Laboratory 4
Simulated Annealing in Binary Images

Filip Zieliński

03/04/2024 r.

Problem Description

A binary image is an two dimensional array consisting of only two values.
Consider a binary image of a given size. Let $\mathit{Energy} : (x, y) \to \mathbb{R}_+$ be a function that evalates an energy of a single cell.
We define Energy of state to be sum of energy of every cell. Our goal is to minimize the binary image energy without changing the cardinality of zero's and one's in it.
We usually evaluate energy in context of defined *Neibourghood* of a cell. Standard Neighbourhoods examples:

- 4-Neibourghood (up, down, left, right),
- 8-Neibourghood (4-Neibourghood + diagonals),
- Diagonal-Neibourghood (only diagonals),
- 15-Neibourghood (4x4 square around a cell).

Energy can be defined regardless of Neibourghood, but usually combining those two concept gives the most interesting results.

Simualted Annealing

Simulated Annealing is an iterative metaheuristic for finding the function extremum. In this case we start with a randomly generated binary image of given size and by swapping random elements we try to localize the energy function minimum.
Crucial aspect of simulated annealing is a fact, that if we find a state that lower cost function, we always accept it, but sometimes we also accept worse state (especially at begining of a simulation). This technique increases probability of finding global minimum, not local one.
The Algorithm works as follows:

1. Initialize the binary image randomly.
2. Calculate the energy of the initial state.
3. Repeat for given number of iterations:
    (a) Generate a neighboring solution by making a small random change to the current solution.
    (b) Calculate the energy of the new state.
    (c) If the new state has lower energy, accept it as the current state.
    (d) If the new state has higher energy, accept it with a probability based on the energy difference and a cooling schedule.

Energy Functions

Energy functions that I have used and tested are:

1. *Normal Energy* - the Energy increases by one for every neighbour of the same type.
2. *Unity Energy* - the Energy increases by one for every neighbour of diferent type.
3. *Difference Energy* - the Energy of a cell is absolute value of difference of numbers of black and white neighbours.
4. *Center Energy* - white cells lose energy on a verge of an image, black cells lose energy in the center of an image, but there's also a factor of normal energy, meaning cell increases energy if it is only surrounded by it's type.
5. *Parabola Energy* - cell loses energy when it has nearly 0 neighbours of it's type or nearly all of it's neighbours are it's type.

Code and Implementation

Entry Parameteres related to simulation:

1. Neibourghood type
2. Energy type
3. Size of an image
4. Percentage of black cells
5. Number of iterations
6. Initial temperature

7. Coefficient for multiplying temperature

Result:

1. Image of starting position
2. Image of final result
3. Annealing process plot
4. Gif of the simulation

I encourage you too see full Implementation here:
All examples were generated by this code.

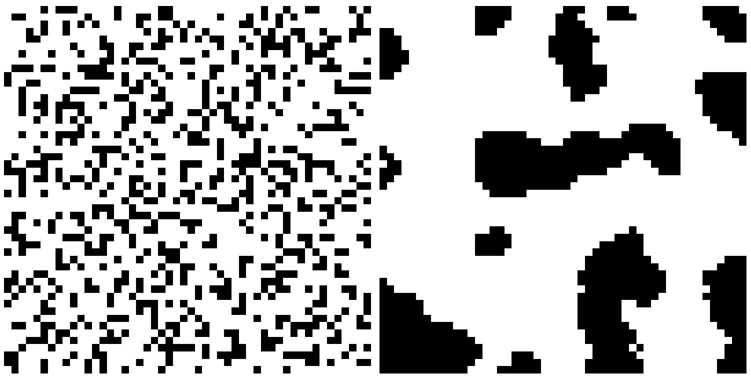
Examples of Results

Example 1

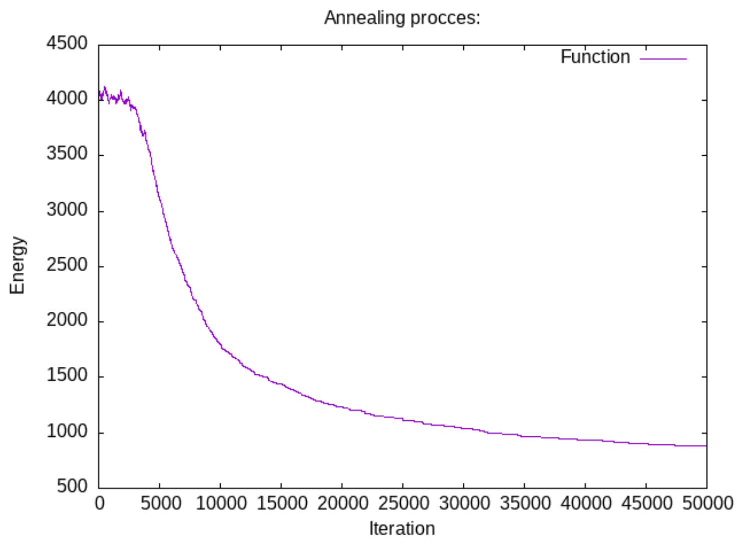
Parameteres:

1. 15-Neibourghood.
2. Unity Energy
3. 30% black cells
4. Image size - 50x50
5. 50 000 iterations
6. Initial temperature - 200
7. Alpha coefficient - 0.999

First iteration vs Final result:



Annealing plot:



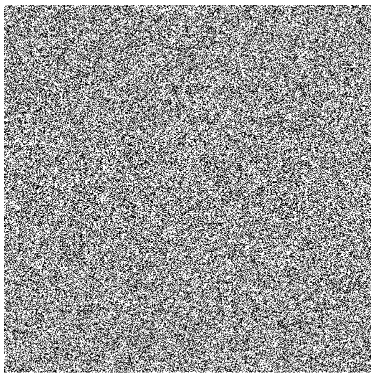
Simulation gif can be found as gifs/example1.gif.

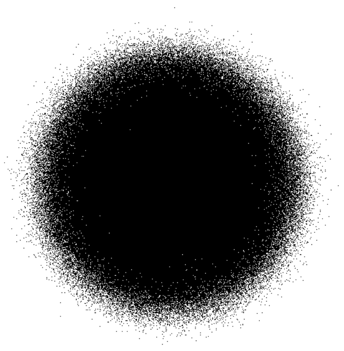
Example 2

Parameteres:

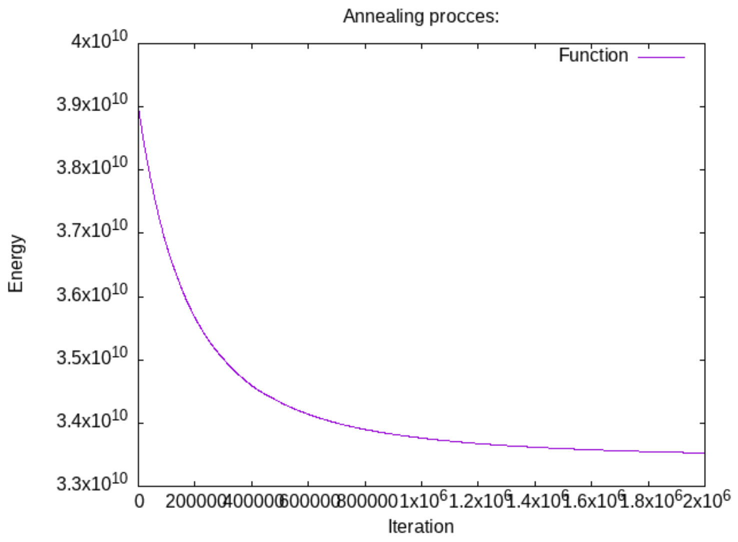
1. 15-Neibourghood.
2. Center Energy
3. 40% black cells
4. Image size - 512x512
5. 200 000 iterations
6. Initial temperature - 1000
7. Alpha coefficient - 0.9999

First iteration vs Final result:





Annealing plot:



Simulation gif can be found as gifs/example2.gif.

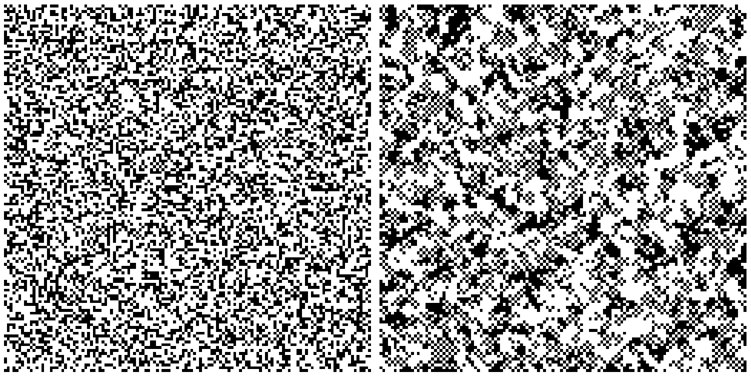
Example 3

Parameteres:

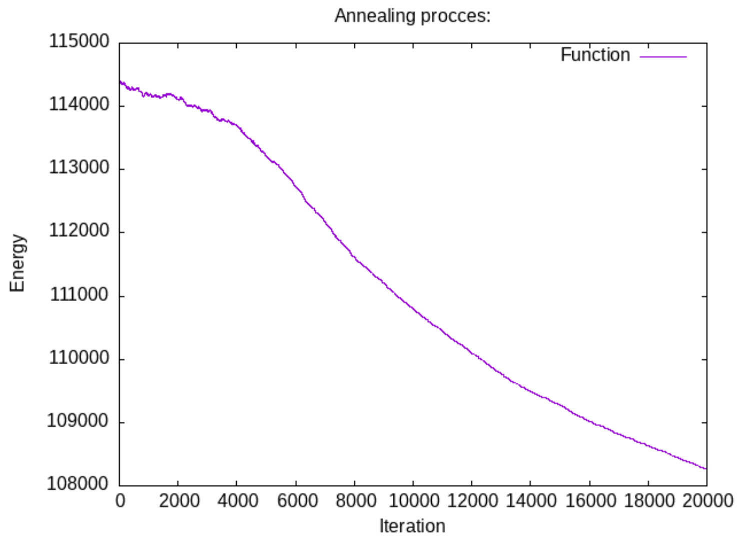
1. Diagonal-Neibourghood.
2. Unity Energy
3. 40% black cells
4. Image size - 128x128
5. 20 000 iterations
6. Initial temperature - 50

7. Alpha coefficient - 0.9994

First iteration vs Final result:



Annealing plot:



Simulation gif can be found as gifs/example3.gif.

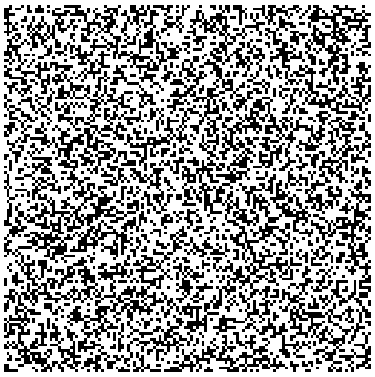
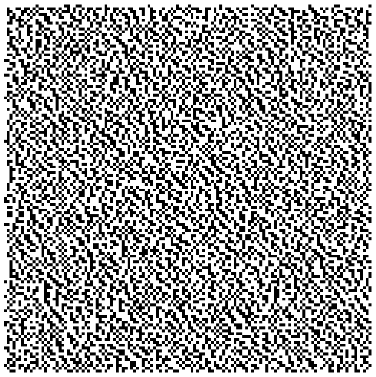
Example 4

Parameteres:

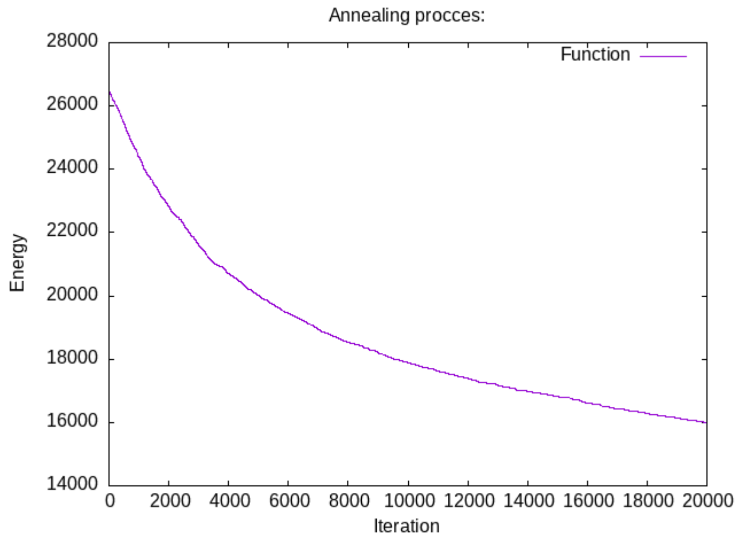
1. 15-Neibourghood.
2. Difference Energy
3. 40% black cells
4. Image size - 128x128

5. 20 000 iterations
6. Initial temperature - 10
7. Alpha coefficient - 0.99

First iteration vs Final result:

Annealing plot:



Simulation gif can be found as gifs/example4.gif.

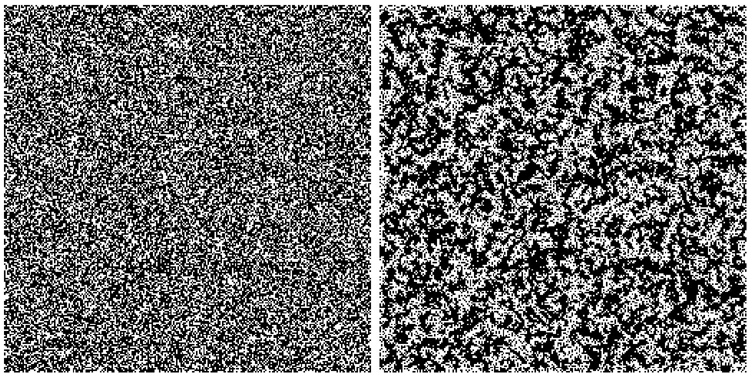
Example 5

Parameteres:

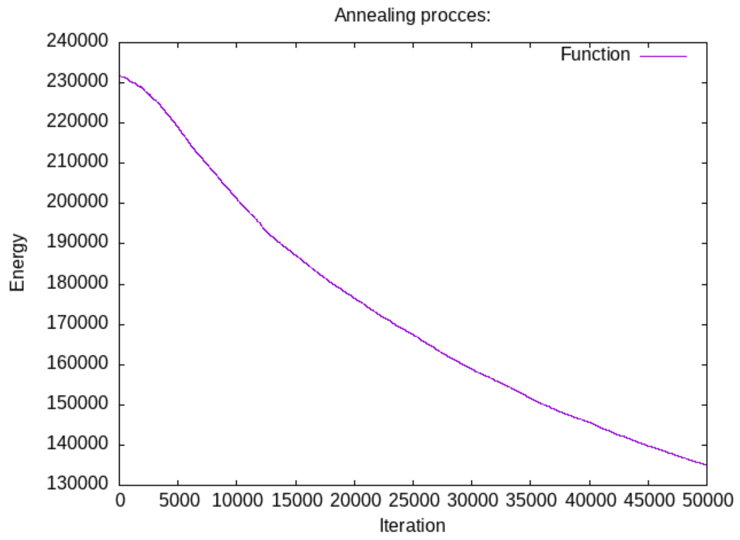
1. 15-Neibourghood.
2. Parabola Energy

3. 60% black cells
4. Image size - 256x256
5. 50 000 iterations
6. Initial temperature - 1000
7. Alpha coefficient - 0.9999

First iteration vs Final result:



Annealing plot:



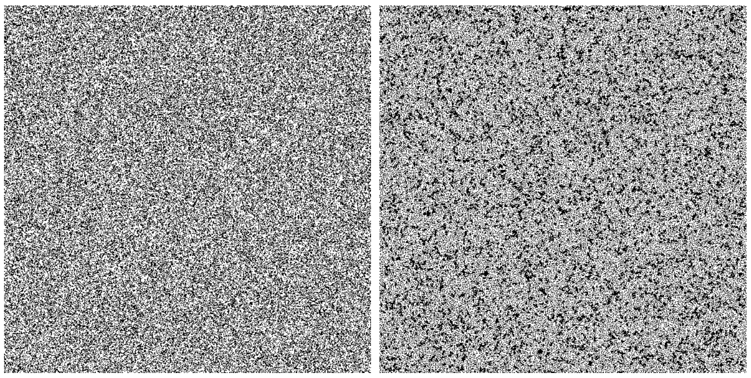
Simulation gif can be found as gifs/example5.gif.

Example 5

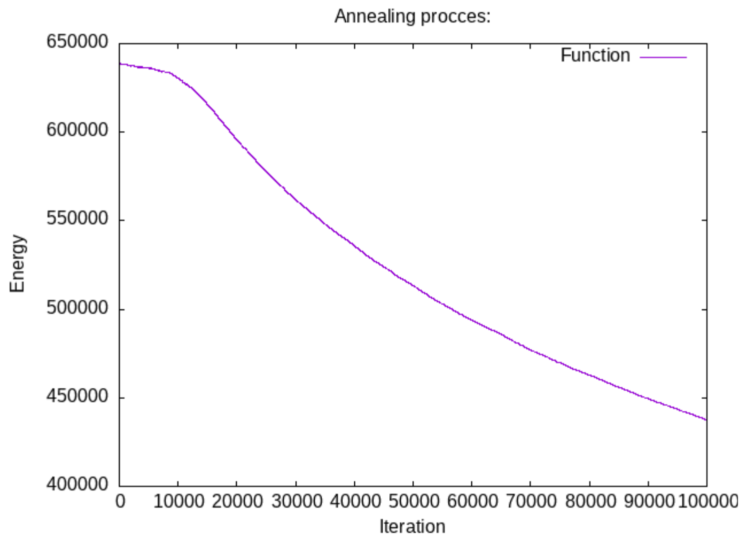
Parameteres:

1. 15-Neibourghood.
2. Parabola Energy
3. 40% black cells
4. Image size - 512x512
5. 100 000 iterations
6. Initial temperature - 500
7. Alpha coefficient - 0.9997

First iteration vs Final result:



Annealing plot:



Simulation gif can be found as gifs/example6.gif.