



PRACA MAGISTERSKA

ZASTOSOWANIE ALGORYTMU GENERALIZED EXTREMAL OPTIMIZATION W PROBLEMIE SZEREGOWANIA ZADAŃ

Paweł Kaczanowski

Promotor: dr inż. Anna Piwońska

Zakres prezentacji

- Opis problemu szeregowania zadań
- Przedstawienie algorytmu uogólnionej ekstremalnej optymalizacji (GEO)
- Zastosowanie GEO w problemie szeregowania
- Przykład mutacji
- Przedstawienie wyników pracy
- Propozycja dalszych prac

Opis problemu

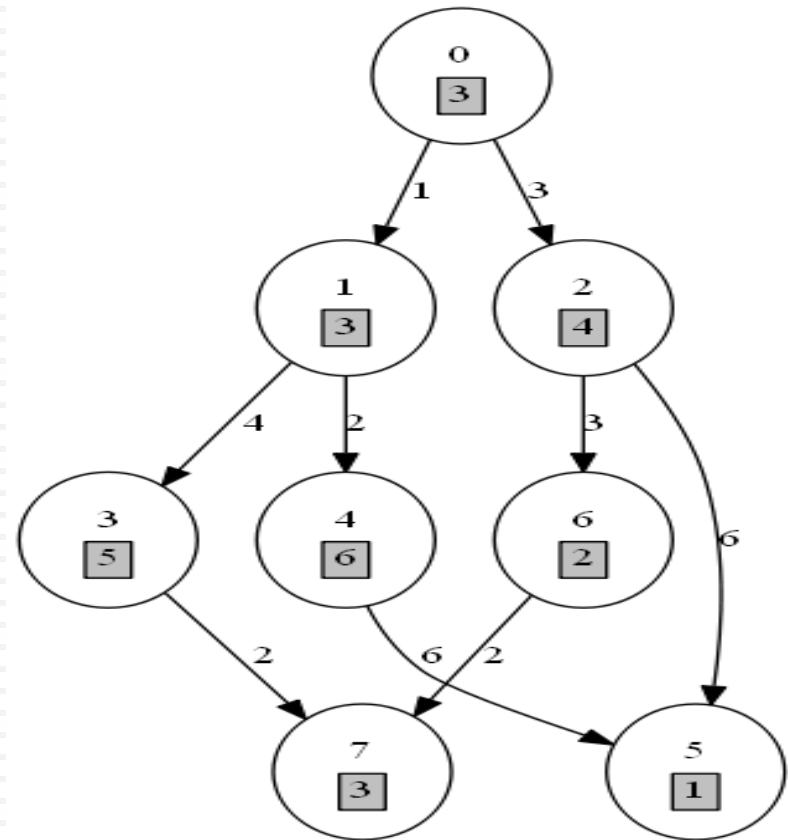
- Problem szeregowania należy do klasy problemów NP – zupełnych
- Nie może być rozwiązany w czasie wielomianowym
- Próba wykorzystania algorytmów heurystycznych przyspieszających znalezienie rozwiązania
- Zastosowany algorytm powinien cechować się dużą skutecznością

Problem szeregowania zadań

- Dotyczy wielu dziedzin życia
- Cele szeregowania
- Można dokonać podziału na wiele typów szeregowania:
 - ▣ Statyczne lub dynamiczne
 - ▣ Koszty przesyłania zadań
 - ▣ Przerywalne i nieprzerywalne
 - ▣ Liczba dostępnych zasobów (procesorów)

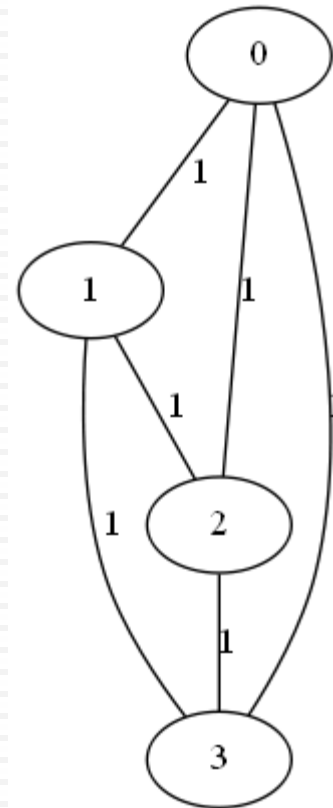
Graficzne przedstawienie problemu

- Acykliczny, skierowany graf ważony programu równoległego
- Wierzchołki reprezentują zadania
- Krawędzie reprezentują koszt przesyłania zadania pomiędzy procesorami



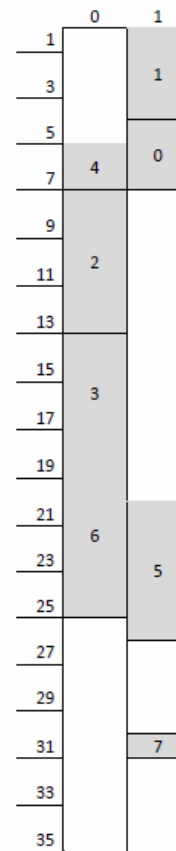
Graficzne przedstawienie problemu

- Nieskierowany graf systemu
- Krawędzie reprezentują dwukierunkowe połączenia pomiędzy procesorami

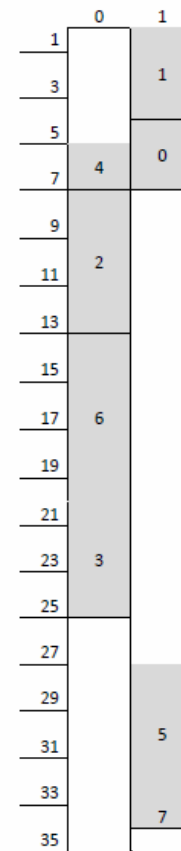


Polityki szeregowania

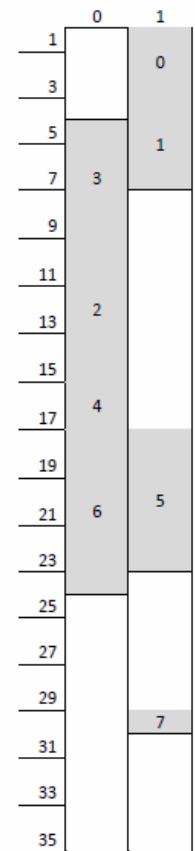
- Istnieje wiele polityk szeregowania:
 - ▣ Wg wysokości
 - ▣ Wg wysokości dynamicznej
 - ▣ Wg liczby następników
 - ▣ Losowo
 - ▣ Wg numerów zadań
 - ▣ Inne
- Ogromne znaczenie dla wyniku szeregowania



wysokość
dynamiczna



wysokość



liczba
następników

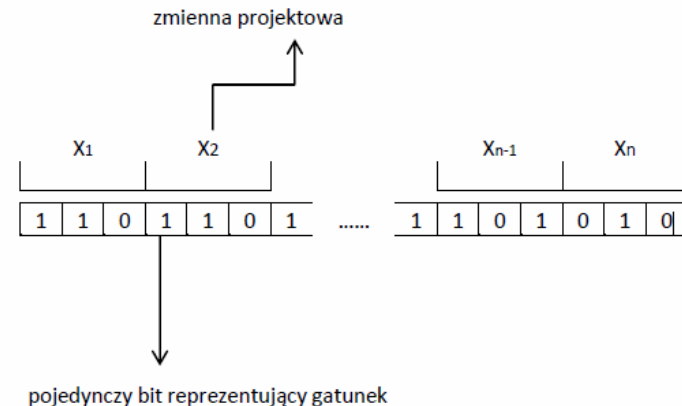
Algorytm uogólnionej ekstremalnej optymalizacji (GEO)

- Metaheurystyczny algorytm oparty na procesach zachodzących w naturze
- Zgodny z teorią samoregulującej się krytyczności (SOC)
- W GEO mutacji może być poddany dowolny osobnik

GEO w problemie szeregowania

- Stworzenie modelu na potrzeby problemu
- Zmienna projektowa zbudowana z m bitów wg wzoru:

$$2^m \geq \left\lceil \frac{(x_j^u - x_j^l)}{p} + 1 \right\rceil$$



GEO w problemie szeregowania

□ Inicjacja początkowej populacji

zadanie	0		1		2		3		4		5		6		7	
bity	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0

□ Iteracyjne mutowanie pojedynczych bitów

mutujący bit	zadania																T
	0		1		2		3		4		5		6		7		
1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	23
2	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	23
3	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	20
4	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	23
5	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	23
6	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	24
7	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	24
8	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	24
9	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	24
10	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	24
11	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	24
12	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	24
13	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	23
14	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	29
15	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	24
16	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	23

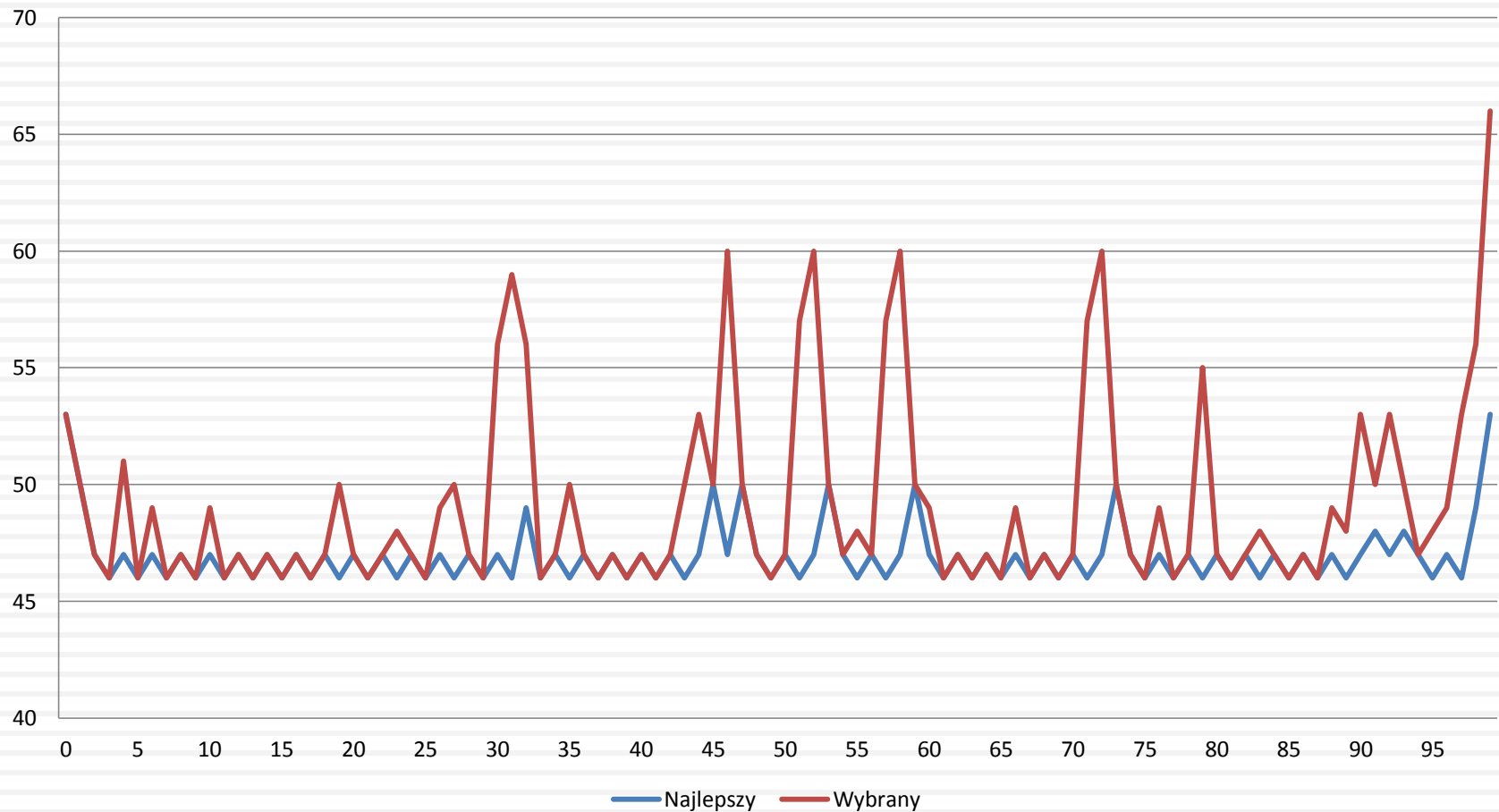
GEO w problemie szeregowania

- Wybór konfiguracji poddanej mutacji wg wzoru:

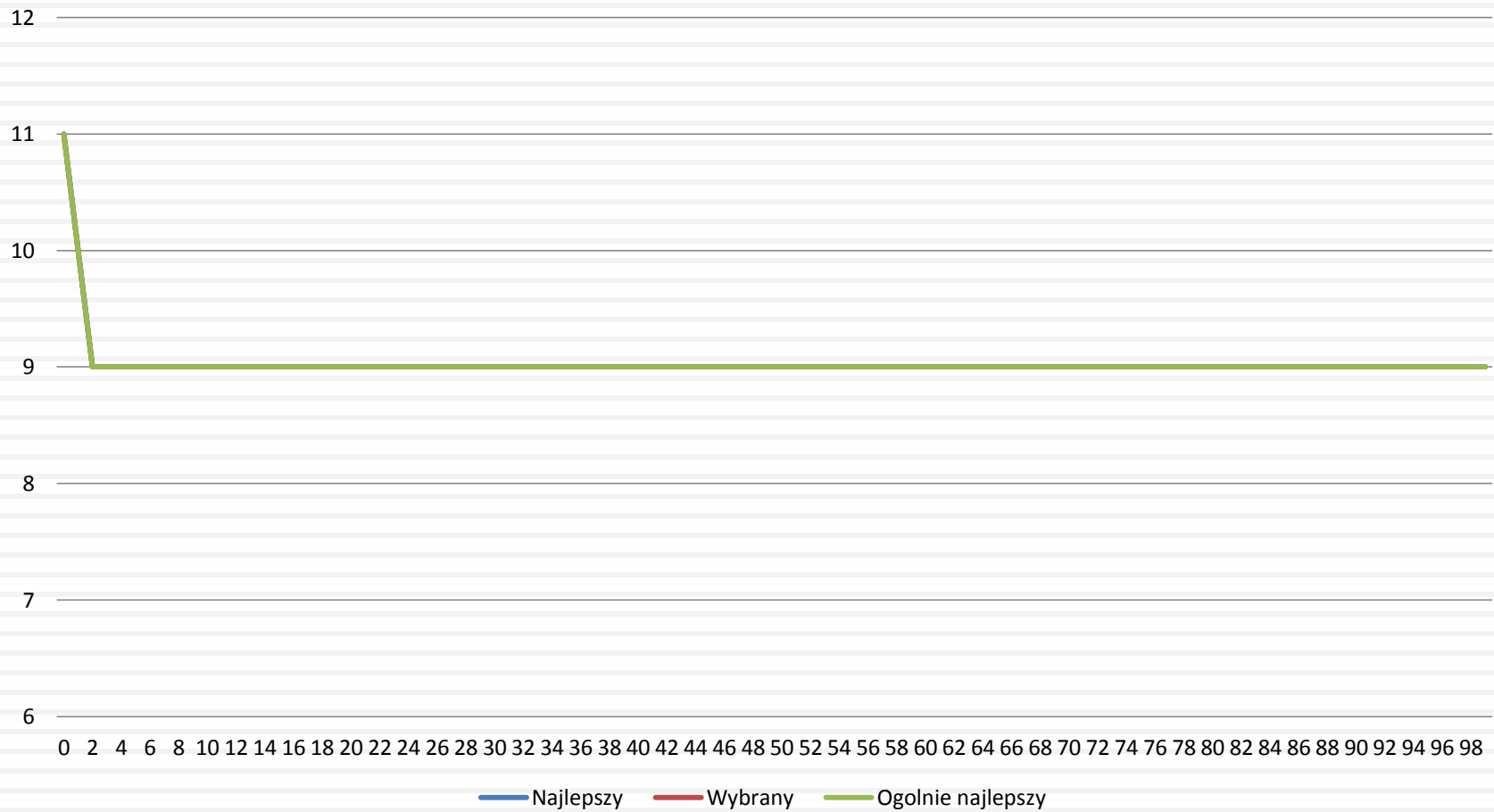
$$P_k = k^{-\tau}$$

ranking	mutujący bit	zadania														T		
		0		1		2		3		4		5		6			7	
1	3	0	0	1	0	1	0	0	1	0	0	0	1	0	0	0	0	20
2	1	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	23
3	2	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0	0	23
4	4	0	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	23
5	5	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	23
6	13	0	0	0	0	1	0	0	1	0	0	0	1	1	0	0	0	23
7	16	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	1	23
8	6	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	0	24
9	7	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	24
10	8	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	24
11	9	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	24
12	10	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	0	24
13	11	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	24
14	12	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	24
15	15	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	24
16	14	0	0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	29

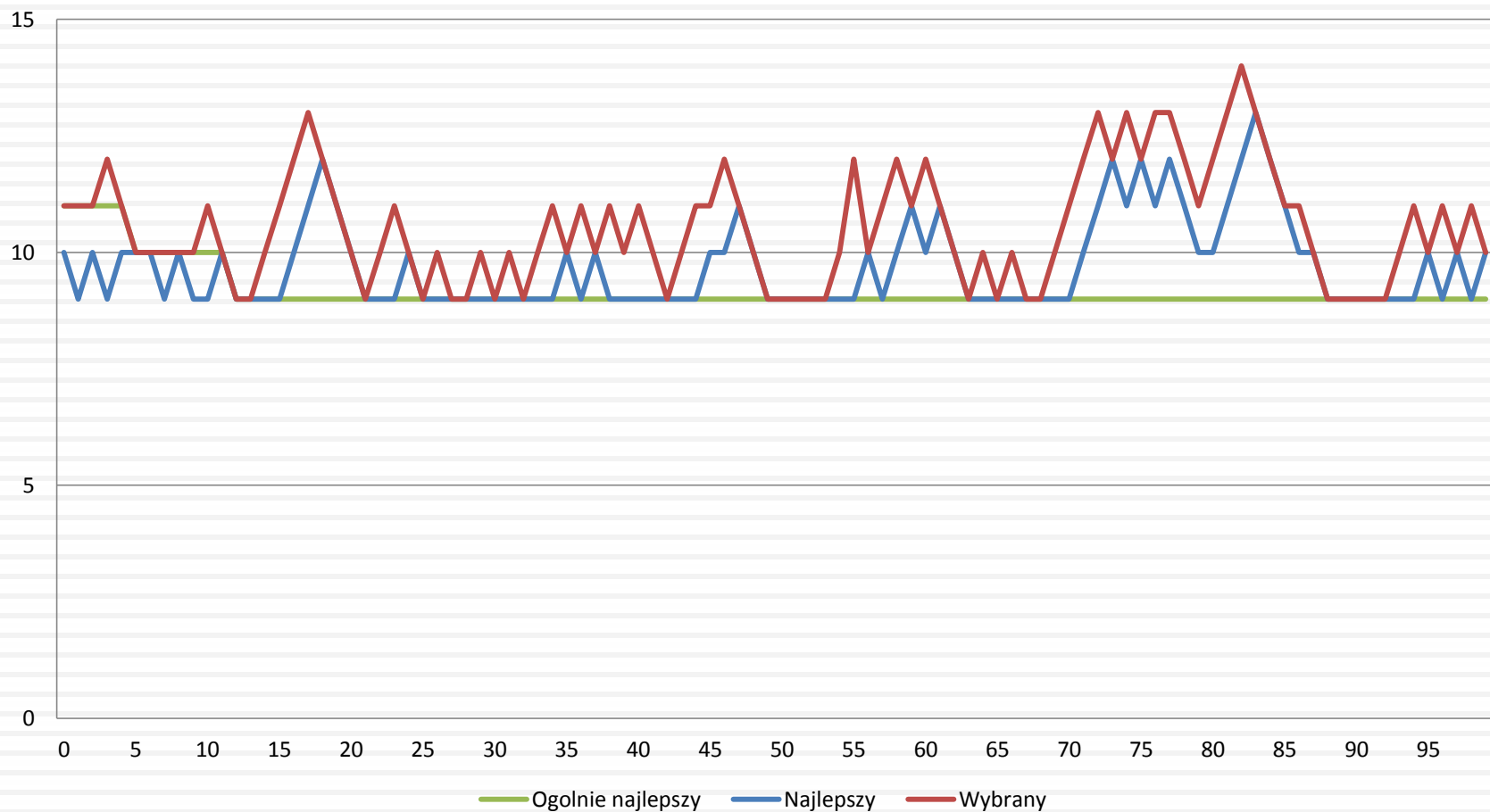
Typowy przebieg algorytmu $\tau = 0.5$



Przebieg dla $\tau = 8.0$



Przebieg dla $\tau = 0.1$



Zestawienie wyników

□ Na 2 procesorach – 1 000 przebiegów

graf programu	standardowy algorytm genetyczny	algorytm GEO
tree15	9	9
intree15	9	9
gauss18	44	44,128
g18	46	46
g40	80	80,001

□ Na 4 procesorach – 1 000 przebiegów

graf programu	standardowy algorytm genetyczny	algorytm GEO
tree15	7.0	7.0
g18	26.0	26.12
g40	45.0	45.4
gauss18	44.0	44.8

□ Na 8 procesorach – 500 przebiegów

graf programu	standardowy algorytm genetyczny	algorytm GEO
tree15	7.0	7.0
g18	24.0	24.01
g40	33.0	33.0
gauss18	44.0	45.7

Podsumowanie wyników

- GEO znajduje najlepsze rozwiązania dla prostych i złożonych grafów programu
- Może być stosowany do nieograniczonej liczby procesorów, jednak znacznie wydłuża się czas poszukiwania i ilość wykonań funkcji celu (określenia czasu szeregowania) wraz ze wzrostem liczby procesorów

Podsumowanie wyników

- Bardzo istotny dobór parametru prawdopodobieństwa τ , zalecany w przedziale $[0.5;1.0]$
- Określenie odpowiedniej ilości iteracji – kryterium stopu
- Czasami zauważalna jest zbyt wolna mutacja, zwłaszcza przy wprowadzeniu wielu procesorów

Propozycja dalszych prac

- Zmiany parametru τ na funkcję kary
- Modyfikacja algorytmu pozwalająca za zwiększenie liczby modyfikowanych w jednej iteracji zmiennych projektowych



Dziękuję za uwagę