Charge Configuration Optimization with Simulated Annealing

Paweł Grabiński

Abstract—Optimization is an important problem as well in theoretical computer science as in the applied sciences and industrial solutions. But many optimization problems require either computationally expensive brute-force solutions or heuristics developed for just a single problem. Despite the progress in the "divide and conquer" methods or the convex optimization, a general framework for computationally efficient approximated solutions can be an option worth the pursuit for practical applications. Such methods are often obtained by Monte Carlo approach. And here, the concept of simulated annealing relying on the Metropolis algorithm is introduced. It can be used to minimize a functional of many variables as is shown in an example of minimizing the energy of a system of N identical electric charges.

I. Introduction

Optimization is a subject which has been receiving a lot of research throughout the last decades. Solving the optimization problems with brute-force methods often leads to high computational complexity. Thus, approximated solutions with better time efficiency are often worth investigating. Here, following [1], we present a Monte Carlo method for such approximated solutions in form of the simulated annealing with an example of optimization of a system of electric charges.

In the part II, we describe the general problems and approaches to the optimization with the stress on the dual need of both large and small scale effects. Next, in the part III, general Metropolis algorithm is described. Finally, in the part IV, we describe how to use the Metropolis algorithm for optimization by defining the framework of simulated annealing. And in the part V, our experiment is described in detail. In the final part VI, we discuss the results and insights. The code available for reproduction of the results is available at the repository[1].

II. Optimization

The optimization is a process of rearranging the given system in order to minimize a given objective function which is a functional of the system's state. The objective function is often called the loss function.

In the case of the discrete optimization, the direct approach is to check all possible configurations by brute-force. Without any constraints assumed, it leads to the computational complexity of $N!$. On the other hand, in the case of convex optimization, the potential number of configurations goes to the infinity. Additionally, in the continuous optimization problems, one encounters the problem of distinction among local and global minima. Many of the solutions rely on the heuristic methods. Due to that, they are only applicable to a single problem.

The easiest to implement and simultaneously the most computationally expensive methods are trying all possible permutations for discrete problems and generating random configurations for continuous problems. A straightforward upgrade of such solutions is the iterative approach where one generates modifications to the current state by either random changes or based on the gradient of the objective function and accepts only such modifications that lower the value of the loss function. This approach can be very computationally effective, but oftentimes it gets stuck in a local minimum due to only the small-scale optimization. Such a situation requires multiple simulations for different initial configurations, but with no guarantee that the system will not land in the same or other local minima.

Most of the popular problems were answered with efficient solutions till this day by the "divide and conquer" or some gradient-based optimization methods. The "divide and conquer" approach grants both the small-scale and large-scale effects. In case of the continuous optimization, one can never be sure it reaches the global minima, but the modern methods grant some possible extensions to search through most of the space. Despite that, sometimes it might be worth to look for a more computationally efficient, but approximate solution.

III. Metropolis algorithm

In statistical physics, one wants to describe a system of many particles - more than 10^{23}, in such a way that general macroscopic properties of the system can be concluded. A single macro-state of the system corresponding to some macroscopic features as energy or temperature can be realized as many micro-states. The probability of a given configuration (micro-state) C is defined as [2]:

$$P(C) = Z^{-1} \exp(-\beta H(C)), \tag{1}$$

where the function $H(C)$ is the Hamiltonian describing the energy of the system and $\beta = (k_B T)^{-1}$ is the Boltzmann factor depending on temperature.

The Metropolis algorithm is a method basing on Markov chains used in the statistical mechanics to obtain physically relevant configurations as the generation of such multi-dimensional random variables directly from the distribution is impossible. It can be described by the following steps [3]:

1) generate a random configuration C_i,

[1] http://github.com/PGrabinski/SimulationMethods

2) take an element of the state C_i at random and change it to obtain C_t (e.g. flip a random spin),
3) if $E_t - E_i = \Delta E \leq 0$ or a random uniformly sampled number $u \in [0, 1)$ fulfills $u < \exp(-\Delta E/T)$, then set $C_{i+1} = C_t$ accepting the change, otherwise set $C_{i+1} = C_i$,
4) repeat $2-3$ steps N times, where N is the number of elements of the system,

This describes a single Monte Carlo step. At first, n_{therm} of steps are required for thermalization of the system without yielding any reliable knowledge on the system. After the thermalization phase, every Monte Carlo step corresponds to a distinct configuration which can be taken into the considered statistical ensemble.

IV. Simulated annealing

In simulated annealing, we use the analogy to cooling the melted alloys into solids. If one cools liquid metal too quickly, it solidifies into an amorphous structure without any large-scale structure. In case of slow cooling, one can achieve symmetric crystal which shows large-scale organization. Using this analogy requires the introduction of an additional order-related parameter corresponding to the temperature T in the physical picture.

Using this idea, we can minimize any functional of the system state $E(C)$ by the Metropolis algorithm. This approach shows the dual nature similar to the "divide and conquer" methods acting on low-scale due to minimization of the functional $E(C)$ and large-scale duo to the probabilistic acceptances depending on the parameter T. The clue is the slow cooling meaning using a given number n_{MCS} of Monte Carlo Steps per given temperature T_i and slowly lowering the temperature as $T_{i+1} = s \cdot T_i$ where the cooling ratio s is a number close to 1, e.g. $s = 0.9$ or $s = 0.95$.

The challenge in this approach is developing a proper configuration modification generator. For example, in case of the travelling salesman, one could could consider either reversing two neighboring points $(x_i, x_{i+1}) \to (x_{i+1}, x_i)$ or sliding two points along the travel track $(x_i, x_{i+1}) \to (x_{i\pm1}, x_{i\pm1+1})$. These two transformations can bring any given permutation $p(x_1, \ldots, x_N)$, but the number of steps (thermalization) needed for that oftentimes is much higher than the assumed n_{MCS}. This requires at least investigation of the behavior of the system in order to check the thermalization pace, what usually ends in rising the thermalization parameter n_{MCS}. Alternatively, one can introduce some more greedy modification generator either by more significant changes to the configurations or by some heuristic approach in the case where some assumptions can be made.

A more sophisticated approach was proposed in [1]. One can compute an object analogous to the specific heat basing on the variance of the functional being optimized.

$$C_V = \frac{Var(E_T)}{NT^2}$$

Then make the number of steps a function of the specific heat $n_{MCS} = n_{MCS}(C_V)$. This should make the ratio between small-scale and large-scale interactions more effective resulting in a higher chance for finding a better -possibly global- minimum.

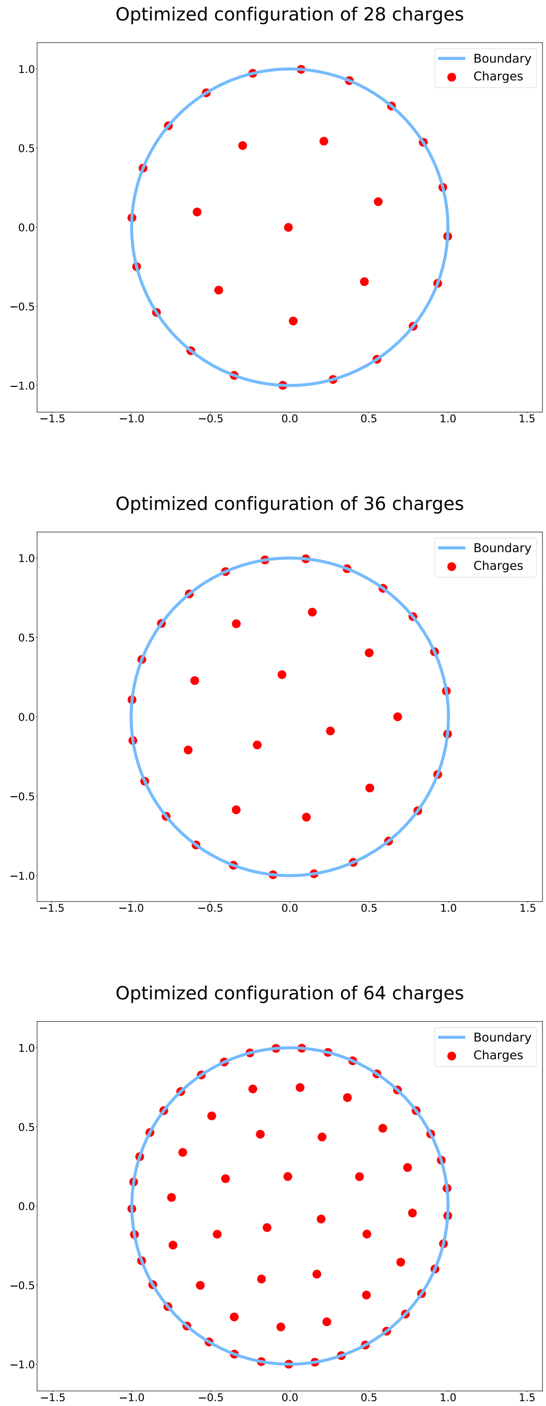


Figure 1. Optimized configurations for the number of charges $N > 10$ show emerging structures in the interior of the system space. These structures form new orbits analogous to the charges on the boundary of the system.

Concluding the simulated annealing as an algorithm:

1) generate initial (random) configuration and set initial temperature $T_i = T_{init}$,
2) perform n_{MCS} steps of the Metropolis algorithm with a given functional instead of the Hamiltonian ,
3) cool the system by setting $T_{i+1} = s \cdot T_i$,
4) repeat the steps 2-3 as long as $T_i > T_{final}$.

V. Methods

In our experiment, we use the simulated annealing to minimize the energy of a system of N identical charges in a space bounded by a unit circle. The energy of the system is physically given as:

$$E(C) = \sum_{i,j=1,\ i\neq j}^{N} k\frac{q_i q_j}{r_{ij}}.$$

As we know that the charges are identical and we can set them as $\forall_i q_i = \sqrt{\frac{1}{k}}$. So we get:

$$E(C) = \sum_{i,j=1,\ i\neq j}^{N} \frac{1}{r_{ij}}.$$

Where the r_{ij} is the distance between charges i and j.

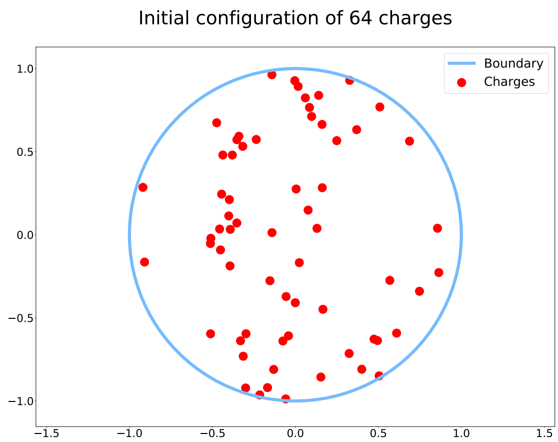


Figure 2. Example of initial random configuration of the system generated in order to obey the boundary limit.

In order to preserve maximal generalization of the solution, we constructed the configuration modification generator as addition of a random uniformly sampled vector $\vec{u} \in (-\frac{\epsilon}{2}, \frac{\epsilon}{2})^2$, where epsilon is a small parameter of order $\epsilon \simeq 10^{-2}$. In case of sampling such a vector $\vec{u}$ that violates the boundary, it is generated repeatedly until an acceptable translation is found.

The experiment was conducted on the following parameter space ($k_B = 1$):

- Initial temperature $T_{init} \in \{1.0, \quad 0.5, \quad 0.1\}$,
- Final temperature $T_{final} = 10^{-5}$,
- Change parameter $\epsilon \in (0.1, 0.5)$,
- Number of charges $N \in \{10, 11, \ldots, 39, 40, 64\}$,
- Cooling ratio $s \in \{0.5, \quad 0.8, \quad 0.9, \quad 0.95\}$,
- Monte Carlo steps $n_{MCS} \in \{10, \quad 15, \quad 25, \quad 50\}$.

VI. Results and Discussion

It can be seen that the procedure leads to the creation of configurations with symmetric ordered structures as in the figure 1. Such configurations correspond to some energetic minima. The emergent structures are additional orbits. The charges tend to take place not only on the boundary but also inside the system on regularly separated rings.

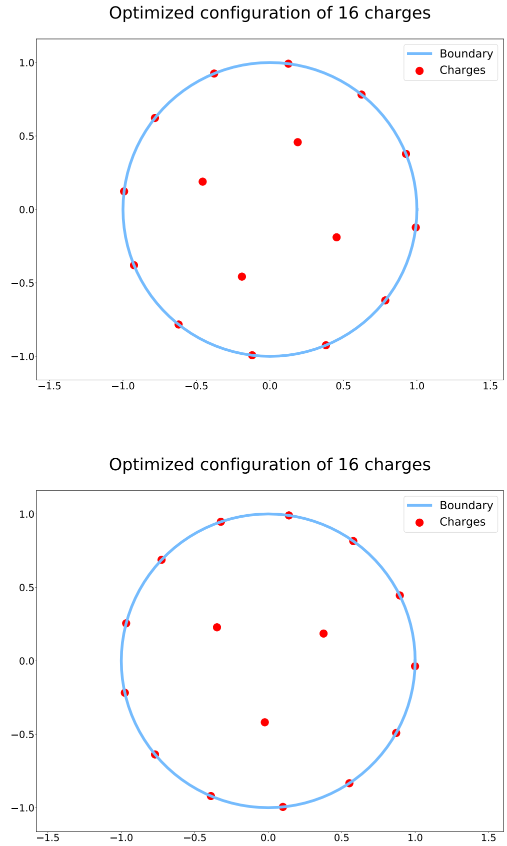
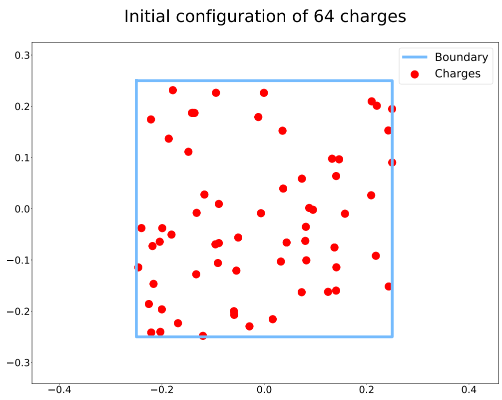


Figure 3. A configuration of 16 charges with 4 charges (top) in the middle of the space corresponding to an alternative local minimum of the energy of 118.94 in contrast to the configuration with 3 charges in the middle (bot) with the energy of 117.46.

The minima can be either local or global - there is no other way to verify it other than finding a better minimum as it was shown in the figure 3.

The main problem of the research when it comes to checking different combinations of parameters is that the system generated in a totally random manner would require a thermalization similar to the one performed in the statistical physics - the phase when observables are not measured as they are not representing the demanded distribution. In this case, the source of the problem is that the random configuration requires so much optimization that combination of low Monte Carlo steps n_{MCS} and cooling ratio $s < 0.9$ yield results without optimization even in the small-scale sense. In case of such parameters initial optimization even basing on iterative improvement alone without the probabilistic acceptance could be valuable.



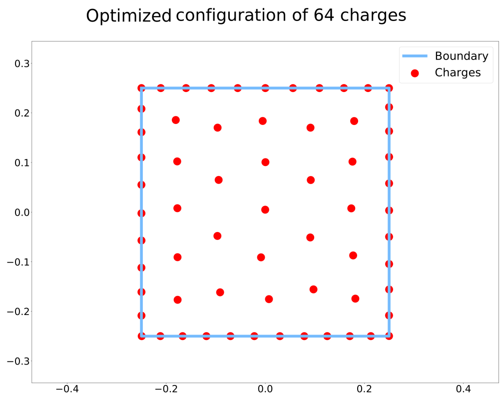


Figure 4. The optimization procedure is easy to generalize to different geometries thanks to the general construction of the modification generation.

In the implementation of this method, we have made a trade-off in order to retain as much generalization as possible what was shown in the figure 4. The generalization was traded for computational efficiency. The optimization is definitely not fast and there is a chance that other methods could achieve similar results quicker or with greater precision.

VII. Conclusions

The simulated annealing method is a method definitely easy to implement in the most general way which can solve many problems. Despite being definitely more practical than simple checking all possible configurations or randomly sampling them, this approach can be very computationally expensive. It can be mitigated by introducing more greedy heuristic modification strategies or binding the number n_{MCS} to the behavior of the system. But

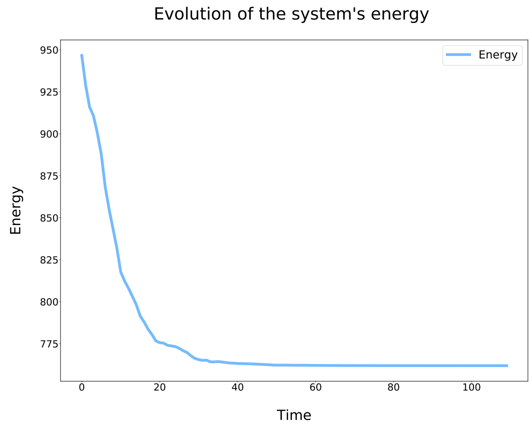


Figure 5. Energy being minimized during the optimization procedure.

this investigation is left for future research. In the case of developing solution to a very repetitive optimization, it could be even reasonable to optimize the avarage time of optimization by chaning the modification generator with evolutionary algorithms, simulated annealing or other methods.

References

[1] Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. "Optimization by simulated annealing." science 220, no. 4598 (1983): 671-680.

[2] Binder, Kurt, Dieter Heermann, Lyle Roelofs, A. John Mallinckrodt, and Susan McKay. "Monte Carlo simulation in statistical physics." Computers in Physics 7, no. 2 (1993): 156-157.

[3] Oleksy, Czesław, "Lecture Notes on Simulation Methods", University of Wrocław 2019.