

Na prvem primeru lahko vidimo rezultate pri paralelizaciji s pomočjo knjižnice OpenMP.

MAX_NFES	Time in milliseconds	Best value
<i>1 thread</i>		
10 000	1153	4.22762
100 000	11135	4.34501
1 000 000	111546	3.71319
<i>2 threads</i>		
10 000	664	< 0
100 000	6169	< 0
1 000 000	61058	< 0
<i>4 threads</i>		
10 000	389	< 0
100 000	3832	< 0
1 000 000	36588	< 0
<i>8 threads</i>		
10 000	308	< 0
100 000	2899	< 0
1 000 000	27630	< 0

Po paralelizaciji s pomočjo C++ knjižnice std::thread pa rezultati izgledajo tako kot kaže spodnja tabela.

MAX_NFES	Time in milliseconds	Best value
<i>1 thread</i>		
10 000	1493	4.75542
100 000	13450	4.74588
1 000 000	115296	4.30094
<i>2 threads</i>		
10 000	700	< 0
100 000	7098	< 0
1 000 000	72258	< 0
<i>4 threads</i>		
10 000	601	< 0
100 000	6176	< 0
1 000 000	62821	< 0

Pri izvajanju programa z OpenMP paralelizacijo sem uporabljal procesor Intel Core i7 3770, ki omogoča sočasno delovanje 8 niti.

Pri izvajanju programa z std::thread pa sem uporabljal procesor Intel Core i5 Dual-Core, ki pa omogoča sočasno delovanje 4 niti, zato pri drugi tabeli manjka zadnji odstavek.

Po krajši analizi lahko vidimo, da OpenMP knjižnica deluje malenkost hitreje v skoraj vseh primerih, je pa res, da testni pogoji niso bili najboljši, saj je testiranje potekalo na dveh različnih strojih.

Izvajanje je pri obeh potekalo na način, da CEC klika ni v kritični sekciji, zato vrednosti pri paralelizaciji niso pravilne, saj so < 0.

MAX_NFES	Time in milliseconds	Best value
10 000	1950	4.88952
100 000	16780	4.65056
1 000 000	174600	4.3094