 2$	$p_8 = 2$
p_9	$p_9 = 10 \pm 10\%$	$p_9 = 10$	$p_9 = 10$
p_{10}	$p_{10} = 100 \pm 10\%$	$p_{10} = 100$	$p_{10} = 100$

6.7.1.2 Rozkład p7

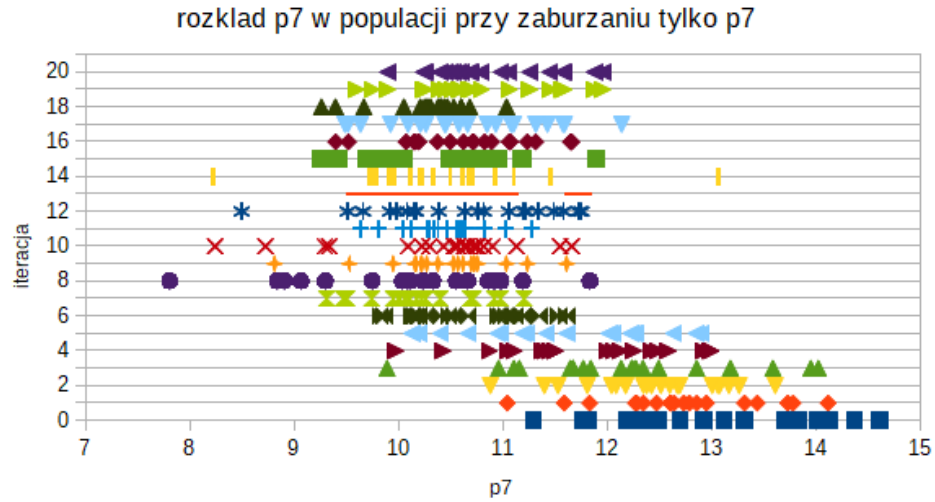


Figure 53: Rozkład p_7 w populacji przy zaburzaniu 1 parametru (P7)

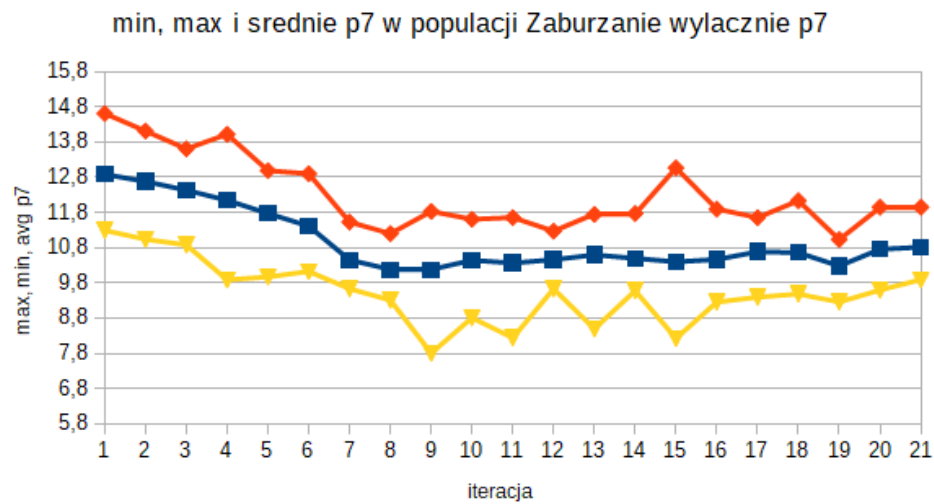


Figure 54: Wartości średnie, maksymalne oraz minimalne p7 na kolejnych etapach działania alg ewolucyjnego przy zaburzaniu p7

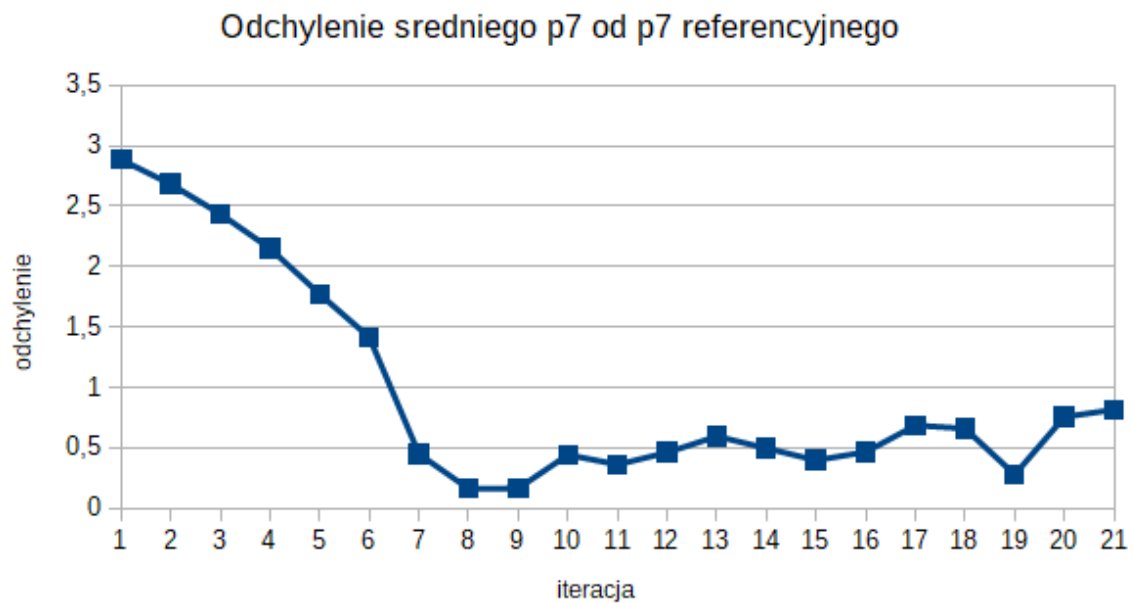


Figure 55: Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu p7

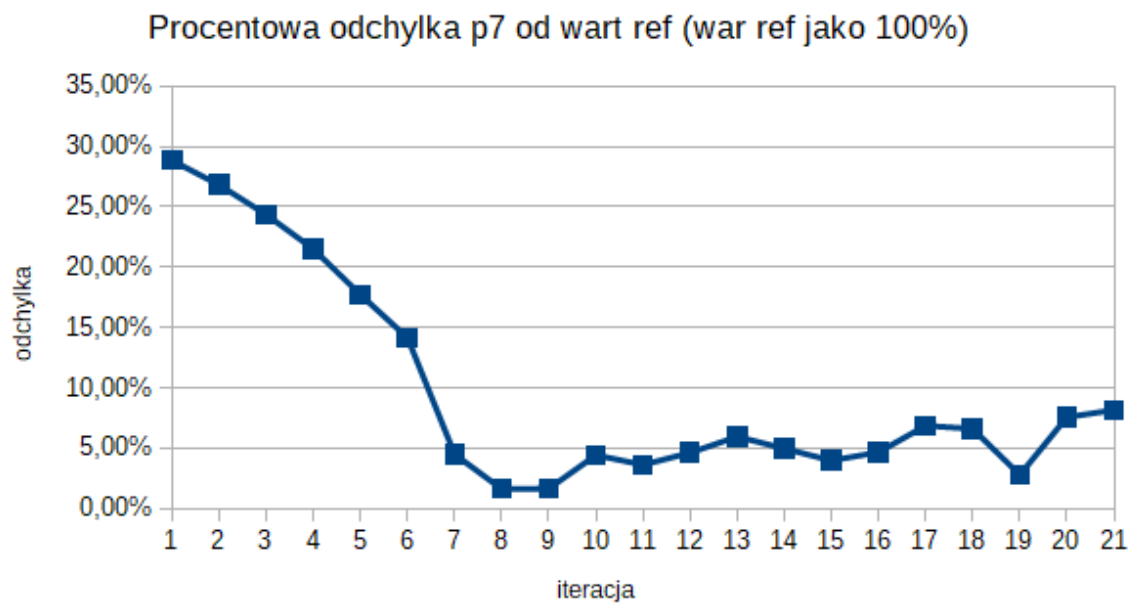


Figure 56: Procentowa Odchyłka średniej wartości p7 w populacji od wartości referencyjnej przy zaburzaniu p7

6.7.1.3 Rozkład TumorVolume

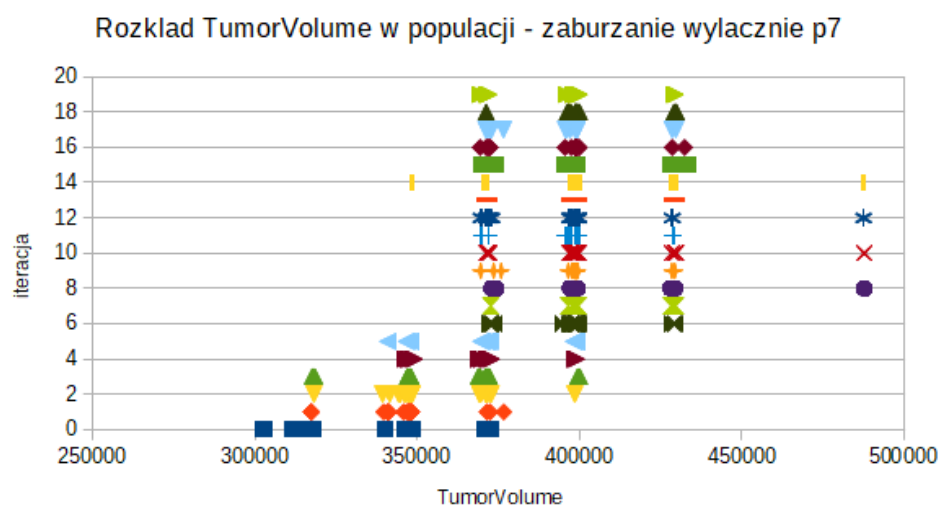


Figure 57: Rozkład *TumorVolume* w populacji przy zaburzaniu p7

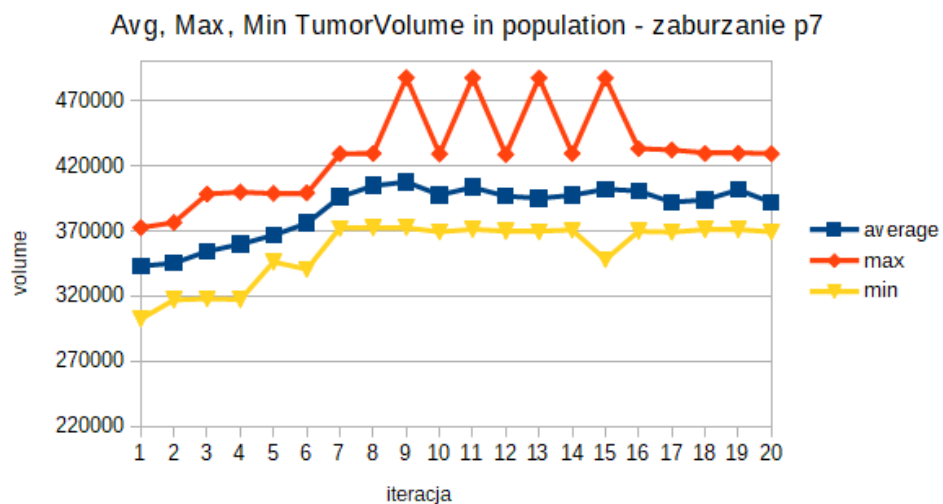


Figure 58: Wartości średnie, maksymalne oraz minimalne TumorVolume na kolejnych etapach działania algorytmu ewolucyjnego przy zaburzaniu p7

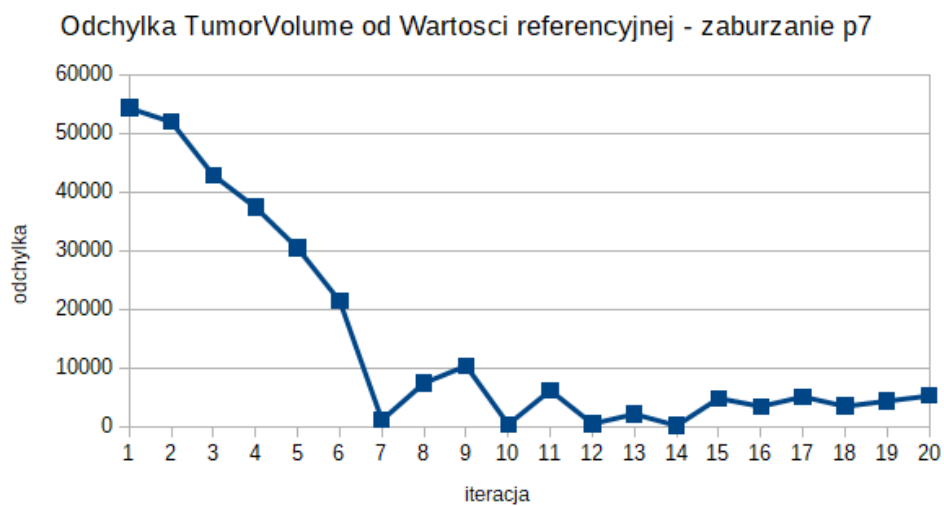


Figure 59: Odchyłka średniej wartości TumorVolume w populacji od wartości referencyjnej przy zaburzaniu p7

Procentowa odchyłka TumorVolume od jego wart ref (ref=100%) zaburzanie p7

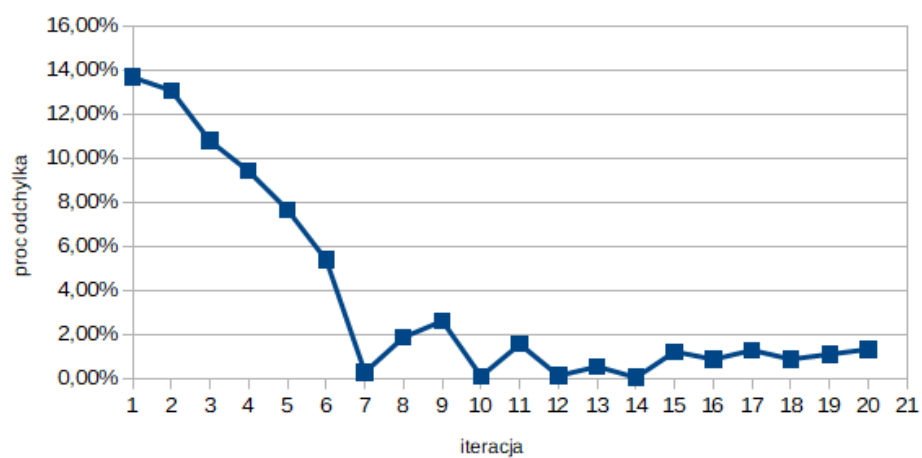


Figure 60: Procentowa Odchyłka sredniej wartosci TumorVolume w populacji od wartosci referencyjnej przy zaburzaniu p7