

Sprawozdanie		
Projekt Algorytmów Genetycznych		
Przygotowali:	Piotr Pietruszka 171842	Szymon Broda 171652
Kierunek: AiR	Specjalizacja: KSD	Grupa: 1

## 1 Wprowadzenie

W projekcie zostało zrealizowane zadanie optymalizacji problemu liniowo-kwadratowego (Zad 1.e), gdzie celem jest minimalizacja wskaźnika  $J$  obliczanego z równania 1.

$$J = x_N^2 + \sum_{k=0}^{N-1} (x_k^2 + u_k^2) \quad (1)$$

Wektor stanu  $\mathbf{x}$  jest wyznaczany z równania 2.

$$x_{k+1} = x_k + u_k \quad (2)$$

gdzie  $k = 0, 1, 2, \dots, N - 1$ , a  $u_k \in [-200, 200]$ .

W celu znalezienia optymalnego rozwiązania zaimplementowano strategię ewolucyjną  $\mu + \lambda$ . Parametrem podlegającym optymalizacji jest wektor pobudzeń  $\mathbf{u}$ . Optymalizacji dokonano dla różnych długości ( $N$ ) tego wektora, która przyjmowała wartości, tak jak w Zad 1.c ( $N = 5, 10, 15, 20, 25, 40, 35, 40, 45$ ). By zadanie nie było trywialne, początkowy stan  $x_0$  musi być różny od 0.

## 2 Program

Napisany program składa się z 2 plików:

- AG.py - przebieg algorytmu i zapisanie wyników do plików
- results.py - odczytanie danych z plików i ich wyświetlenie / zapisanie w formie graficznej

W pliku AG.py zdefiniowano 2 klasy:

- Individual - odpowiada ona pojedynczemu osobnikowi. Zawiera optymalizowany wektor pobudzeń, odpowiadający mu wektor odchyleń standardowych oraz metody liczące wskaźnik  $J$ , stan w czasie i dokonujące mutacji.
- GA - zawiera listę osobników, parametry strategii genetycznej oraz metody odpowiedzialne za przebieg optymalizacji - dokonujące mutacji, krzyżowania, znajdujące najlepsze osobniki itp.

Program zrealizowano zgodnie z opisem strategii  $\mu + \lambda$  znajdującym się w wykładzie. Początkowa populacja (o rozmiarze  $\mu$ ) tworzona jest poprzez wylosowanie z dostępnego przedziału, z rozkładem równomiernym, wartości wektora pobudzeń. Wektory pobudzeń standardowych są losowane, również z rozkładem równomiernym, z przedziału  $[0, 5]$ . W każdej iteracji algorytmu wybieranych jest losowo  $\lambda$  indeksów osobników, który zostają rodzicami, a następnie następuje krzyżowanie arytmetyczne. W dalszej kolejności dokonywana jest mutacja, a na końcu z populacji  $\mu$  rodziców i  $\lambda$  potomków wybieranych jest  $\mu$  najlepszych (o najmniejszym indeksie  $J$ ) osobników, które tworzą nową populację. Kroki te są powtarzane, aż do momentu osiągnięcia zakładanej liczby iteracji.

### 3 Wyniki

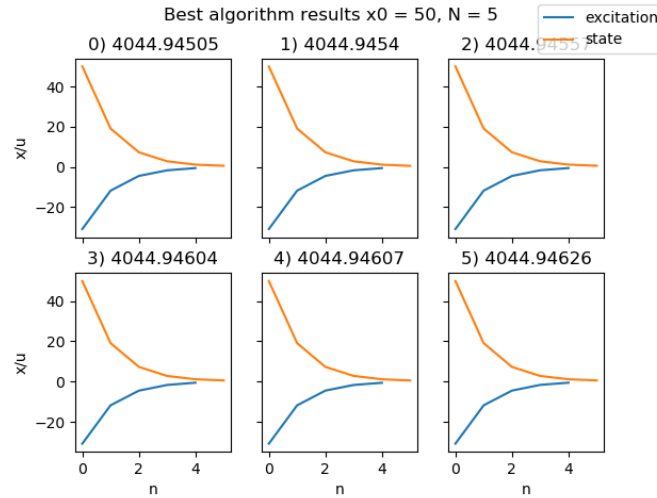
Optymalizację przeprowadzono dla różnych wartości  $N$  ( $N = 5, 10, 15, 20, 25, 40, 35, 40, 45$ ). Dla każdego z tych przypadków wyświetlane są:

- 6 najlepszych, końcowych wyników - przedstawionych jako wykresy pobudzenia i stanu, w zależności od czasu (np. rysunek 1).
- średnia i najlepsza wartość wskaźnika  $J$ , w zależności od iteracji algorytmu (np. rysunek 3).
- średnia i najlepsza wartość wskaźnika  $J$ , w zależności od iteracji algorytmu, począwszy od określonej iteracji. Wartości wskaźnika maleją na ogół drastycznie w początkowej fazie działania algorytmu, przez co nie widać późniejszych zmian. By je zobrazować zostały zamieszczone wykresy, zaczynające się od ok. 50 iteracji (np. rysunek 2).
- średnie odchylenie standardowe (dla wszystkich osobników i elementów w wektorach odchylen), w zależności od iteracji algorytmu (np. rysunek 4).

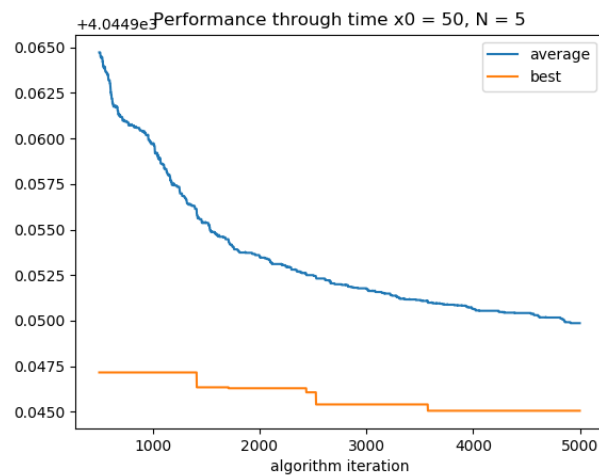
Optymalizację przeprowadzono dla następujących parametrów:

- liczba iteracji: 5000 (ostatni przykład 500)
- $\mu$ : 100
- $\lambda$ : 700
- $x_0$ : 50 (ostatni przykład -800)

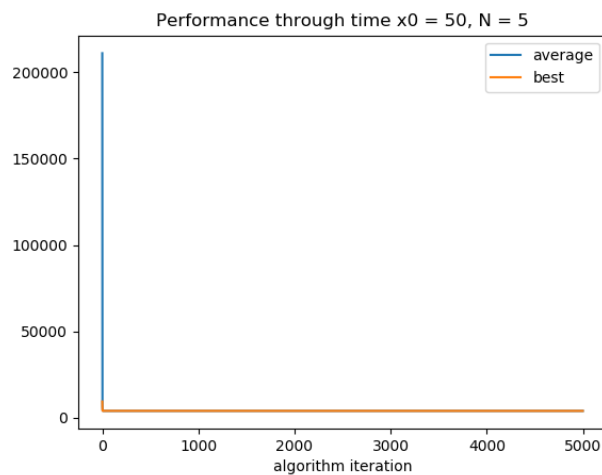
#### 3.1 N=5



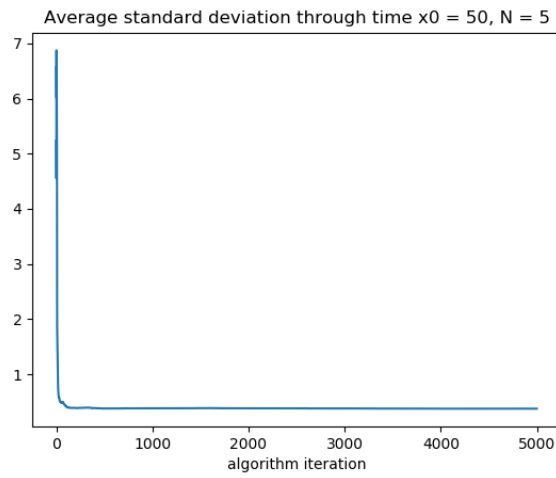
Rysunek 1: Najlepsze wyniki po zakończeniu działania,  $N=5$ .



Rysunek 2: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=5$ .

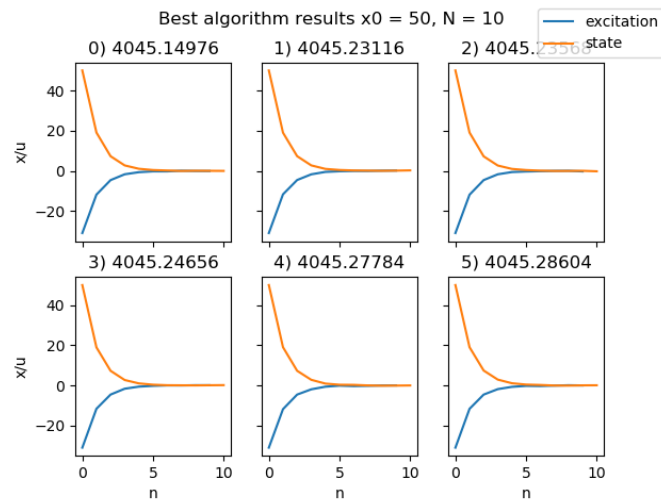


Rysunek 3: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=5$ .

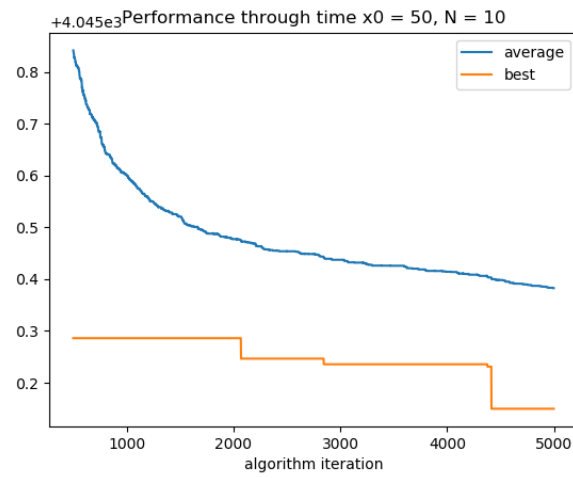


Rysunek 4: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=5$ .

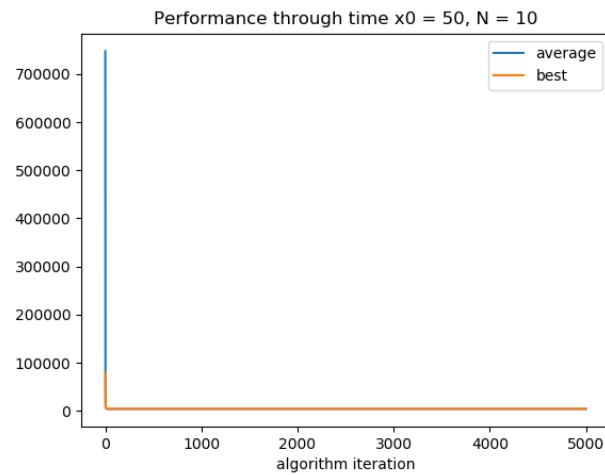
### 3.2 $N=10$



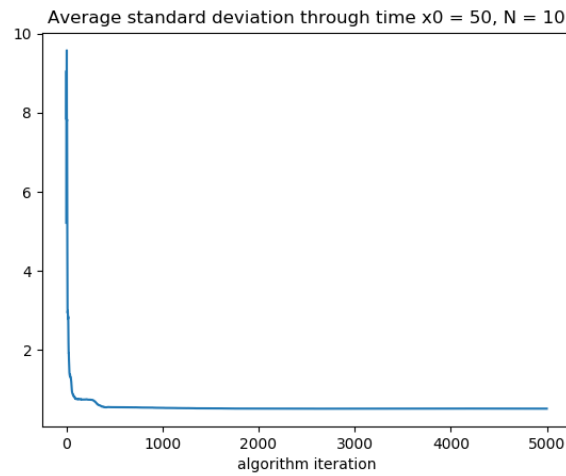
Rysunek 5: Najlepsze wyniki po zakończeniu działania,  $N=10$ .



Rysunek 6: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=10$ .

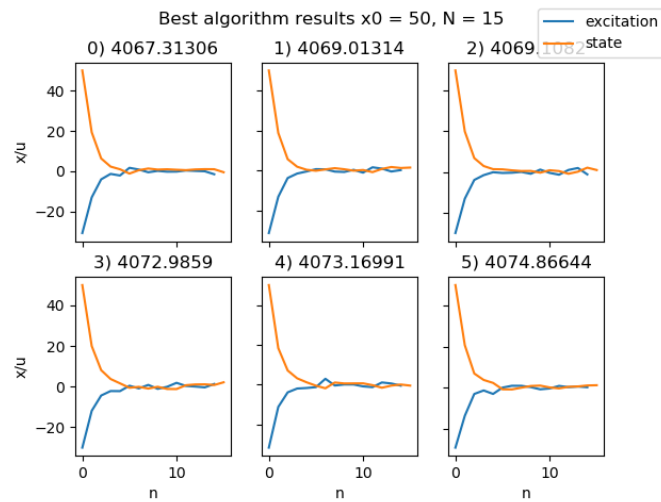


Rysunek 7: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=10$ .

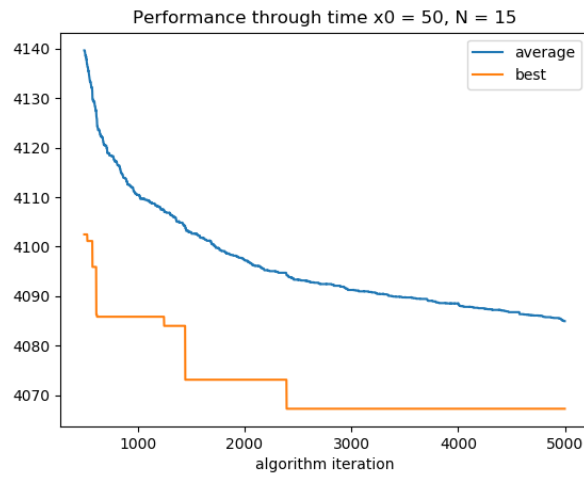


Rysunek 8: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=10$ .

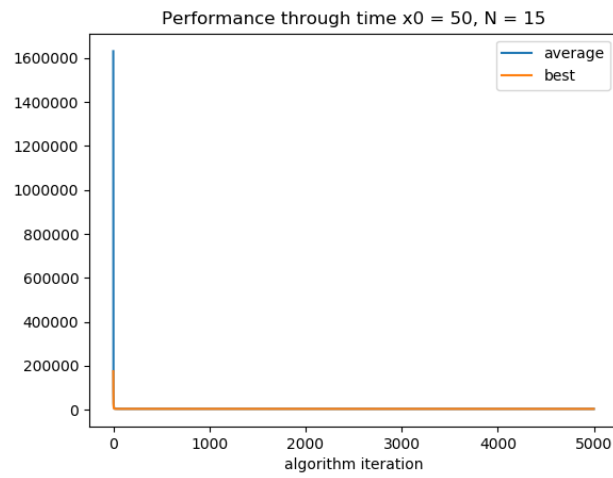
### 3.3 $N=15$



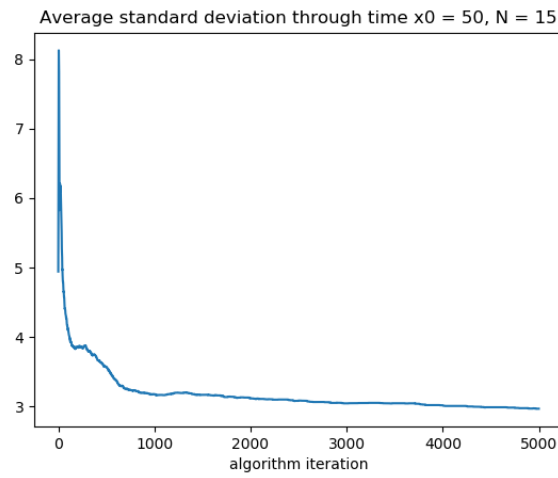
Rysunek 9: Najlepsze wyniki po zakończeniu działania,  $N=15$ .



Rysunek 10: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=15$ .

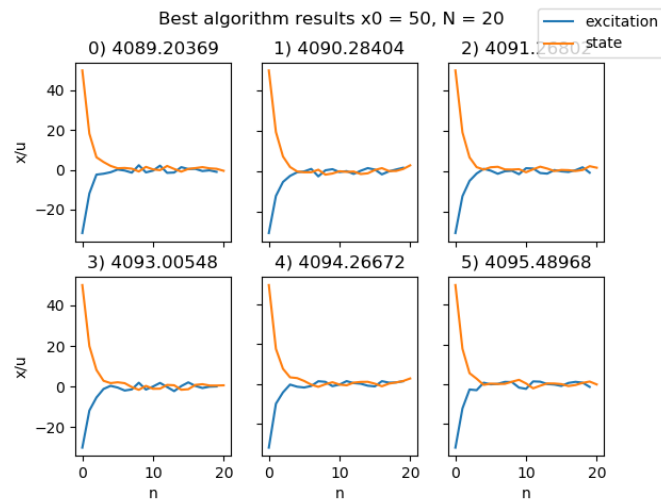


Rysunek 11: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=15$ .



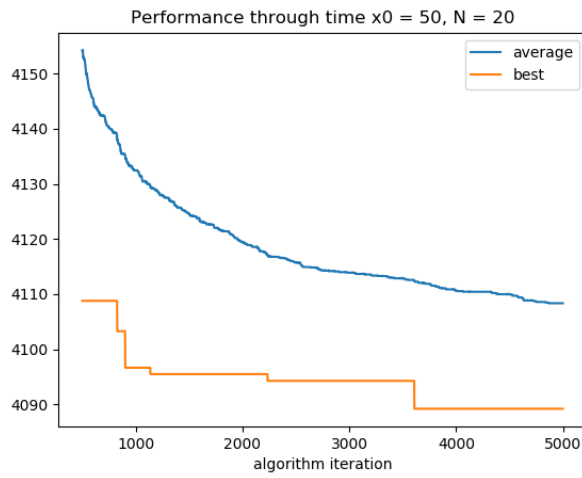
Rysunek 12: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=15$ .

### 3.4 $N=20$

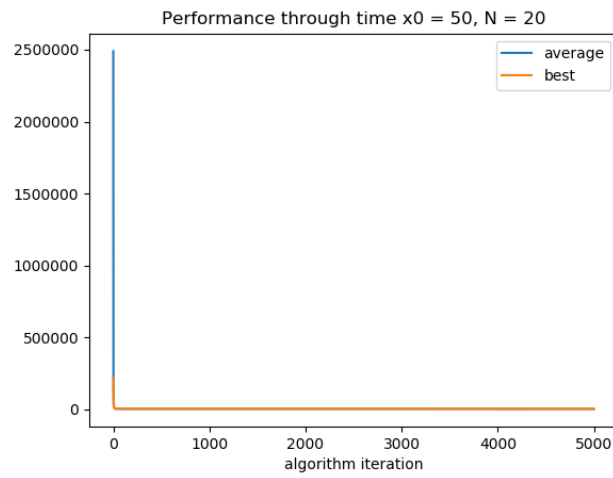


Rysunek 13: Najlepsze wyniki po zakończeniu działania,  $N=20$ .

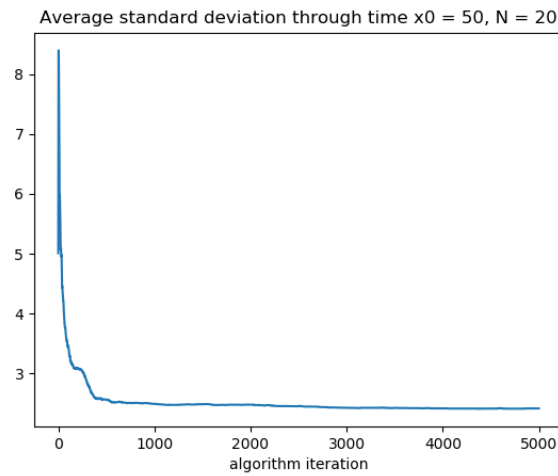




Rysunek 14: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=20$ .

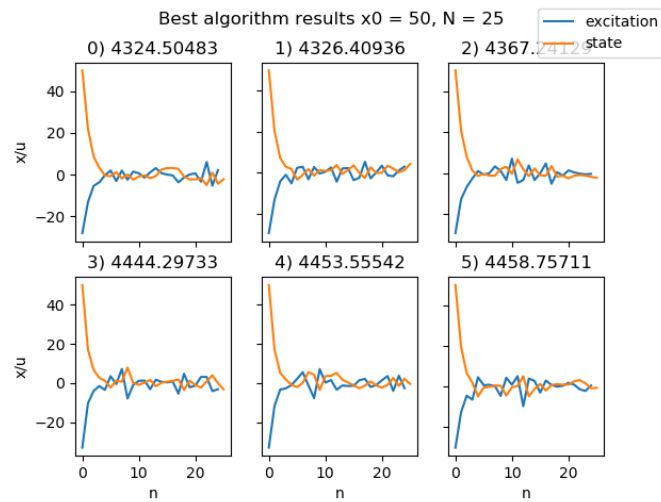


Rysunek 15: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=20$ .

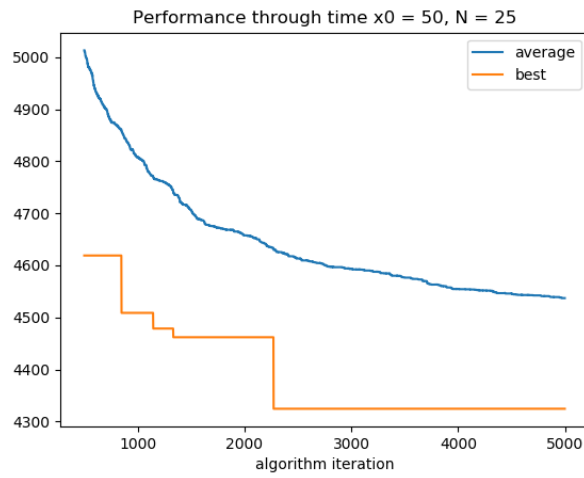


Rysunek 16: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=20$ .

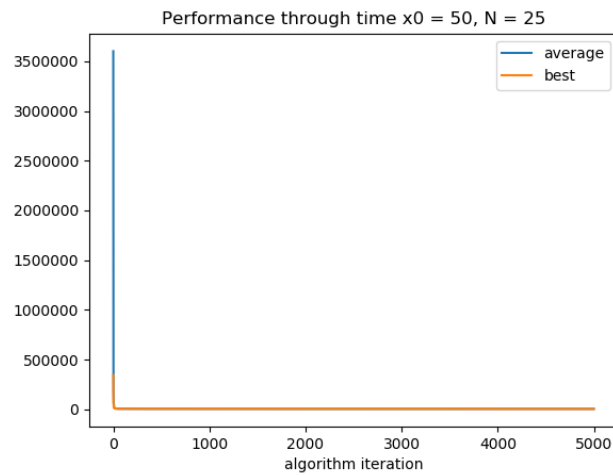
### 3.5 $N=25$



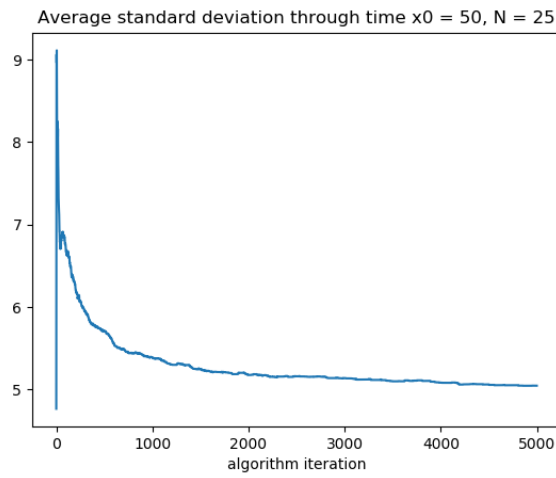
Rysunek 17: Najlepsze wyniki po zakończeniu działania,  $N=25$ .



Rysunek 18: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=25$ .

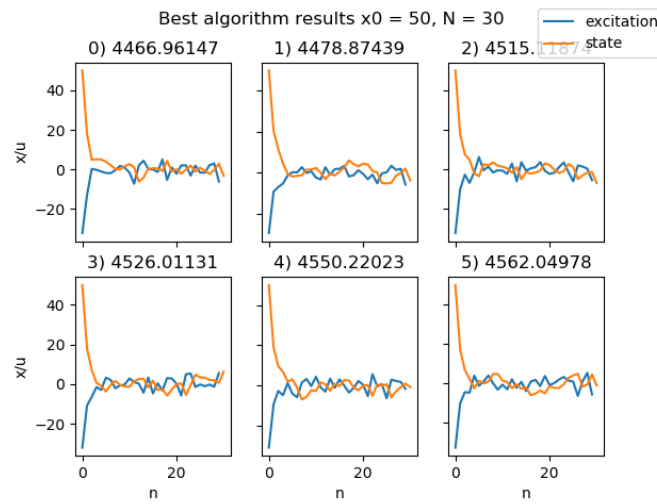


Rysunek 19: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=25$ .

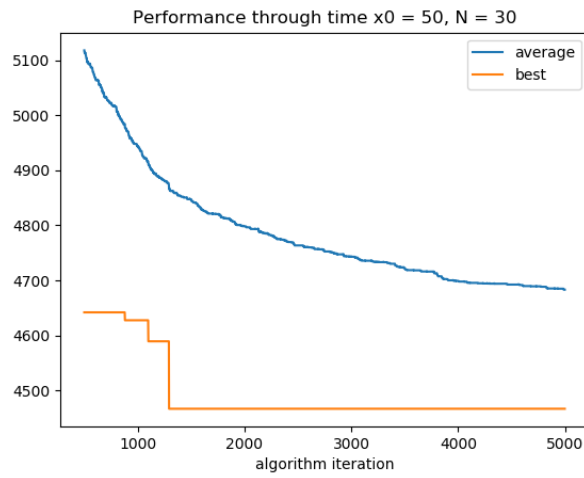


Rysunek 20: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=25$ .

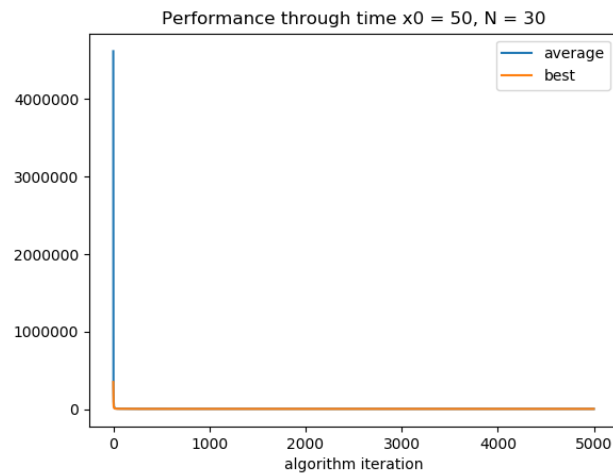
### 3.6 $N=30$



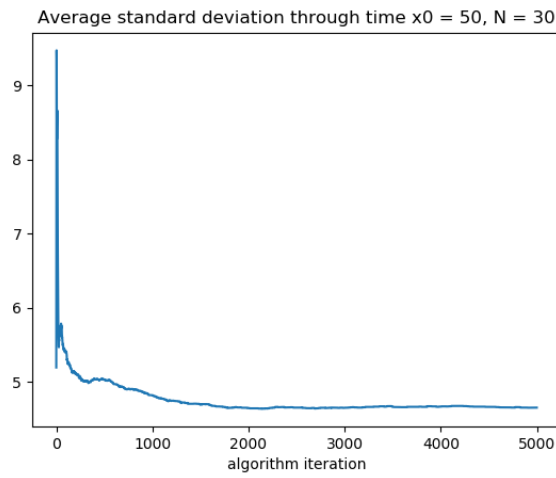
Rysunek 21: Najlepsze wyniki po zakończeniu działania,  $N=30$ .



Rysunek 22: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=30$ .

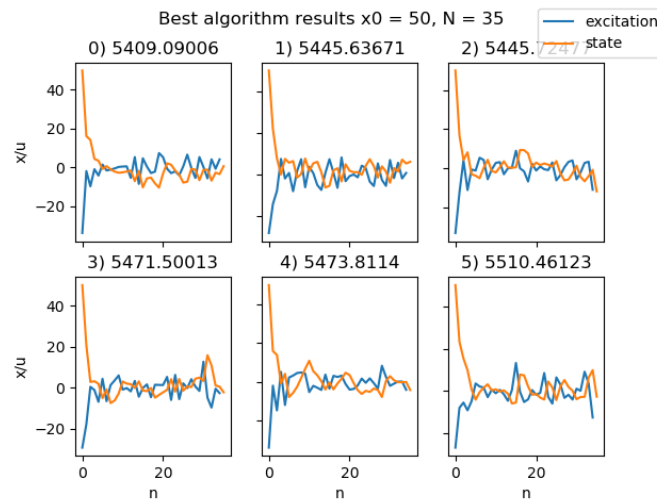


Rysunek 23: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=30$ .

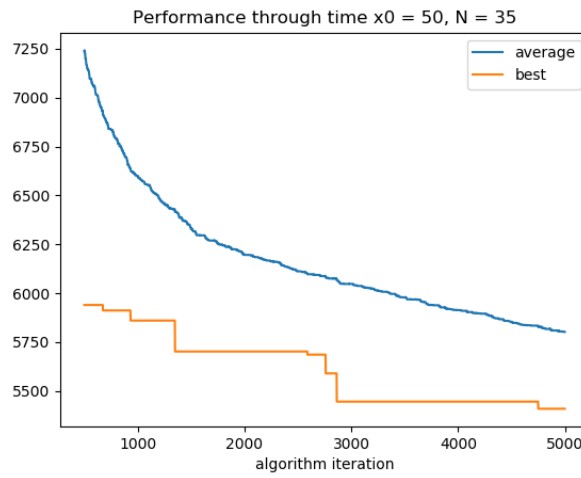


Rysunek 24: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=30$ .

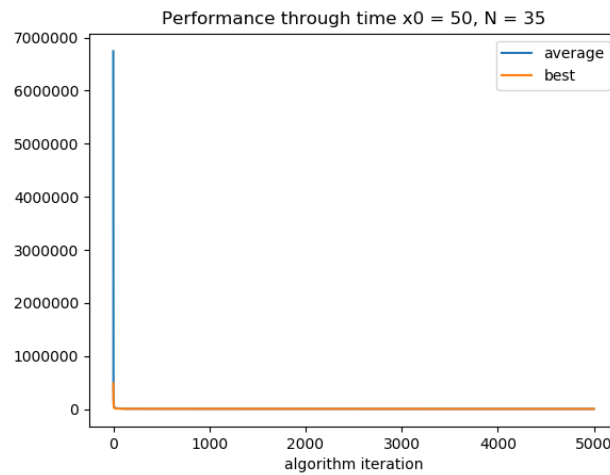
### 3.7 $N=35$



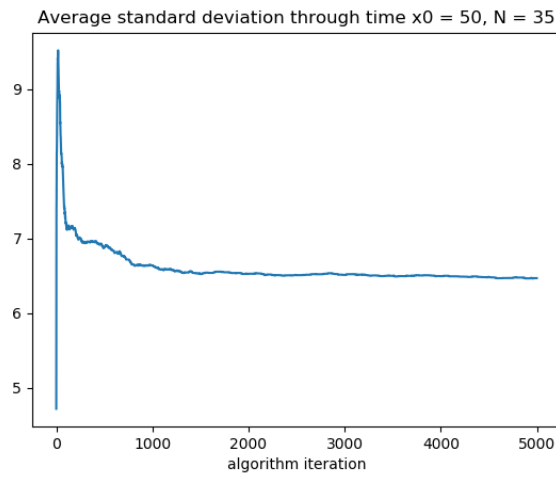
Rysunek 25: Najlepsze wyniki po zakończeniu działania,  $N=35$ .



Rysunek 26: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=35$ .

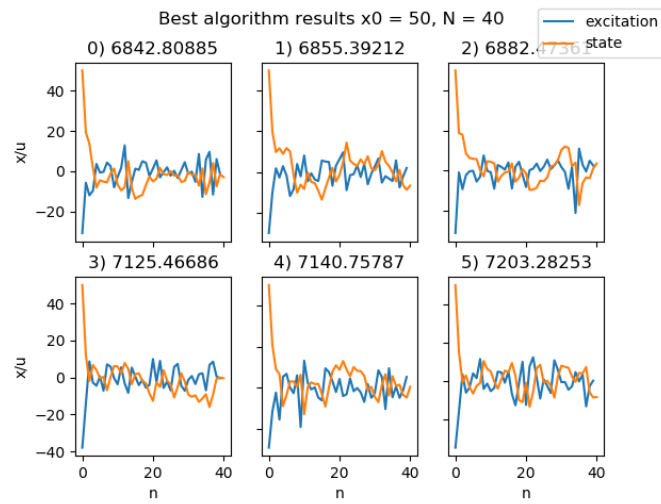


Rysunek 27: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=35$ .



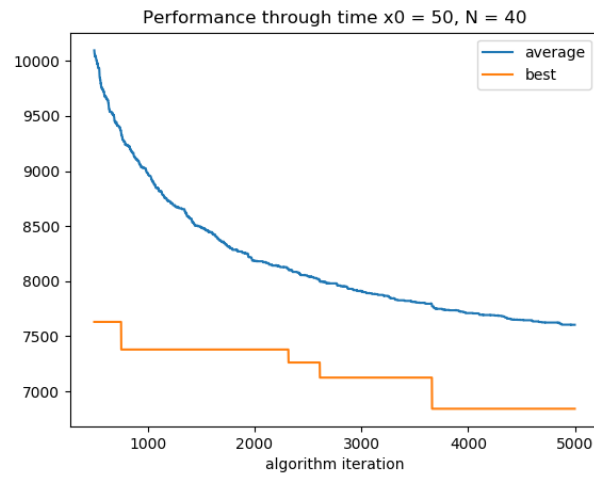
Rysunek 28: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=35$ .

### 3.8 $N=40$

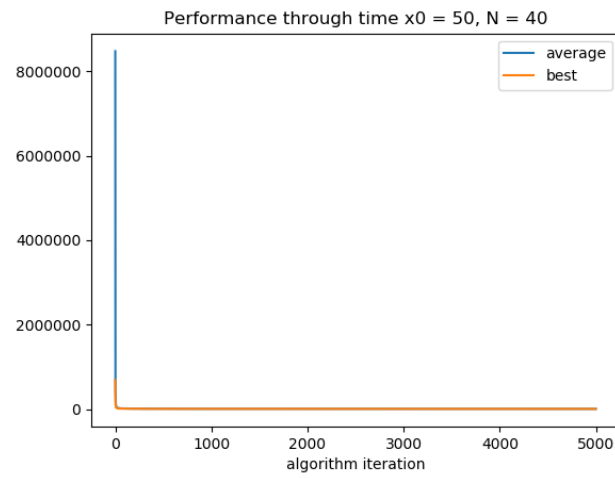


Rysunek 29: Najlepsze wyniki po zakończeniu działania,  $N=40$ .

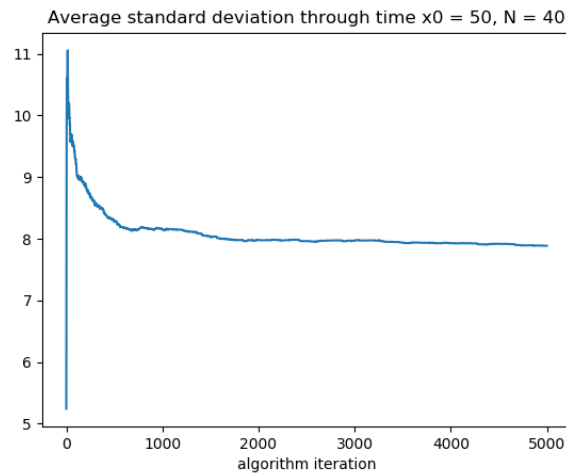




Rysunek 30: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=40$ .

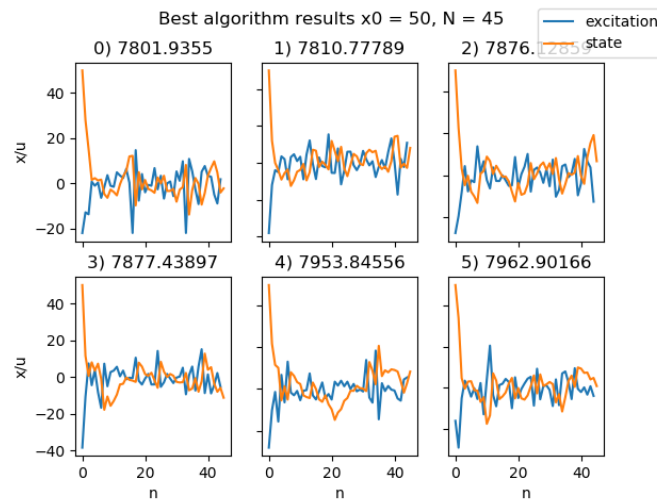


Rysunek 31: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=40$ .

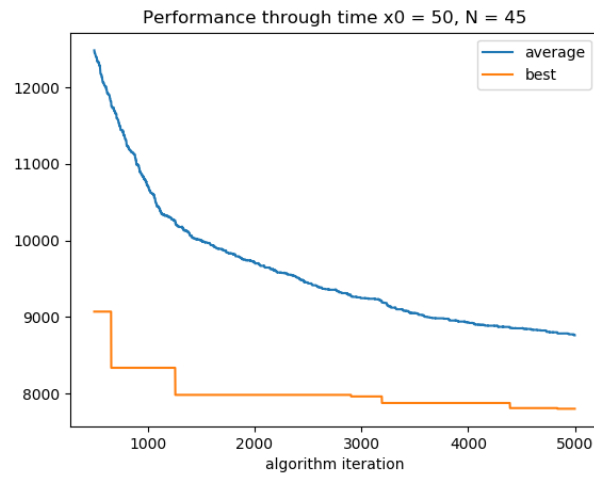


Rysunek 32: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=40$ .

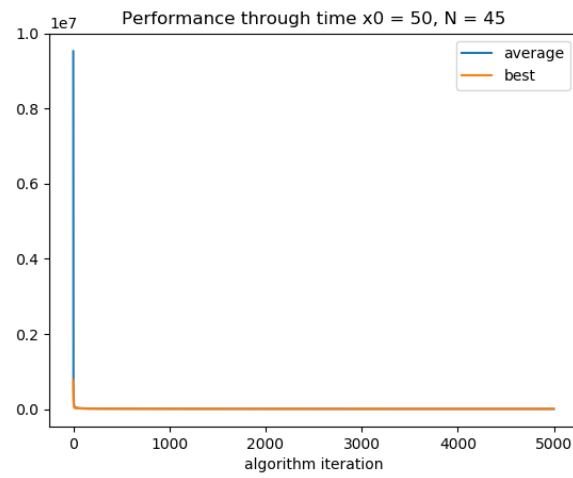
### 3.9 $N=45$



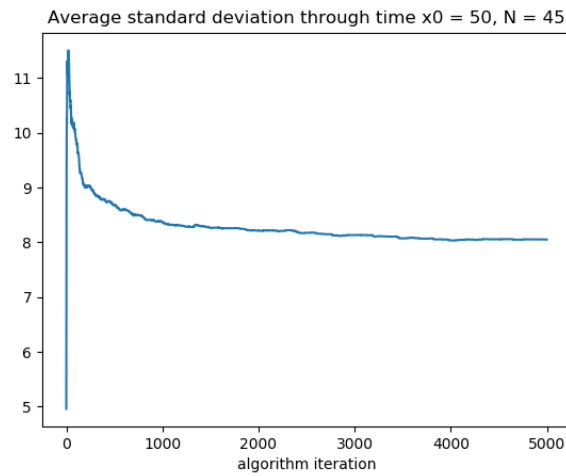
Rysunek 33: Najlepsze wyniki po zakończeniu działania,  $N=45$ .



Rysunek 34: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=45$ .

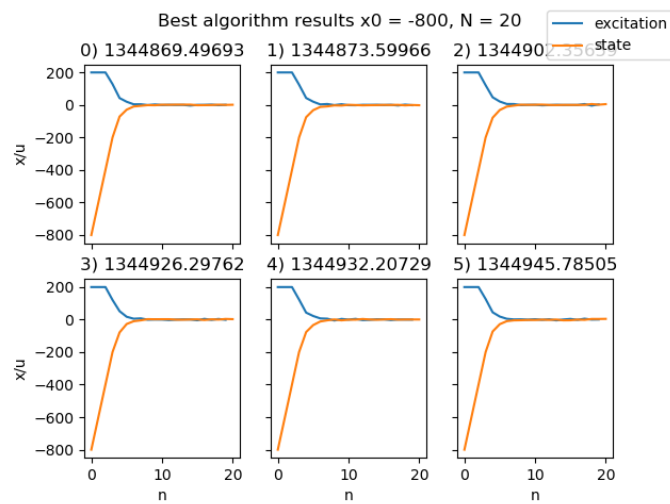


Rysunek 35: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=45$ .

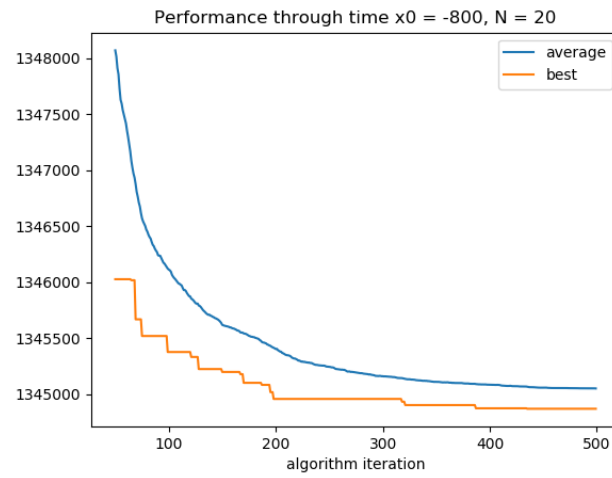


Rysunek 36: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=45$ .

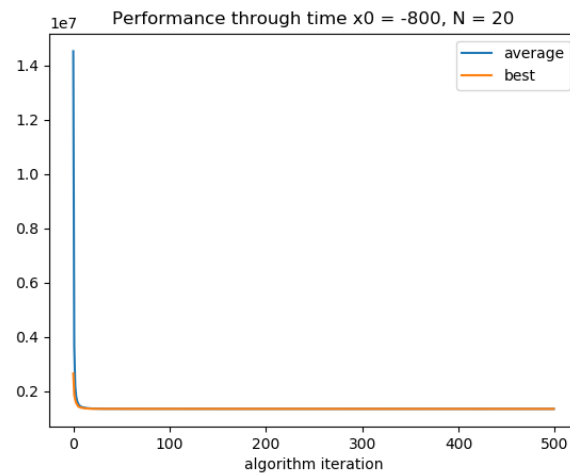
### 3.10 $x_0=-800$ , $N=20$



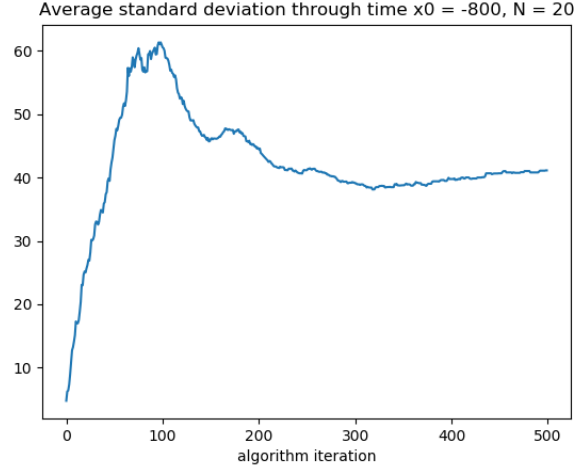
Rysunek 37: Najlepsze wyniki po zakończeniu działania,  $N=20$ .



Rysunek 38: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=20$ .



Rysunek 39: Wartość wskaźnika  $J$  w zależności od iteracji algorytmu,  $N=20$ .



Rysunek 40: Średnie pobudzenie standardowe w zależności od iteracji algorytmu,  $N=20$ .

## 4 Wnioski

Rozważając zadany problem optymalizacji łatwo dojść do wniosku, że w celu minimalizacji wskaźnika  $J$ , sterowanie powinno sprowadzać stan z  $x_0$  w okolice zera, a następnie utrzymywać go na tym poziomie. Rozwiązania znalezione przez algorytm rzeczywiście mają taką postać, co można zaobserwować min. na rysunkach 1, 5, 9. Dla wyższych wartości  $N$  rozwiązania wyglądają podobnie, jednak z powodu dużej liczby elementów w szukanym wektorze  $\mathbf{u}$ , nie zostało odnalezione minimum globalne. Po sprowadzeniu stanu w pobliże 0, pobudzenie nadal przyjmuje duże wartości, co powoduje zaburzenie  $\mathbf{x}$  i wzrost wskaźnika  $J$ , jak wyraźnie widać na rysunkach 25, 29 i 33. Oznacza to, że przyjęta strategia jest dość wolno zbieżna i w celu osiągnięcia minimum globalnego należy zwiększyć liczbę iteracji. Problemem nie jest „utknięcie” w minimum lokalnym, gdyż niemal każde zmiany zmniejszające fluktuacje pobudzenia (po początkowym okresie zmniejszania stanu) prowadziłyby do poprawy wskaźnika  $J$ .

Niezależnie od wartości  $N$ , wskaźnik  $J$  maleje gwałtownie w początkowej fazie działania algorytmu, co widać min. na rysunkach 3 i 27. W dalszej części optymalizacja następuje znacznie wolniej, jak przedstawiono na rysunkach 2 i 26. Widać na nich, że średnia wartość wskaźnika zbliża się do najlepszej.

Na rysunkach 37, 38, 39, 40 przedstawiono wyniki uzyskane dla  $x_0 = -800$ , co jest wartością kilkakrotnie większą (co do wartości bezwzględnej), niż maksymalne możliwe pobudzenie. W konsekwencji optymalne pobudzenie przyjmuje na początku maksymalną dopuszczalną wartość (200), co widać na rysunku 37.

Średnie odchylenie standardowe przedstawia jak dużej zmianie podlegają średnio osobniki podczas mutacji. Jego wykres w zależności od iteracji algorytmu wygląda na ogół podobnie i charakteryzuje się gwałtownym wzrostem i spadkiem w początkowej fazie, a następnie powolnym zmniejszaniem, co widać np. na rysunkach 12, 16 i 28.

Podsumowując można uznać, że zaimplementowana strategia działa poprawnie. Wskaźnik  $J$  jest znacznie zmniejszany w trakcie jej działania. Pomimo gwałtownej poprawy w początkowej fazie działania programu, dostrajanie wartości pobudzenia trwa jednak bardzo długo, szczególnie dla dużych przestrzeni poszukiwań. Nie udało się przez to odnaleźć minimum globalnego, przy większych wartościach parametru  $N$ .