

Zastosowanie algorytmu symulowanego wyżarzania

Dla problemu ładowania miejskich hulajnóg elektrycznych

**Jan Sawicki
Hubert Sujka
Jakub Pudło**

1. Opis problemu

Analizowanym problemem jest zagadnienie zbieracza hulajnóg elektrycznych jakich obecnie możemy wiele zaobserwować na ulicach naszego miasta. Polega ono na wybraniu optymalnej trasy zbierania hulajnóg do ładowania przez noc aby zysk zbieracza był jak największy.

2. Model zagadnienia

Ze względu na brak posiadania przez nas nadających się do przetwarzania danych na temat rozmieszczenia hulajnóg przyjęliśmy ich rozkład na mapie przez funkcję losową.

Ponadto założyliśmy że między dwoma dowolnymi punktami na mapie można przemieścić się po trasie będącej linią prostą.

Ze względu na uwagi do naszego algorytmu kierowane podczas zajęć przyjęliśmy że mogą one mieć różną wagę i cenę za ich pozbieranie (w rzeczywistości hulajnogi danej firmy mają równą masę i cenę za ich zebranie i naładowanie) co wpłynęło na zwiększenie analizowanych zmiennych w naszym algorytmie.

Funkcją celu naszego algorytmu przy tych założeniach jest układ

$$f = \sum_{i=0}^n p_i - a \left(\sum_{i=0}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} + \sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} \right)$$
$$\sum_{i=0}^n m_i \leq b$$

,gdzie:

a - cena paliwa/km

m_i - masa hulajnogi

p_i - zysk za zebranie hulajnogi

x_i, y_i - współrzędne hulajnogi

x_0, y_0 - adres człowieka zbierającego hulajnogi

b - pojemność samochodu

n - zbiór hulajnóg do których zdecydowaliśmy się

3. Algorytm

Teoria:

Algorytm symulowanego wyżarzania należy do algorytmów heurystycznych. Cechą charakterystyczną tej metody jest występowanie parametru sterującego zwanego temperaturą, który maleje w trakcie wykonywania algorytmu.

Opis algorytmu:

Przed rozpoczęciem algorytmu wybieramy parametry początkowe:

- Początkowa wartość temperatury T ,
- Wartość temperatury minimalnej T_{min}
- Sposób obniżania temperatury $\alpha < 1$
- Liczbę prób przeprowadzanych w ramach jednej epoki

Działanie algorytmu:

1. Losowanie rozwiązania początkowego X
2. Wybranie losowego rozwiązania X'
3. Jeżeli nowe rozwiązanie jest lepsze to przyjmujemy ($X=X'$) w przeciwnym razie należy wyznaczyć prawdopodobieństwo przyjęcia nowego rozwiązania używając wzoru $e^{\frac{-\Delta}{T}}$. Następnie losuje się liczbę z przedziału $[0,1]$ i jeśli jest ona mniejsza od liczonego prawdopodobieństwa, przyjmujemy je jako nowe rozwiązanie (pomimo, że jest gorsze)
4. Sprawdzamy, czy wykonano odpowiednią liczbę prób w danej epoce, jeśli nie wykonano wróć do 2.
5. Zmniejsz temperaturę $T = \alpha \cdot T$
6. Jeżeli $T > T_{min}$ wróć do pkt 2.

4. Aplikacja

Wymagania odnośnie uruchomienia

Do uruchomienia aplikacji jest wymagany interpreter języka Python3 w wersji 3.5 lub wyższej z zainstalowanymi następującymi pakietami:

- tkinter (domyślna biblioteka graficzna języka Python3 instalowana razem z interpreterem)
- numpy (pakiet pip)
- matplotlib (pakiet pip)
- colorama (pakiet pip)

Format Danych i wyników

Dane przekazywane są w formie specjalnie przygotowanego pliku tekstowego. Przykładowy plik powinien wyglądać w sposób (tekst wyboldowany to komentarz nie należy go umieszczać w pliku z danymi):

```
[26, 26, 26, -36, -36, -36, -36, 59, 59, -13] #Pozycje "x" hulajnóg
[56, 56, 56, 29, 29, 29, 29, 3, 3, 58] #Pozycje "y" hulajnóg
[37, 37, 40, 39, 33, 40, 39, 22, 40, 45] #Ceny za zabranie hulajnogi
[0.898, 1.006, 0.847, 1.192, 1.058, 0.939, 0.937, 1.034, 0.954, 1.045] #Wagi hulajnóg
[-60, 60] #Rozmiar "x" mapy
[-60, 60] #Rozmiar "y" mapy
10 #Ilość miejsc z hulajnogami
[20, 50] #Przedział w którym generowane są ceny
5 #Pojemność samochodu
1 #Cena paliwa
```

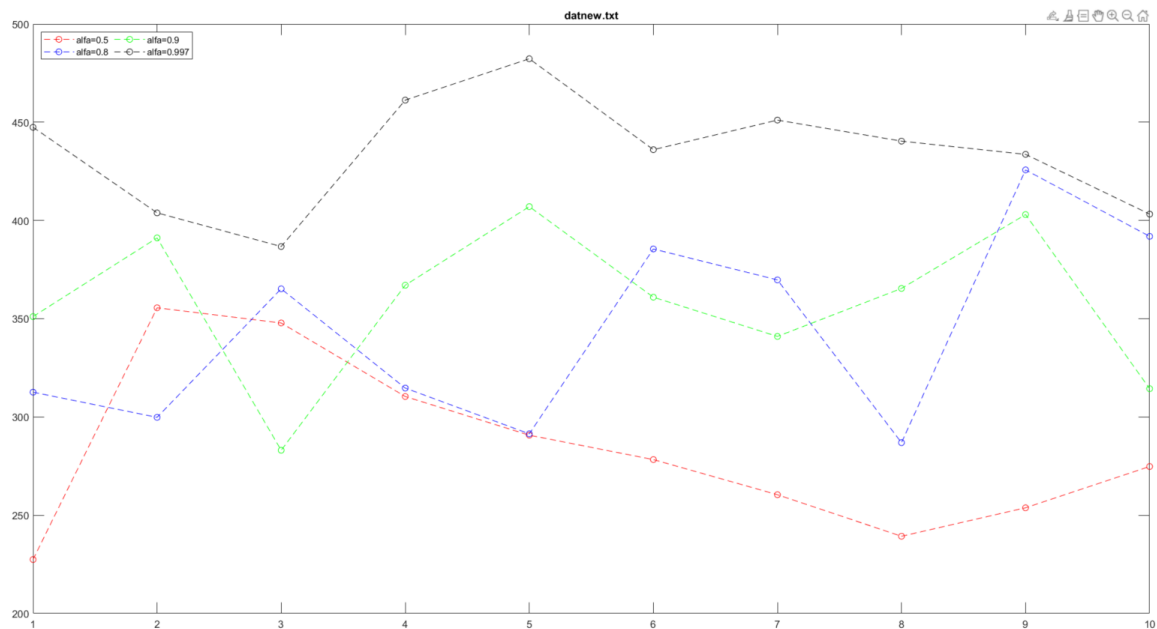
Wynikami jest wykres "Mapa miasta" gdzie przedstawiona jest droga dostawcy, wykres "przebieg algorytmu" gdzie widać zmieniającą się wartość funkcji celu oraz dane na konsoli przedstawiające po kolei ścieżkę dostawcy (wraz z parametrami danych wierzchołków) oraz ostateczną wartość funkcji celu, do lepszego śledzenia wyników.

Krótko opisana funkcjonalność (w punktach):

1. Generowanie plików z danymi do testowania oprogramowania. Plik generate_datafile.py (interfejs konsolowy).
2. Graficzny Interfejs użytkownika. Plik main.py :
 - Ustawianie parametrów uruchomienia algorytmu
 - Ustawianie parametrów generowania losowych danych na potrzeby symulacji
 - Wczytywanie danych z pliku przez podanie ścieżki do wczytywanego pliku
 - Sprawdzanie poprawności formatu podanych danych
 - Wyświetlanie wyników działania algorytmu w postaci wykresów
 - Wyświetlanie obecnie ustawionych parametrów algorytmu w czasie rzeczywistym
 - Uruchamianie symulacji

5. Testy

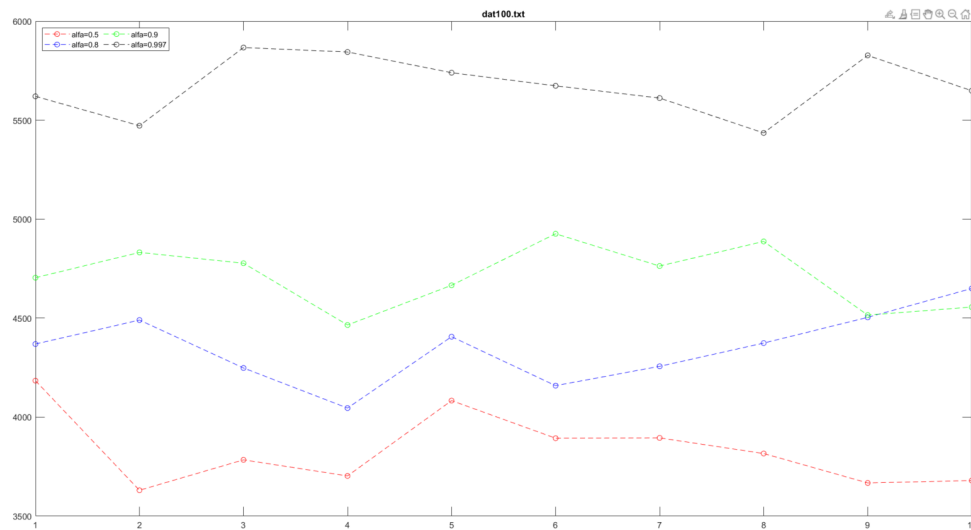
5.1. Sprawdzenie wpływu parametru α na jakość działania algorytmu



Rys1. Dane początkowe: $T=21370$, $T_{min}=0.1$, ilość hulajnóg 10, pojemność samochodu 5,

Tabela 1: Wyniki dla różnych wartości parametru α przy pojemności samochodu = 5

α	0,5	0,8	0,9	0,997
Średnia	283,9	344,4	358,4	434,6
Odchylenie standardowe	43	49	38	29
Wartość max	355,6	425,7	407,1	482,3

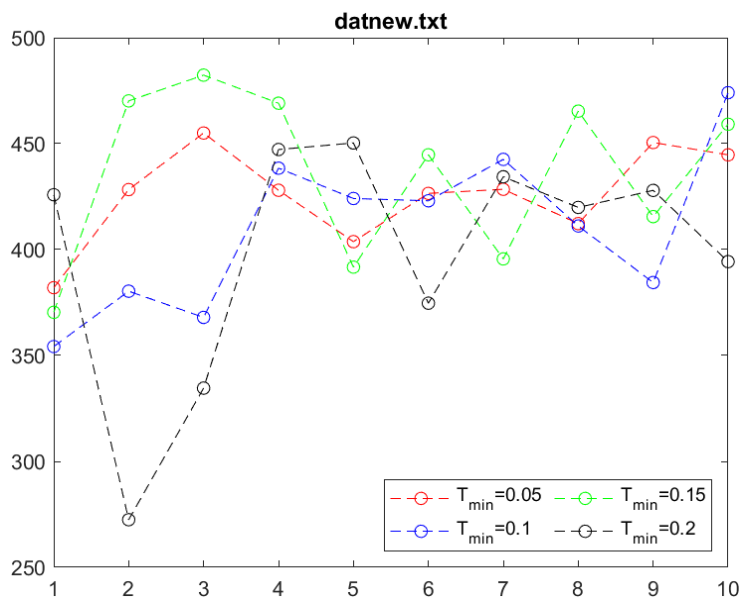


Rys2. Dane początkowe: $T=21370$, $T_{min}=0.1$, ilość hulajnog 100, pojemność samochodu 50

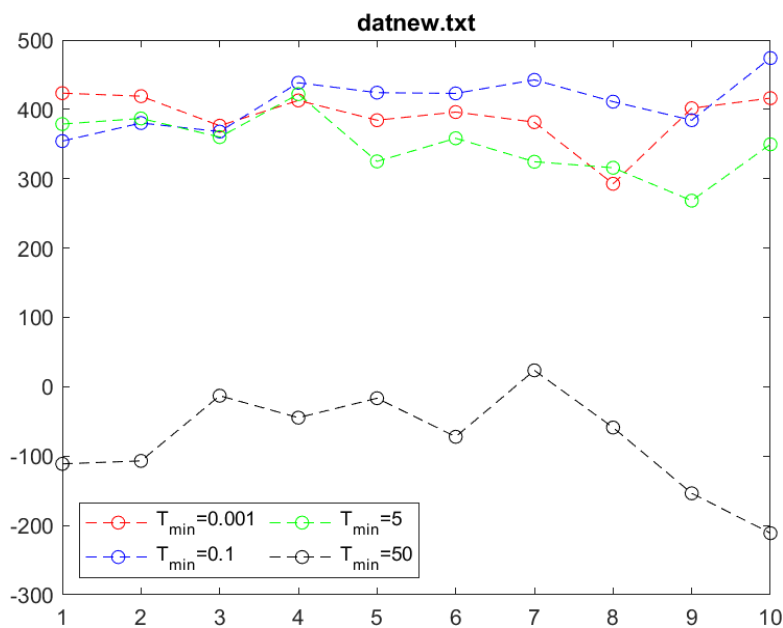
Tabela 2: Wyniki dla różnych wartości parametru α przy pojemności samochodu = 50

α	0,5	0,8	0,9	0,997
Średnia	3833,5	4350,2	4709,2	5673,9
Odchylenie standardowe	184	178	157	148
Wartość max	4183,9	4649,6	4926,1	5866,7

5.2. Sprawdzenie wpływu parametru T_{min} na jakość działania algorytmu



Rys3. Dane początkowe: $T=21370$, $\alpha=0.98$, ilość hulajnóg 100, pojemność samochodu 50

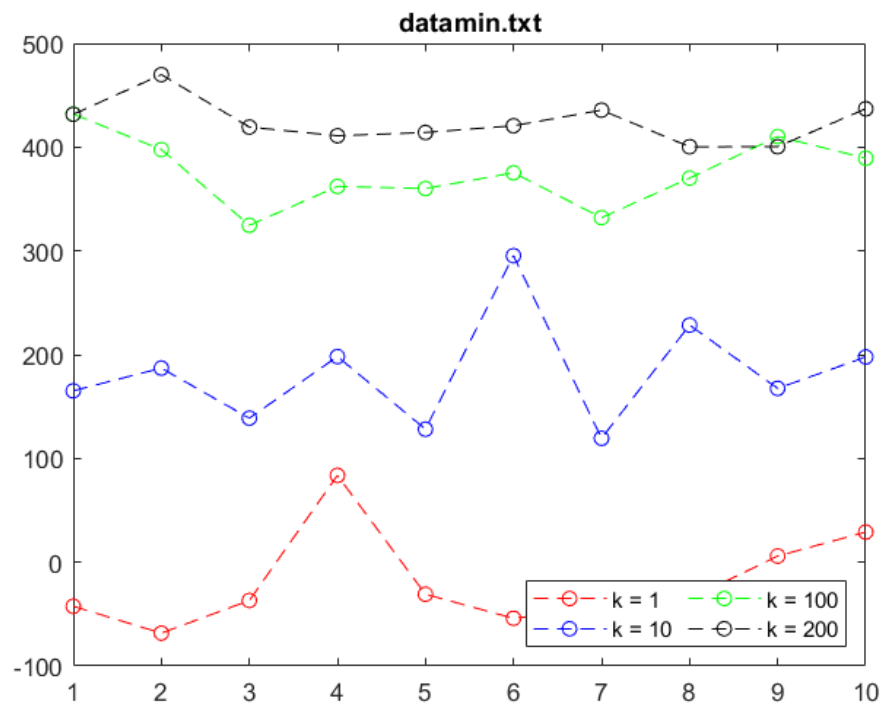


Rys4. Dane początkowe: $T=21370$, $\alpha=0.98$, ilość hulajnóg 100, pojemność samochodu 50

Tabela 3: Wyniki dla różnych wartości parametru T_{min}

T_{min}	0,001	0,5	0,1	0,15	0,2	5	50
Średnia	390,44	425,95	410,03	436,41	398,17	348,95	-76,39
Odchylenie standardowe	38,14	22,19	37,64	39,72	56,73	43,01	70,89
Wartość max	423,38	455,014	474,049	482,336	450,323	422,01	23,69

5.3. Sprawdzenie wpływu parametru k na jakość działania algorytmu, gdzie k to krotność powtórzeń dla danej epoki



Rys 5. Dane początkowe: $T=21370$, $\alpha=0.9$, ilość hulajnóg 50, pojemność samochodu 20

Tabela 4: Wyniki dla różnych wartości parametru K

K	1	10	100	200
Średnia	-19,43	182,60	375,33	423,98
Odchylenie standardowe	46,05	52,41	33,29	20,76
Wartość max	83,67	295,41	431,86	469,87

6. Podsumowanie

Wnioski

Test 1, pokazuje nam, że wartość parametru α jest bardzo ważna oraz że powinna znajdować się powyżej wartości 0,9. Najlepsze wyniki uzyskaliśmy dla wartości 0,997 czyli im wartość α jest bliższa 1 to algorytm pokazuje lepsze wyniki, lecz wydłuża się czas pracy takiego algorytmu.

Test 2, pokazuje że niewielkie zmiany wartości T_{\min} w okolicach zakresu pracy (rysunek 3) nie wpływają znacząco na zmianę działania algorytmu. Natomiast na podstawie drugiego testu (rysunek 4) gdzie zmiany były znaczące widać że zbyt duża wartość T_{\min} może wpłynąć negatywnie na działanie algorytmu nawet poprzez spowodowanie zwracania negatywnych(niepoprawnych, bo zakładających stratę) wyników.

Test 3, wykazuje, że wpływ liczby powtórzeń dla jednej epoki jest znaczący. Lepsze wyniki oraz najmniejsze odchyły otrzymujemy dla coraz to większej liczby powtórzeń. Dla bardzo małej liczby powtórzeń algorytm wyrzuca bardzo złe wyniki, nawet wychodzące poniżej 0.

Stwierdzone problemy

Kierunki dalszego rozwoju

Można wzbogacić algorytm o dodatkową analizę trasy przejazdu w rzeczywistych warunkach drogowych(siatka ulic) z uwzględnieniem godzin szczytu i najbardziej zakorkowanych miejsc co by wpływało na koszt przejazdu z punktu do punktu. Jeżeli dodatkowo połączyć by naszą aplikację z aplikacjami firm hulajnogach można by z nich pobierać dane i wykorzystać do analizy rzeczywistych przypadków. Niestety wymagałoby to również danych i umiejętności których nie posiadamy.