NONOGRAM:

Chromosom to dwieście dwadzieścia pięć zer lub jedynek dla nonogramu motyla. Gdzie jedynka oznacza pole zamalowane.

Funkcja fitness jest dosyć rozległa. Na początku przyjmuje chromosom (225 bitów) następnie tworzy z niego macierz 15x15. Każde 15 wierszy jest porównywana z zadanymi liczbami i analogicznie dla kolumn. Po sprawdzeniu czy ilość jedynek w wierszu lub kolumnie zgadza się z zadanymi liczbami, algorytm sprawdza czy pomiędzy odpowiednimi grupami jedynek występuje przerwa (zgodnie z zasadami rozwiazywania nonogramu).

Główna cześć funkcji fitness to:

fitnessFunc <- function(chr) {

macierz = matrix(

chr,

nrow=15,

ncol=15)

correctness = 0;

for (i in 1:15) {

if(sum(macierz[,i]) == sum(wartosciWierszy[i])){

correctness = correctness+0.25;

czyDodac = sprawdzCzyLiczbySieZgadzaja(macierz[,i],wartosciWierszy[,i])

correctness = correctness + czyDodac

}

}

for(i in 1:15) {

if(sum(macierz[i,]) == sum(wartosciKolumn[i])){

correctness = correctness+0.25;

czyDodac = sprawdzCzyLiczbySieZgadzaja(macierz[i,],wartosciKolumn[,i])

correctness = correctness + czyDodac

}

}

return(-correctness)

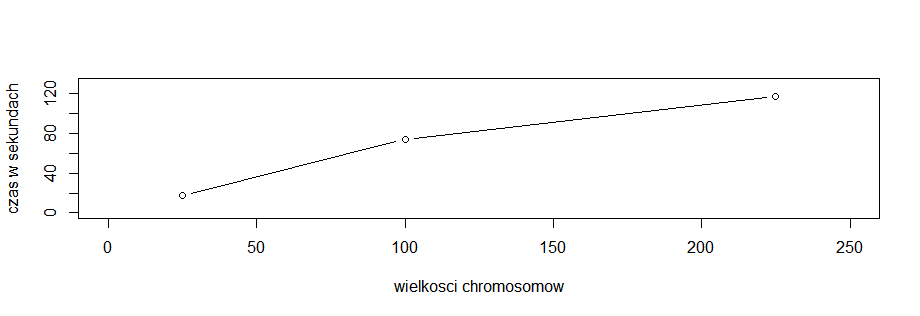
}

gdzie funkcja sprawdzCzyLiczbySieZgadzaja sprawdza czy występuje przerwa pomiędzy grupami jedynek.

Za każdą poprawność funkcja przyznaje 0.25 pkt. Suma jedynek się zgadza? +0.25. Istnieją przerwy pomiędzy grupami jedynek? +0.25. Jest to wykonywane dla kolumn i wierszy, co łącznie daje 1 pkt. Cały chromosom może uzyskać 15pkt(-15)

W pliku mamy trzy zadania. Motyla z zajęć oraz dwa inne nonogramy 5x5 i 10x10. Wystarczy podmienić wartosciKolumn na kolumny, wartoscWierszy na wiersze. Zmienić wielkość chromosomu na 100 i macierz na 10x10. to samo dla nonogramu 5x5.

Prędkość wykonania algorytmu działa w miarę liniowo:



Różnica pomiędzy nonogramem 5x5 a 15x15 to około 100 sekund. Najlepsze wyniki uzyskałem dla mutationChance 0.05. Zwiększanie populacji i iteracji zbliża nas do rozwiązania, ale większa wartość bardzo obciąża sprzęt. Ciężko stwierdzić jakie rozwiązanie będzie najefektywniejsze. Przy sprzęcie konsumenckim, rozsądnymi danymi będzie populacśja: 500 oraz iteracje 200. Przy takich ilościach nonogram jest rozwiązany w połowie.