Machine Coded Genetic Algorithms for Real-Valued Optimization Problems

Raport z eksperymentu

Wstęp

Testy algorytmu mcga zostały przeprowadzone zgodnie z konwencją benchmarka CEC2017, który testuje algorytm w 30 różnych funkcjach.

Ograniczenia, które przyjęliśmy to:

- Dopuszczalne granice dla każdego wymiaru to *[-100.0, 100.0]*
- Liczba wywołań funkcji dopasowania *MaxFES = 10000*D*, gdzie *D* to liczba wymiarów, dla każdej funkcji w benchmarku przyjęliśmy *maxiter = MaxFES/popsize*

Zaś konwencja zapisywania wyników została przyjęta zgodnie z wytycznymi prowadzącego, czyli:

```
kod_funkcji_f;licznik_wywołań_f;raportowana_wartość_f
f11;1;0.5
f11;2;0.45
[...]
```

Gdzie *f* to funkcja minimalizowana.

Niestety ze względu na duże rozbieżności co do wyników w funkcji nr 2, postanowiliśmy nie uwzględniać jej w dokumentacji.

Niestety w dokumentacji algorytmu nie jest nigdzie wykazane z jakich metod uwzględniania ograniczeń kostkowych korzysta funkcja mcga (wiadomo nam jedynie, iż korzysta z pakietu GA do generowania nowych osobników), zaś modyfikowanie funkcji celu w celu uwzględnienia ograniczeń uznaliśmy za niewskazane, gdyż byłaby to ingerencja w pakiet benchmarku CEC2017. Ustaliliśmy więc jedynie parametry *minval* i *maxval*, na granice wskazane w dokumentacji CEC2017.

Parametry

Parametry przekazane do algorytmu zostały określone empirycznie na podstawie testów oraz teoretycznie na podstawie naszej wiedzy o algorytmach heurystycznych:

- *popsize = 200*, jako że wielkość populacji wpływa na maksymalną liczbę pokoleń, duża populacja powoduje małą liczbę pokoleń, a tym samym mniejszy wpływ selekcji na wyniki algorytmu, powoduje także, że dla dużej liczby wymiarów, obliczenie funkcji dopasowania dla każdego osobnika znacząco się wydłuża, dlatego zależało nam, aby liczba osobników nie była zbyt duża, metodą empiryczną określiliśmy, że 200 to wystarczająco odpowiednia liczba
- *crossprob = 0.8,* prawdopodobieństwo krzyżowania jest głównym ‘motorem’ napędowym algorytmu mcga, dlatego zbyt mała wartość hamuje algorytm, zbyt duża zaś wartość wydłuża czas obliczeń, po testach, wartość 0.8 uznaliśmy za wystarczająco odpowiednią
- *mutateprob = 0.01,* prawdopodobieństwo mutacji nie jest główną cechą tego algorytmu, a jako że wiedzieliśmy, że zbyt duża wartość często powodowałaby powstawanie niepożądanych osobników, zostawiliśmy ją na poziomie domyślnym, czyli 1%, co powoduje, że nadal istnieje opcja ‘szczęśliwego strzału’, ale nie zaburza zbytnio działania algorytmu

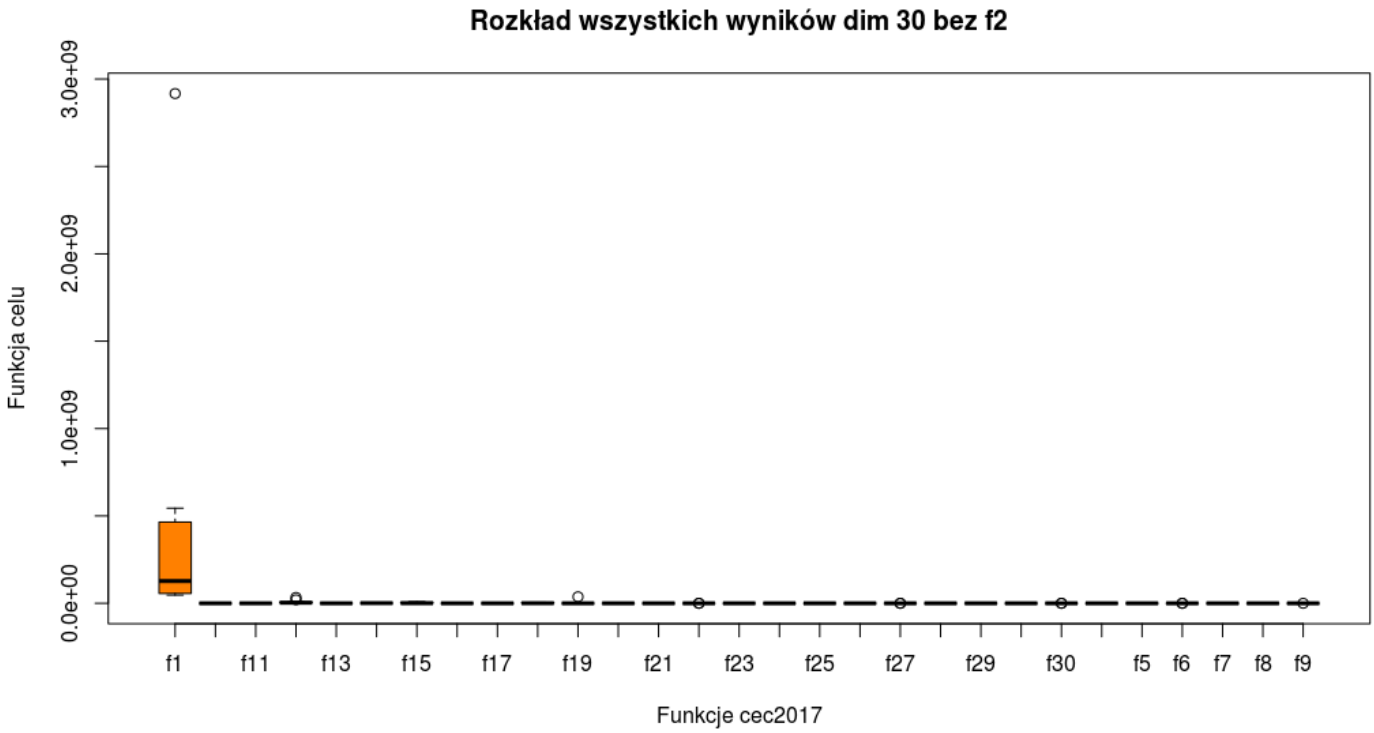
- *elitism = 1,* liczbę osobników elitarnych również pozostawiliśmy na poziomie domyślnym, czyli 1 osobnik na populację, ze względu na niezbyt dużą liczbę osobników w populacji

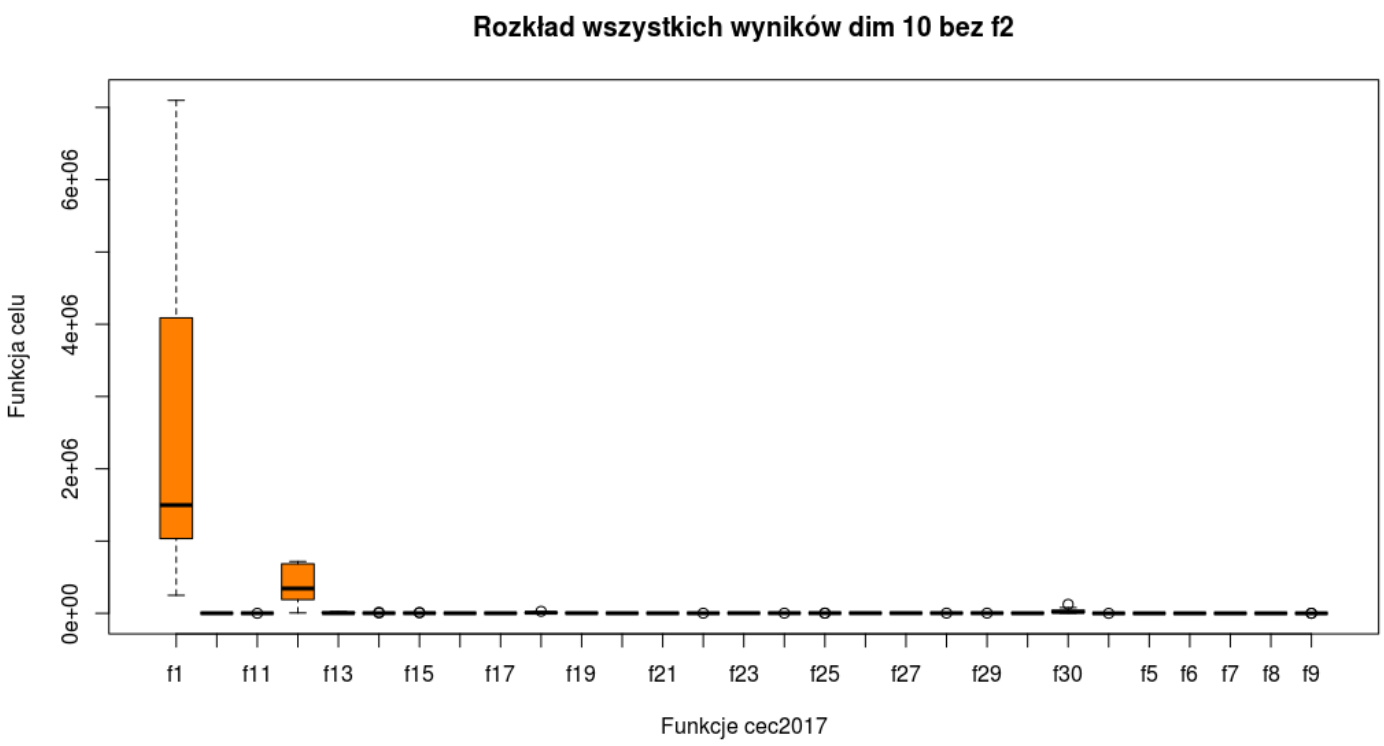
Pozostałe parametry określane były zgodnie z konwencją benchmarka CEC2017:

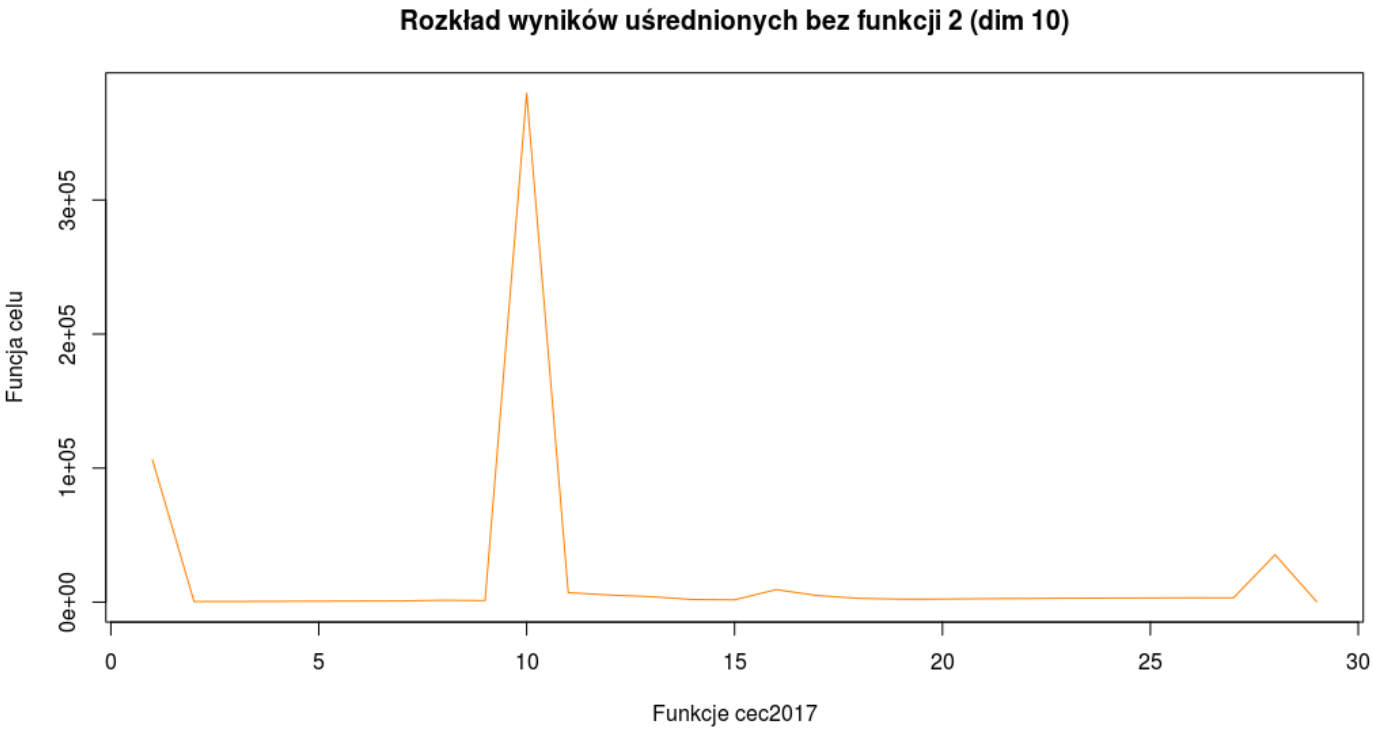
- *minval = -100.0,* minimalna wartość dla każdego z wektorów bajtowych zamienionych na liczbę rzeczywistą dla populacji początkowej
- *maxval = 100.0,* maksymalna wartość dla każdego z wektorów bajtowych zamienionych na liczbę rzeczywistą dla populacji początkowej
- *chsize = 10/30,* liczba wymiarów użyta do eksperymentu, jedne wyniki dla 10 wymiarów i drugie dla 30 wymiarów
- *maxiter = (10000*chsize)/popsize,* maksymalna liczba pokoleń, obliczona na podstawie podanej w dokumentacji wartości MaxFES, czyli 10000*D, gdzie D – liczba wymiarów
- *evalFunc,* jedna z 30 funkcji benchmarka CEC2017, która wywoływana jest kolejno 10 razy, aby zmniejszyć czynnik losowy

Wyniki

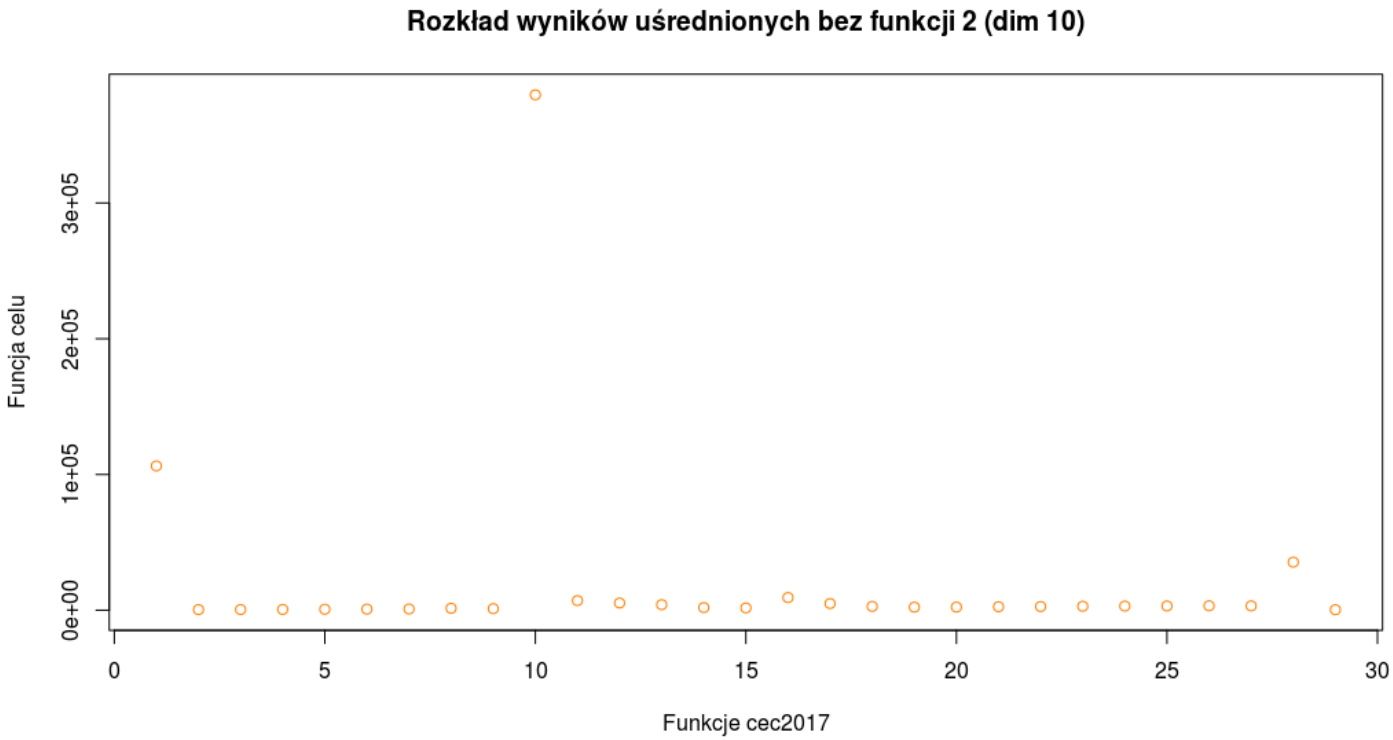
Poniżej prezentujemy wyniki eksperymentu w postaci graficznej, odpowiednio opisane, bez funkcji f2:

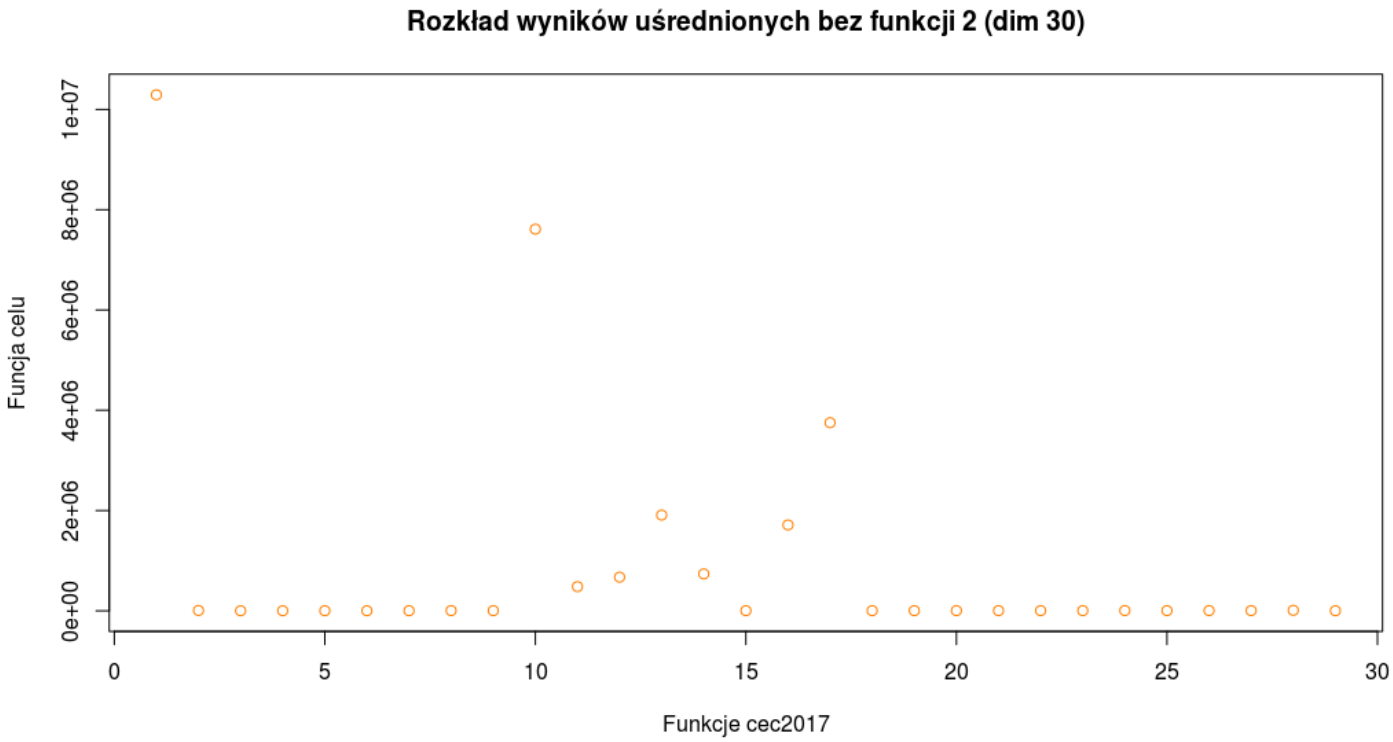

Rozkład wszystkich wyników dim 30 bez f2
Funkcja celu
0.0e+00
1.0e+09
2.0e+09
3.0e+09
f1
f11
f13
f15
f17
f19
f21
f23
f25
f27
f29
f30
f5
f6
f7
f8
f9
Funkcje cec2017


Rozkład wszystkich wyników dim 10 bez f2
Funkcja celu
0e+00
2e+06
4e+06
6e+06
f1
f11
f13
f15
f17
f19
f21
f23
f25
f27
f29
f30
f5
f6
f7
f8
f9
Funkcje cec2017


Rozkład wyników uśrednionych bez funkcji 2 (dim 10)
Funcja celu
0e+00
1e+05
2e+05
3e+05
0
5
10
15
20
25
30
Funkcje cec2017


Rozkład wyników uśrednionych bez funkcji 2 (dim 30)
Funcja celu
0e+00
2e+06
4e+06
6e+06
8e+06
1e+07
0
5
10
15
20
25
30
Funkcje cec2017


Rozkład wyników uśrednionych bez funkcji 2 (dim 10)
Funcja celu
0e+00
1e+05
2e+05
3e+05
0
5
10
15
20
25
30
Funkcje cec2017


Rozkład wyników uśrednionych bez funkcji 2 (dim 30)
Funcja celu
0e+00
2e+06
4e+06
6e+06
8e+06
1e+07
0
5
10
15
20
25
30
Funkcje cec2017