

Sprawozdanie do projektu z Metaheurystyk

Optymalizacja Funkcja Schaffera F7
Metoda Wyważania
Mutacja o rozkładzie normalnym
Kodownie osobnika

Wstęp.

Celem projektu było przeanalizowania i dostrojenie parametrów algorytmu symulowanego wyważania w zadaniu minimalizacji funkcji Schaffera F7.

W tym celu został napisany program w języku Java implementujący algorytm symulacji wyważania. Program poza samym algorytmem posiada funkcjonalność zapisywania wyników pośrednich i summarycznych w plikach „.csv”. Ponadto program umożliwia konfigurację parametrów algorytmu.

Opis programu.

Program został napisany w języku *Java*, do kompilacji wymagane jest narzędzie *maven*. Aby skompilować program należy go rozpakować do katalogu poczym z tego katalogu uruchomić komendę z po przez polecenie:

mvn clean install

Uruchomić programu można za pomocą polecenia:

java -jar target/meh-jar-with-dependencies.jar src/main/resources/p1.properties

target/meh-jar-with-dependencies.jar – ścieżka do programu wykonywalnego
src/main/resources/p1.properties – ścieżka do pliku konfiguracyjnego

Program posiada plik konfiguracyjny w którym znajdują się parametry dla algorytmu wyważania oraz parametry programu.

```
# ilość powtórzeń całego algorytmu.  
simulation.count=10  
  
# ilość mutacji w stałej temperaturze.  
max.iteration.per.temperature=50  
  
# maksymalna ilość zmian temperatury  
# ( warunek stopu z jak parametr „temperature.end”)  
max.iteration=2000  
  
# ilość wymiarów w jakiej optymalizowana jest funkcja  
dimensions=1  
  
# jeśli ten parametr jest większy niż parametr dimensions to program wykona symulacje  
# tyle razy ile wynosi różnica.  
# przykład: dimensions=1;dimensions.end=5;  
# program wykona symulacje dla zadania jedno wymiarowego,
```

```
# dwu wymiarowego, trzy wymiarowego, cztero i pięć wymiarowego,  
dimensions.end=1  
  
# Zasięg mutacji. Losowana jest liczba -1,1 z rozkładu normalnego i  
# mnożona przez ten parametr.  
mutations.range=10  
  
# Schemat chłodzenia: geometryczny, liniowy, logarytmiczny, other1  
# parametr cooling.alpha brany pod uwagę tylko przy schemacie  
# geometrycznym,liniowym  
cooling=geometryczny  
  
# Pr parametr alpha używany w schematach chłodzenia geometrycznym i  
# liniowym  
cooling.alpha=0.8  
  
# jeśli ten parametr jest ustawiony na „true” to program wykona symulacje dla  
# wszystkich dostępnych schematów chłodzenia. Jeśli „false” tylko dla tego  
# schematu który jest wpisany w parametrze „cooling”.  
cooling.all=false  
  
# temperatura z której zaczyna algorytm.  
temperature.start=1000  
  
# temperatura przy której osiągnięciu algorytm się zatrzymuje.  
# ( warunek stopu z jak parametr „max.iteration”)  
temperature.end=0.5  
  
# prefix z jakim będą generowane pliki z wynikami  
result.file=result  
  
# parametr K oznaczający stałą Boltzmanna.  
K=0.1
```

Kodowanie rozwiązania (osobnika).

Głównymi elementami programu są 3 klasy:

Solution – kodowanie rozwiązania (osobnika)

Algoryt

Założenia doświadczeń.

Doświadczenia były przeprowadzane w seriach po 10 przebiegów pełnego algorytmu wyważania dla jednego zestawu parametrów.

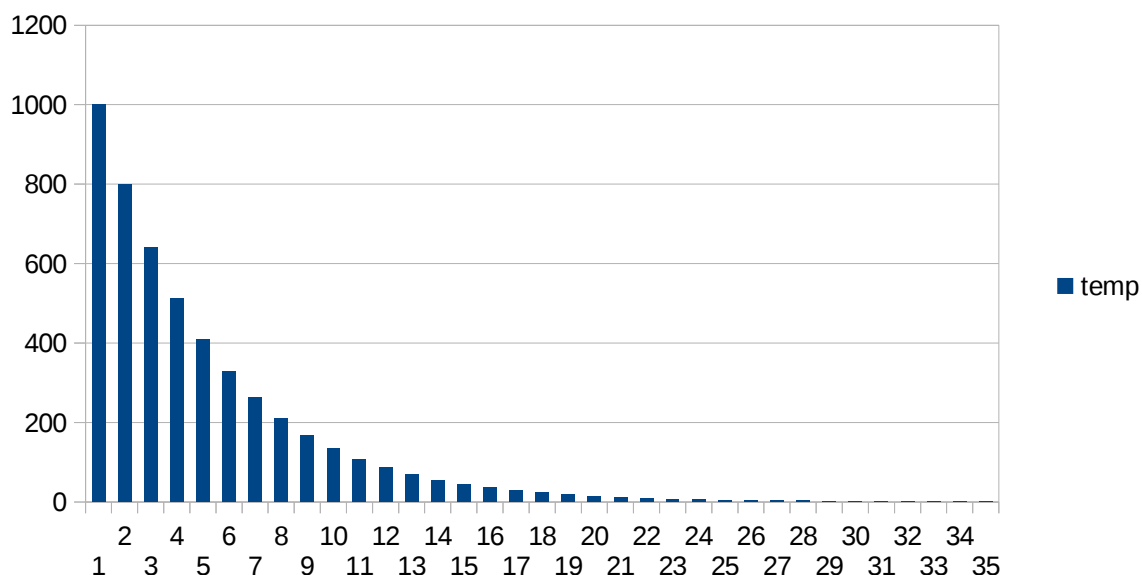
Doświadczalnie wybrano temperaturę starową na 1000 jednostek a warunek stopu na bliskiej zeru tzn. 0.5. Okazało się jednak że przy schemacie chłodzenia Logarytmicznym warunek stop $temperatura < 0.5$ wymagał bardzo długiego czasu do zakończenia działania dla tego dodano drugi warunek stop ograniczający ilość zmian temperatur do 2000.

Przy tak dobranych parametrach startowych przystąpiono do analizy 3 parametrów algorytmu. Wszystkie 3 parametry były strojone dla zadania 2 wymiarowego.

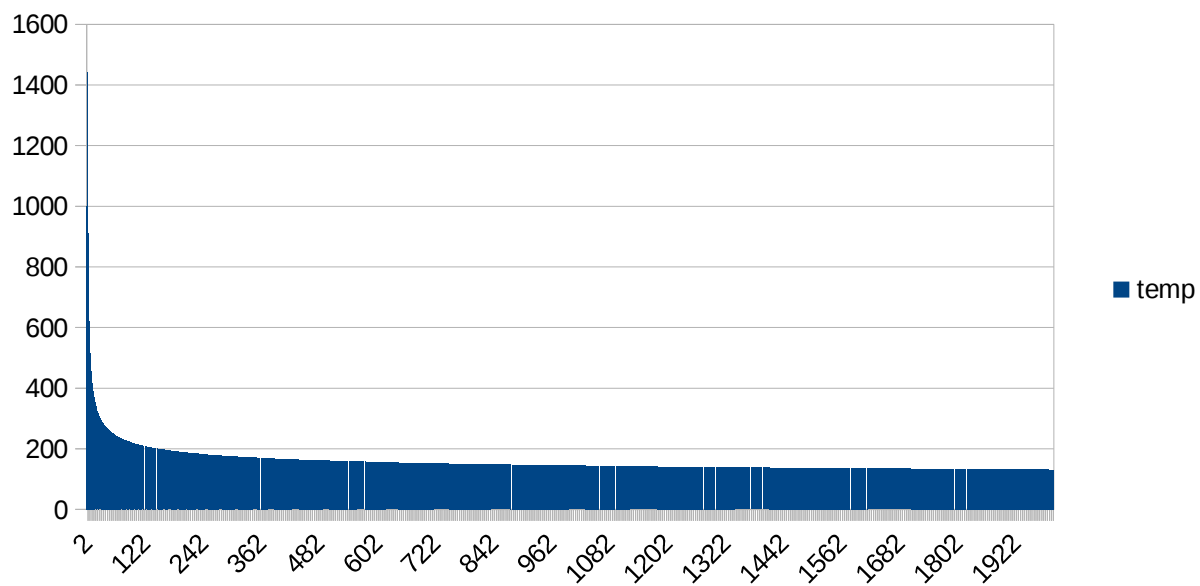
K – stała boltzman od której zależy prawdopodobieństwo z jaką może zostać przyjęty gorszy wyniki niż obecny

cooling - schemat schładzania zostały wybrane 4:

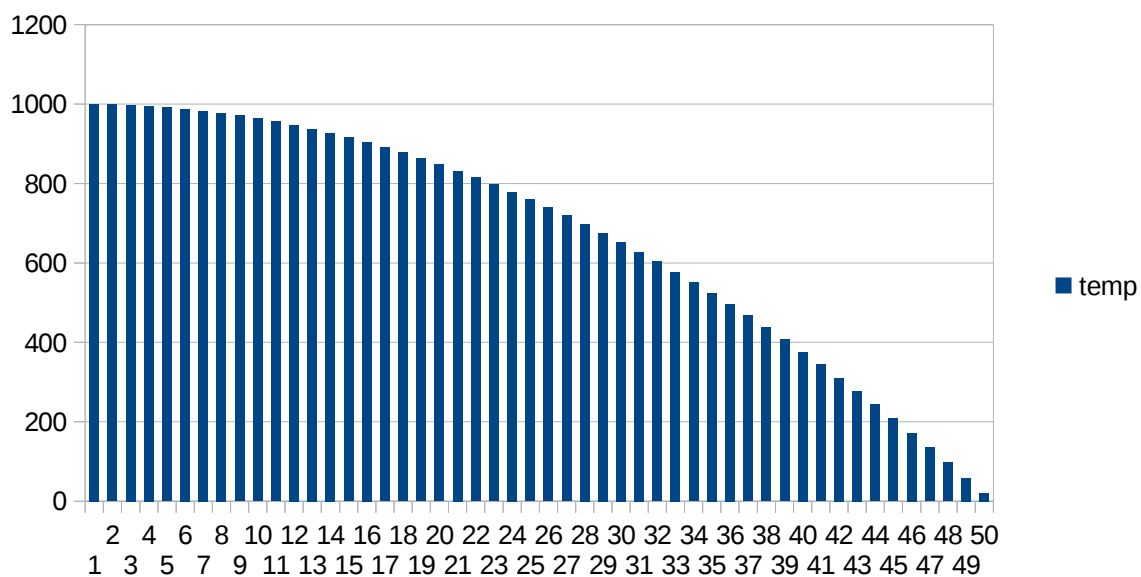
- geometryczny



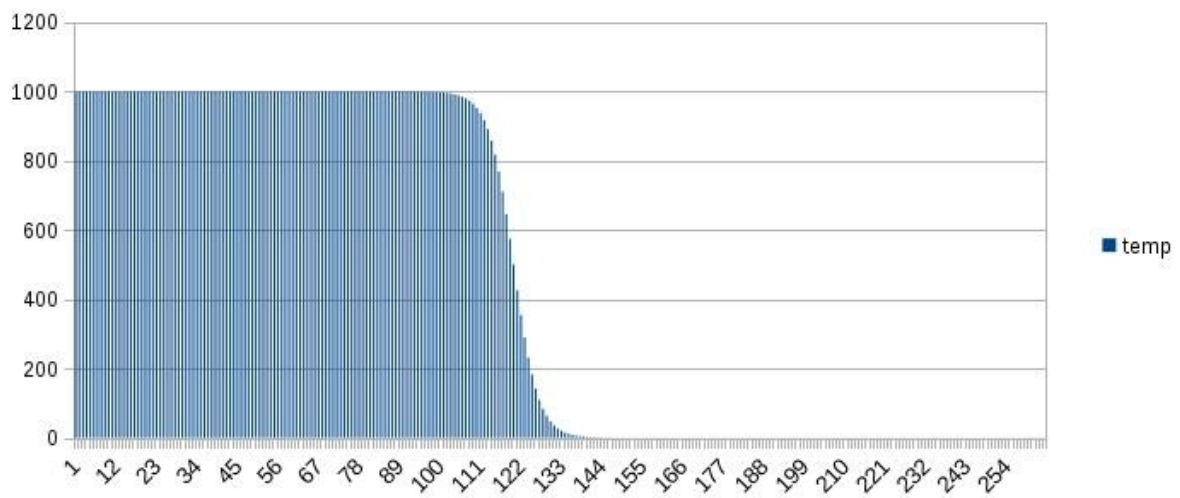
- logarytmiczny



- liniowy



- o rozkładzie: oś pionowa – temperatura, oś pozioma – ilość zmian temperatur



mutations.range - zasięg mutacji o rozkładzie normalnym.

Dodatkowo został przeprowadzony eksperyment sprawdzający jak poszczególne z wybranych schematów chłodzenia zachowują się przy zwiększaniu wymiarów zadania.

Wyniki

1. Strojenie parametru K (stała boltzman)

1. K = 1 dla każdego z schematów schładzania:

○

○

id	best-x1	best-x2	best-value	worst-x1	worst-x2	worst-value
0	2,468	7,318	2,837	100	91,888	23,307
1	-1,883	5,898	2,651	-92,238	99,731	23,309
2	-4,215	2,623	2,341	-100	-92,043	23,309
3	4,318	-4,957	3,409	100	-91,997	23,309
4	9,897	-6,29	3,48	-100	92,006	23,309
5	-0,251	4,017	2,02	100	-92,172	23,303
6	1,688	0,908	1,595	-100	92,063	23,308
7	1,12	0,041	1,695	-92,053	-100	23,308
8	-0,846	-2,263	1,557	100	92,032	23,309
9	0,463	-0,727	1,833	-97,418	-94,941	23,299

				Liniowy		
id	best-x1	best-x2	best-value	worst-x1	worst-x2	worst-value
0	5,007	-0,153	2,262	92,763	100	23,187
1	1,452	0,193	1,549	100	-91,509	23,261
2	1,323	2,654	2,422	100	92,109	23,306
3	0,823	4,038	2,332	-67,725	-100	21,952
4	-1,613	-1,04	1,607	100	91,26	23,199
5	2,909	2,716	2,003	91,839	-100	23,304
6	-1,079	-2,129	1,598	100	-93,696	22,72
7	-1,149	3,911	2,143	-100	-93,22	23,001
8	-0,342	-1,792	1,384	100	91,89	23,307
9	1,858	-2,534	1,775	-100	69,868	21,051