Sprawozdanie MOWNiT

Laboratorium 5

Paweł Maczuga

W pliku pdf nie działają gif-y
Google doc:
https://docs.google.com/document/d/19_Y42QnujCYYTgSBbGrndVQLDTvsAKbGO9eycey9VYM/edit?usp=sharing

zad. 1

Wyniki dla 10 miast:
Wyżarzanie:
Time annealing: 0.0464174747467041
energy difference: 226.3507817412028

Algorytm dokładny:
Time precise: 28.934009313583374
energy difference: 258.30167277735427

Różnica w czasach jest ogromna, w energii dość niewielka.

zad. 2

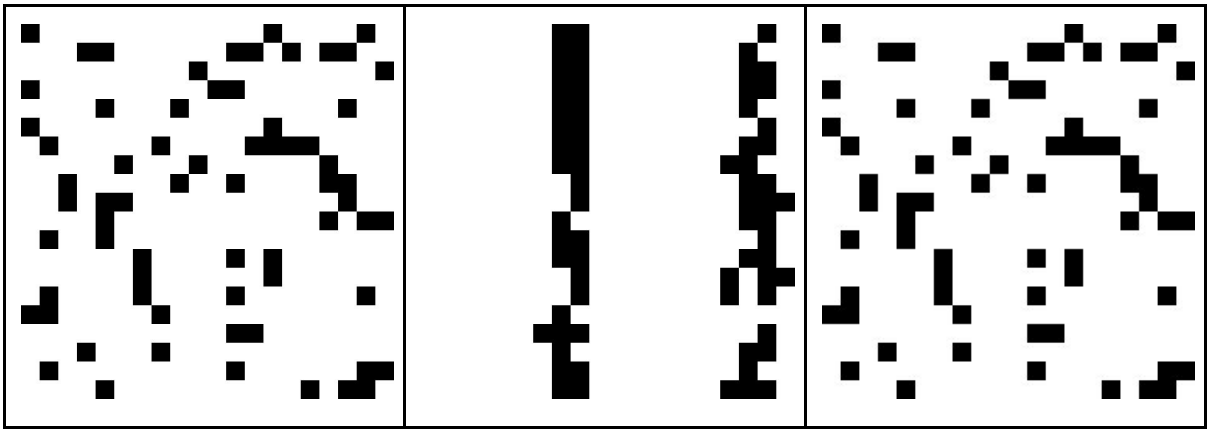
Symulacja własnej grawitacji:
Na białej płaszczyźnie znajdują się czarne punkty.
Aby wyznaczyć symulowane wyżarzanie należy dostarczyć funckję wyliczającą energię punktów. Argumentami tej funkcji są:

- funkcja wyznaczającą energię w zależności od pozycji punktu
- funkcja wyznaczająca energię w zależności od odległości od pozostałych puntków

Odpowiednio:
position_fun(y, x)
between_fun(r)

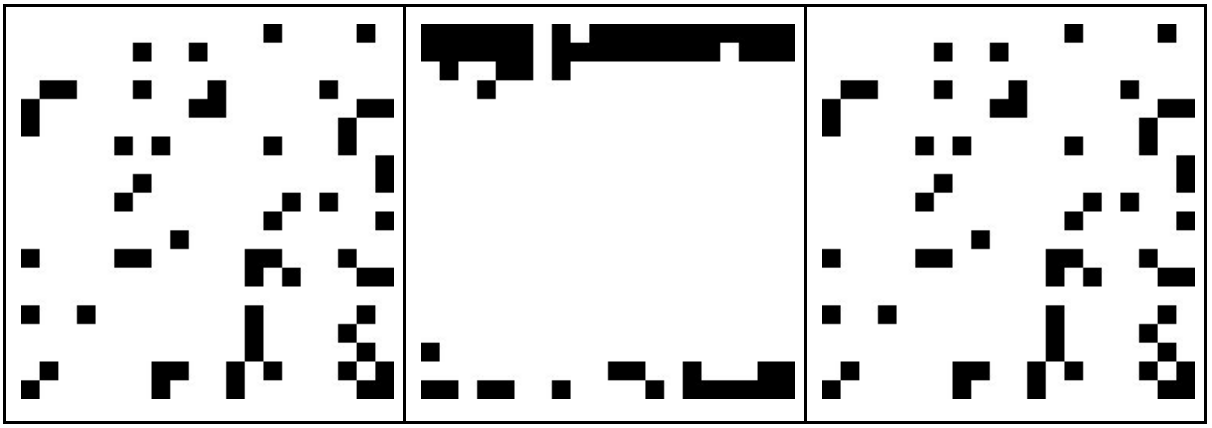
Kilka przykładowych grawitacji wraz z wizualizacją:

Skupiska wzdłóż osi ox:
*posistion_fun = lambda y, x: 1000 * math.sin(x * 4 * math.pi)*
between_fun = lambda r: 0



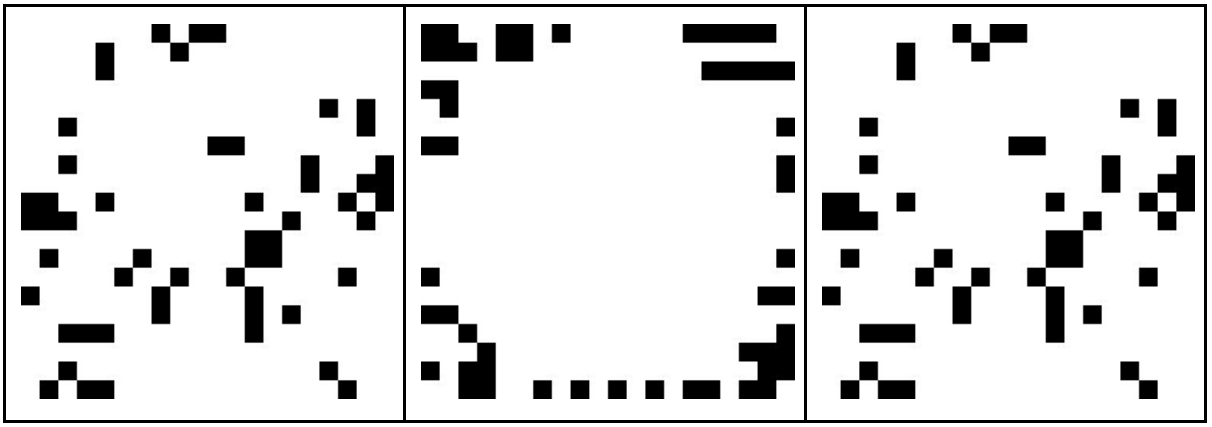
Przyciąganie w górę i w dół:

*posistion_fun = lambda y, x: 500 * math.sin(y * math.pi)*
between_fun = lambda r: 0



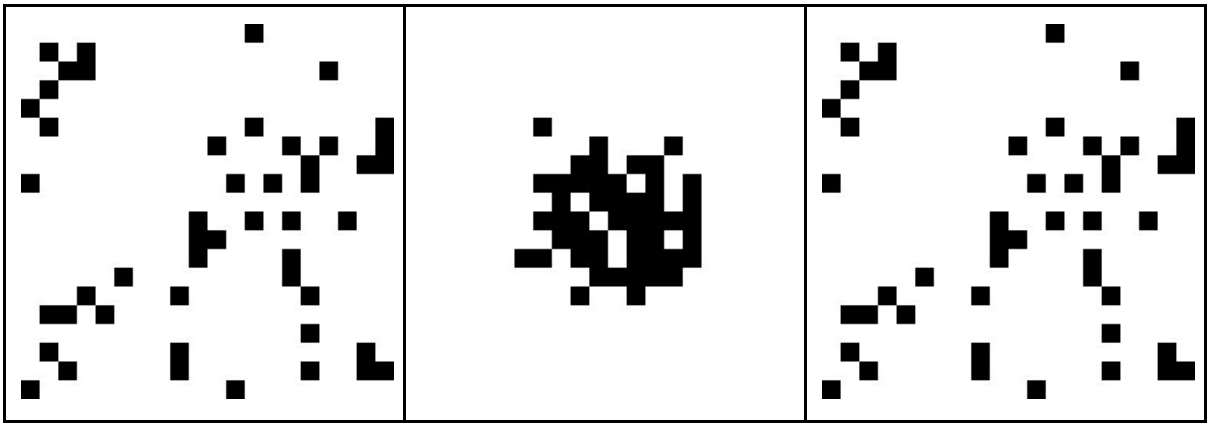
Przyciąganie góra, dół plus punkty się odpychają:

*posistion_fun = lambda y, x: 500 * math.sin(y * math.pi)*
between_fun = lambda r: 0



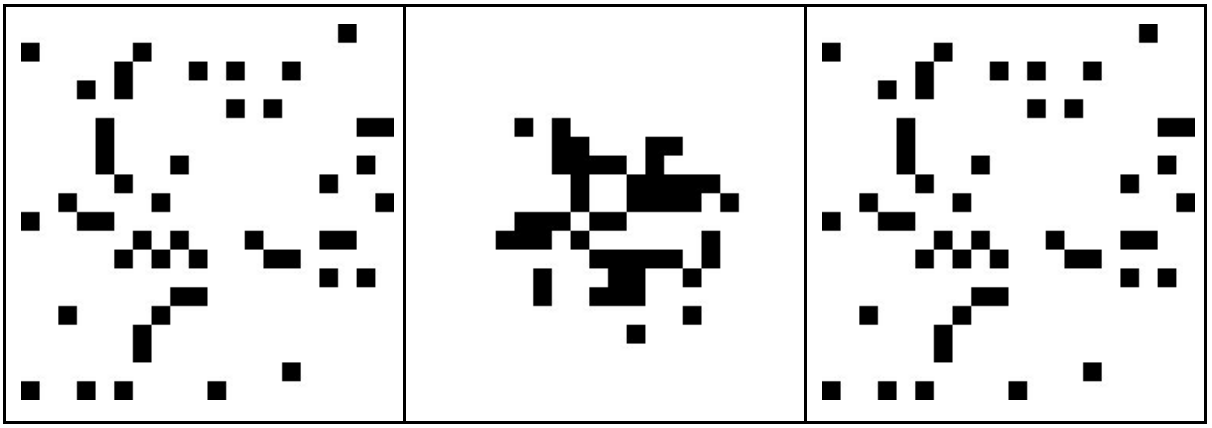
Przyciąganie do środka:

```
posistion_fun = lambda y, x: 100 * (math.sin(y * math.pi + math.pi) + \
                                    math.sin(x * math.pi + math.pi))
between_fun = lambda r: 0
```



Przyciąganie do środka plus punkty się odpychają:

```
posistion_fun = lambda y, x: 500 * (math.sin(y * math.pi + math.pi) +\
                                    math.sin(x * math.pi + math.pi))
between_fun = lambda r: -20 * r
```



zad. 3

Sudoku

Lekka modyfikacja algorytmu wyżarzania:

- *n* iteracji po każedej zmiana temperatury
- dla każej temperatury *m* kolejnych iteracji
- koniec po *n* iteracjach (nie znaleziono) lub po znalezieniu rozwiązania

Do algorytmu przekazywana jest klasa Sudoku.
Zawiera ona obecny stan oraz stan początkowy.
Klasę tworzy się przekazując do niej *numpy.array* reprezentującą sudoku, gdzie w miejscach do uzupełnienia są zera.

Na początku zera zamieniane są na liczby 1-9, aby w każdym wierszu nie było duplikatów.
Energia wyliczana jest na podstawie duplikujących się liczb w każdej kolumnie i kwadracie.

Następna permutacja to zamiana dwóch liczb w losowym wierszu. Gdy jedna z nich jest niezmienna (była na dancyh wejściowych) losowane są inne w tym samym wierszu. Dopiero po 4 próbach zmieniany jest wiersz

Przykładowe wejście, oraz wynik:

```
sudoku = simulated_annealing(
    sudoku,
    get_energy,
    next_perm,
    n=50,
    m=6500,
    T=10,
    cooling_rate=0.1)
```

```
[[5 3 0 0 7 0 0 0 0]
 [6 0 0 1 9 5 0 0 0]
 [0 9 8 0 0 0 0 6 0]
 [8 0 0 0 6 0 0 0 3]
 [4 0 0 8 0 3 0 0 1]
 [7 0 0 0 2 0 0 0 6]
 [0 6 0 0 0 0 2 8 0]
 [0 0 0 4 1 9 0 0 5]
 [0 0 0 0 8 0 0 7 9]]

Solution found after:  27 iterations

[[5 3 4 6 7 8 9 1 2]
 [6 7 2 1 9 5 3 4 8]
 [1 9 8 3 4 2 5 6 7]
 [8 5 9 7 6 1 4 2 3]
 [4 2 6 8 5 3 7 9 1]
 [7 1 3 9 2 4 8 5 6]
 [9 6 1 5 3 7 2 8 4]
 [2 8 7 4 1 9 6 3 5]
 [3 4 5 2 8 6 1 7 9]]

End energy:  0
```