The Thomson problem

Elvira Zainulina
Optimization Class Project. MIPT

Introduction

The objective of the Thomson problem is to determine the minimum electrostatic potential energy configuration of N electrons constrained to the surface of a unit sphere that repel each other with a force given by Coulomb's law.

The physical system embodied by the Thomson problem is a special case of one of eighteen unsolved mathematics problems proposed by the mathematician Steve Smale "Distribution of points on the 2-sphere". However, the Thomson problem was analytically solved in several cases: $N \in 2, 3, 4, 5, 6, 12$.

The similar problem can be posed for spaces of bigger dimensions or spaces with non-euclidean geometry. In this field analytical solutions exist for $8D$ $N = 240$, $24D$ $N = 196560$, arbitrary dimension D and $N = D + 1, 2D$.

Optimization problem

The solution of each N-electron problem is obtained when the N-electron configuration constrained to the surface of a sphere of unit radius, $r = 1$, yields a global electrostatic potential energy minimum, $U(N)$.

The electrostatic interaction energy occurring between each pair of electrons of equal charges ($e_i = e_j = e$, with e the elementary charge of an electron) is given by Coulomb's Law,

$$U_{ij} = k_e \frac{e_i e_j}{r_{ij}}.$$

Consequently, the optimization problem is:

minimize $\sum_{i \neq j}^{N} \frac{1}{\|x_i - x_j\|}$

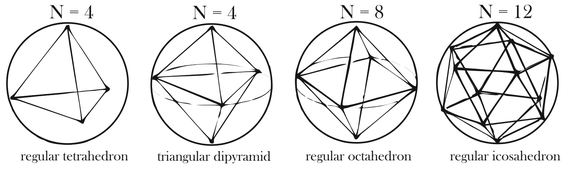
subject to $x_i \in R^D$ $\|x_i\| = 1$ $1 \leq i \leq N$

This conditional optimization problem is not convex, as although the objective function is convex, the set is not. That causes difficulties in finding global minimums. With an increase in the number of charges, the number of metastable states, i.e. local minimums, increases exponentially.

Existing solutions

$2D$ Regular polygons.

$3D$ Minimum energy configurations have been rigorously identified in only a handful of cases: $N = 2$ - antipodal points; $N = 3$ - an equilateral triangle;



Proof of these solutions are in the article "Extreme locations of points on the sphere" (N. N. Andreev, V. A. Yudin), except for $N = 5$. For $N = 5$, a mathematically rigorous computer-aided solution was reported in 2010.

For other N (up to 972) solutions have been got numerically[3].

N-D It was proven that the arrangement of 240 charges in case of $8D$ forms the construction of Korkin-Zolotarev[1].

In case of $24D$ 196560 charges constitute the Leech lattice[2].

In case of the arbitrary dimension D, $D + 1$ charges assemble simplex and $2 \cdot D$ charges turn out to be vertices of an octahedron[2].

Approach to the solution

3 methods were applied to solve this problem. The first one is gradient descent that includes all charges. The second one is gradient descent that includes one charge, often the worst one. The third one is stochastic gradient descent.

The aim was to compare first these methods and find the algorithm that solves the problem better and faster than others.

Algorithm

w is a charge vector.

Algorithm GD:

CHECK(w) ▷ If some charges locate too close to each other, ▷ randomly generate a new arrangement
$err = 1.0$
while $err > 10^{-50}$ **do**
  $w_{new} = gr\_step(w, \eta, U(w), dU(w))$
  $err = |U(w) - U(w_{new})|$
  $w = w_{new}$
return $traj\_w$, $U(w)$

Algorithm DMK:

$err = 1.0$
while not $err \leq 10^{-50}$ 5 times in a row **do**
  $cand = worst\_elem(w)$ ▷ the one that experiences ▷ the greatest force from the rest charges
  $w_{new} = w$
  $w_{new}[cand] = gr\_step(w[cand], \eta, U_{elem}(w[cand]), F(w[cand]))$
  if $U(w_{new}) \geq U(w)$ **then**
    $cand = random()$
    $w_{new}[cand] = gr\_step(w[cand], \eta, U_{elem}(w[cand]), F(w[cand]))$
  $err = |U(w) - U(w_{new})|$
  $w = w_1$
return $traj\_w$, $U(w)$

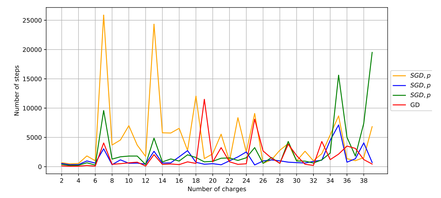
$Gr\_step(x, func, dfunc)$:

$\alpha = \eta$
$err = 1.0$
while $err \geq 10^{-14}$ **do**
  $x_{new} = \text{NORMALIZED}(x - \alpha \cdot dfunc)$
  $err = func(x_{new}) - func(x)$
  $\alpha = \alpha / 2$
return x

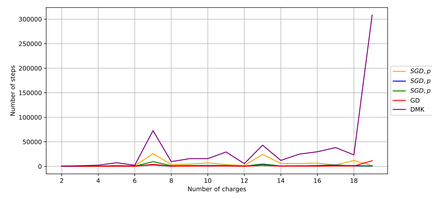
The algorithm SGD has the same structure as the algorithm GD and the stop criterion as the algorithm DMK. There are only some slight differs in gr_step function. The input for it is new functions U_sigma and dU_sigma, which calculate U and dU only for randomly chosen part of all charges.

Comparison of the algorithms

At first, the number of steps required for convergence of SGD was compared for 3 values of the probability parameter: $0.1, 0.5, 0.9$. As it can be seen from the following graph, the probability parameter 0.5 gives relatively better convergence rate for all numbers of charges. This graph also shows the number of steps for GD. GD requires less number of steps than SGD, except for a few points. Therefore, it is more preferable.

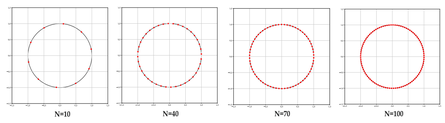


The second graph shows that DMK is the least effective algorithm.

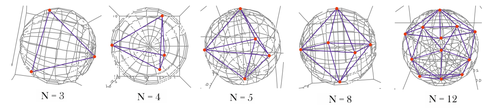


Comparison with proven results

$2D$ As expected, all obtained arrangements of charges form regular polygons.



$3D$ Arrangements of charges obtained by GD completely coincide with analytical solutions:



For other numbers of charges up to 50 obtained energies correspond to existing numerical results[3]. This comparison can be found in the supplementary materials[4].

$N - D$ The algorithms can be applied to solving the problem in different dimensions. However, obtained results can't be evaluated properly.

Conclusion

Proposed algorithms for solving the Thomson problem gave results corresponding to existing and proven ones. Therefore, its results for other dimensions are likely to be the truth. For numerical solutions, GD and SGD with $p = 0.5$ are more preferable, as they have to make fewer steps to converge than others. Improvements for these algorithms can be made by changing the method of choosing the gradient step. However, the main difficulty in this problem consists in the existence of many local minimums. Thus, algorithms should be developed to be able to account for it.

References

[1] A. V. Kolushov, V. A. Yudin. On the construction of Korkin-Zolotarev.
[2] N. N. Andreev. Arrangement of points on a sphere with minimal energy.
[3] The Cambridge Cluster Database
[4] Google Colab - IPython Notebook with algorithms and some demonstrations. Some results.