

Sprawozdanie do zadania domowego

0. Opis projektu

0.0. Struktura chromosomu

Chromosom jest ciągiem binarnym o dł. bitów równej ilości zmiennych. Każdy bit odpowiada kolejnej zmiennej(0 lub 1).

0.1.Które chromosomy są dobre?

Chromosom jest tym lepszy im większa ilość klauzul w równaniu jest zdana(daje rezultat 1/TRUE).

0.2.Działanie funkcji fitness

Funkcja dostaje chromosom chr. Podstawia pod zmienne kolejne bity z chromosomu i oblicza klauzule. Im więcej klauzul jest spełnionych tym lepszą ocenę zwraca fitness. Dane z pliku CNF zostają wrzucone do dataframe. Potem funkcja fitness przechodzi po każdym elemencie i np. napotka zmienną nr "2" to podstawia chr[2] co jest albo true lub false. Jeśli true to zdane_klauzule +1 i przechodzi do następnego wiersza w dataframe. Jeden wiersz reprezentuje 1 klauzulę.

Przykład: fitness(chr<-(0,1,0))

p cnf 3 5

2 -3 1 0

2 -3 -1 0

-3 -1 2 0

2 -3 -1 0

-2 3 1 0

funkcja przechodzi po tabeli pomijając ostatnią kolumnę. Najpierw napotka 2 sprawdza czy chr[2] is true. Jeśli tak to dodaje +1 do zmiennej zdanych klauzul. Po przejściu całej tabelki zwraca zanegowaną zmienną zdane_klauzule jako ocena.

0.3.Maks. zakres funkcji fitness

max: wszystkie klauzule zdane

min: żadna klauzula niezdana

1.Eksperymenty z parametrami

Wykonano po 10 prób każdej komendy.

Legenda:

pop - populacja,

rep - powtórzenia/iteracje

mut - mutation rate

1.1. Testy parametru mutacji

20 zmiennych 91 klauzul

program(pop=300, rep=100, mut=0.05

sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')

```
[1] "Czas działania algorytmu genetycznego"
    user system elapsed
61.014  0.897  61.248
```

Maksymalna ocena już po około 31 iteracjach w 90% testów

program(pop=300, rep=31, mut=0.05,

sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')

```
[1] "Czas działania algorytmu genetycznego"
    user system elapsed
13.668  0.008  13.674
```

Maksymalna ocena osiągnięta w okolicach 30-35 powtórzenia

program(pop=300, rep=31, mut=0.01,

sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')

```
[1] "Czas działania algorytmu genetycznego"
    user system elapsed
14.249  0.008  14.256
```

Maksymalna ocena osiągnięta w okolicach 35-55 powtórzenia

program(pop=300, rep=50, mut=0.10,

sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')

```
[1] "Czas działania algorytmu genetycznego"
    user system elapsed
15.829  0.000  15.845
```

Maksymalna ocena osiągnięta w okolicach 45-50 powtórzenia

Wniosek: mutacja najefektywniejsza przy ustawieniu na 0.05

1.2. Testy parametru populacji

program(pop=200, rep=50, mut=0.05,

sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')

```
[1] "Czas działania algorytmu genetycznego"
user system elapsed
14.307 0.000 14.306
```

Maksymalna ocena osiągnięta w okolicach 25-30 powtórzeń, aczkolwiek 3/10 testów nie osiągnęły maksymalnego wyniku.

Wniosek: Wyższa populacja wcale nie oznacza szybszego osiągnięcia maksymalnej oceny.

```
program(pop=100, rep=70, mut=0.05,
sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')
```

```
[1] "Czas działania algorytmu genetycznego"
user system elapsed
9.966 0.000 9.965
```

Maksymalna ocena najczęściej osiągnięta w okolicach 50-55 powtórzeń, **aczkolwiek ze względu na małą populację 5/10 testów nie osiągnęły maksymalnego wyniku.**

```
program(pop=150, rep=70, mut=0.05,
sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')
```

```
[1] "Czas działania algorytmu genetycznego"
user system elapsed
14.845 0.000 14.844
```

Maksymalna ocena najczęściej osiągnięta w okolicach 30-40 powtórzeń

Eksperymenty wykazują, iż najbardziej optymalne rozwiązanie to pop=300, rep=31, mut=0.05, które wykonuje się mniej więcej w 13,5 sekundy. Nie jest to rozwiązanie najszybsze, aczkolwiek dostajemy maksymalną ocenę w 90% przypadków.

Najszybszym rozwiązaniem okazało się pop=100, rep=70, mut=0.05. Niestety ze względu na małą populację zdarzało się osiągać maksymalnie wynik 88/91 oraz 87/91 oraz w 50% eksperymentów nie osiągnęliśmy maksymalnego wyniku wcale.

2. Porównanie czasu działania z algorytmem DPLL

2.1. (20 zmiennych 91 klauzul)

Algorytm DPLL

```
time ./solver < uf20-027.cnf
```

SAT

```
-1 -2 3 -4 -5 6 7 -8 9 10 11 12 -13 -14 15 -16 -17 -18 19 -20 0
```

```
real 0m0.007s
```

```
user 0m0.003s
```

sys 0m0.004s

Algorytm genetyczny

```
program(pop=300, rep=31, mut=0.05,  
sciezka='/home/mati/IO/Zad_dom_1/uf20-027.cnf')
```

```
[1] "Czas działania algorytmu genetycznego"  
user system elapsed  
13.668 0.008 13.674
```

Wniosek: Przy małych problemach algorytm DPLL znajduje rozwiązanie problemu prawie 200x szybciej.

2.2. (200 zmiennych 860 klauzul)

Algorytm DPLL

```
time ./solver < uf200-014.cnf
```

SAT

```
-1 2 3 -4 5 6 7 -8 -9 10 11 12 -13 -14 15 16 17 -18 -19 -20 -21 22 23 -24 25 26 -27  
28 -29 30 31 -32 33 34 35 36 -37 38 -39 40 41 -42 43 44 45 46 -47 -48 49 50 -51 52  
-53 -54 55 -56 57 -58 -59 60 61 62 63 64 -65 66 -67 68 -69 -70 71 -72 -73 74 75 -76  
77 78 79 80 81 82 -83 84 85 -86 87 88 89 90 91 -92 93 94 -95 96 -97 98 99 100 -101  
-102 -103 104 -105 -106 107 108 -109 110 -111 -112 -113 -114 -115 -116 -117 -118  
119 -120 121 -122 -123 -124 -125 126 -127 128 129 -130 -131 -132 -133 -134 135  
-136 -137 138 -139 140 -141 142 143 144 -145 -146 -147 148 -149 150 151 152 153  
154 155 156 -157 -158 159 -160 161 -162 -163 -164 -165 166 -167 -168 -169 170  
171 -172 -173 -174 175 -176 177 -178 179 180 181 -182 -183 -184 185 -186 187  
188 -189 190 -191 -192 -193 -194 -195 196 197 198 199 -200 0
```

real 0m**20.191s**

user 0m20.126s

sys 0m0.064s

Algorytm DPLL

```
program(pop=200, rep=50, mut=0.05,  
sciezka='/home/mati/IO/Zad_dom_1/problems/uf200-014.cnf')
```

```
[1] "Czas działania algorytmu genetycznego"  
user system elapsed  
167.216 0.212 166.827
```

Wniosek: Przy dużych problemach algorytm DPLL znajduje rozwiązanie problemu prawie 8,5x szybciej.

3. Czas działania alg. gen. przy różnych ilościach zmiennych

3 zmienne 5 klauzul

```
program(pop=200, rep=50, mut=0.05,  
sciezka='/home/mati/IO/Zad_dom_1/problems/3_5_1.dimacs')
```

```
[1] "Czas działania algorytmu genetycznego"  
user system elapsed  
0.961 0.000 0.961
```

10 zmiennych 30 klauzul

```
program(pop=200, rep=50, mut=0.05,  
sciezka='/home/mati/IO/Zad_dom_1/problems/10_30_1.dimacs')
```

```
[1] "Czas działania algorytmu genetycznego"  
user system elapsed  
4.47 0.00 4.47
```

60 zmiennych 100 klauzul

```
program(pop=200, rep=50, mut=0.05,  
sciezka='/home/mati/IO/Zad_dom_1/problems/3_5_1.dimacs')
```

```
[1] "Czas działania algorytmu genetycznego"  
user system elapsed  
19.075 0.000 19.070
```

Wniosek: Czas wzrasta liniowo.