# PHY604 Lecture 23

November 28, 2023

# Review: Steps of Markov chain Monte Carlo:

- 1. Choose random starting state
- 2. Choose a move uniformly at random from set of moves
- 3. Calculate the acceptance probability
- 4. Accept or reject the move
- 5. Measure *X* in current state, add to sum
- 6. Go back to step 2

# Review: Ideal gas

- Consider the quantum states of a particle or atom of mass *m* in cubic box of length *L*
- Energy of one particle given by:

Quantum numbers from 1 to infinity.

$$E(n_x, n_y, n_z) = \frac{\pi^2 \hbar^2}{2mL^2}(n_x^2 + n_y^2 + n_z^2)$$

- Ideal gas: no interactions between particles
  - Energy is sum of individual particles:

$$E = \sum_{i=1}^{N} E(n_x^{(i)}, n_y^{(i)}, n_z^{(i)})$$

# Review: Move set for ideal gas

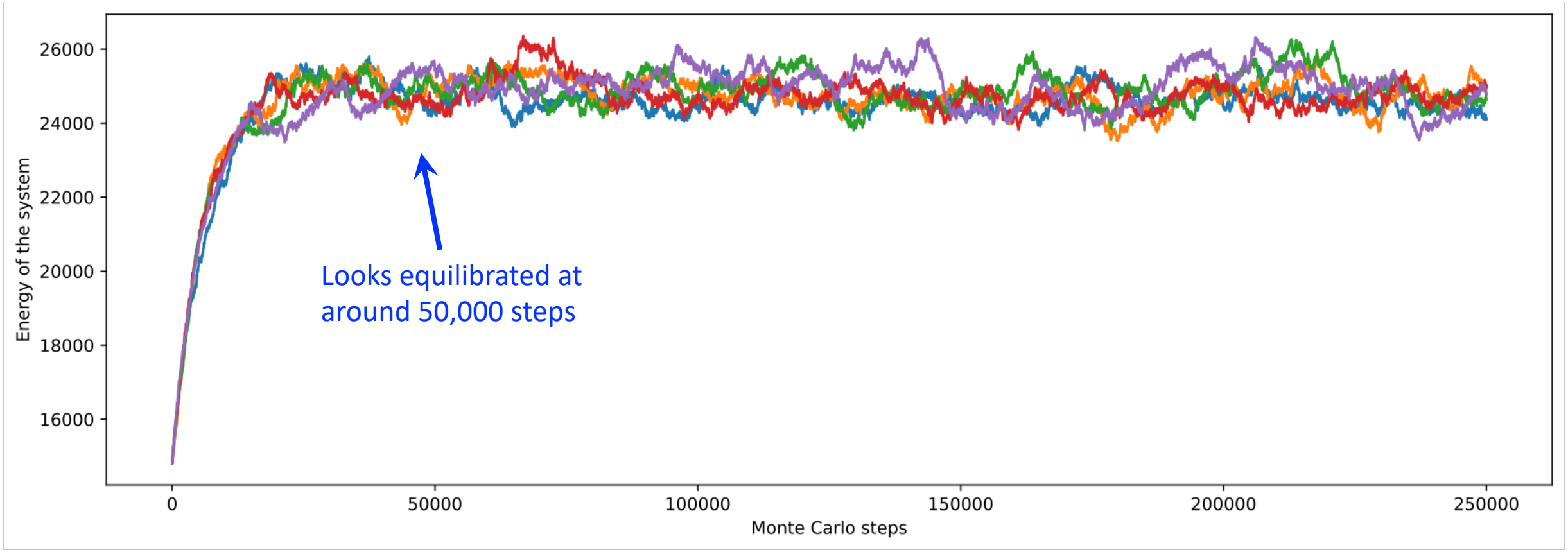
- Choose set of all moves of a single atom to one of the six "neighboring" states where $n_x$, $n_y$, or $n_z$ differ by +/- 1
- Each Monte Carlo step, choose a random particle, chose a quantum number, change it by +/- 1
- Change in total energy just the change for single particle since there are no interactions
  - E.g., increase or decrease $n_x$ of atom *i* by one:

$$\Delta E = \frac{\pi^2\hbar^2}{2mL^2}[(n_x \pm 1)^2 + n_y^2 + n_z^2] - \frac{\pi^2\hbar^2}{2mL^2}(n_x^2 + n_y^2 + n_z^2)$$

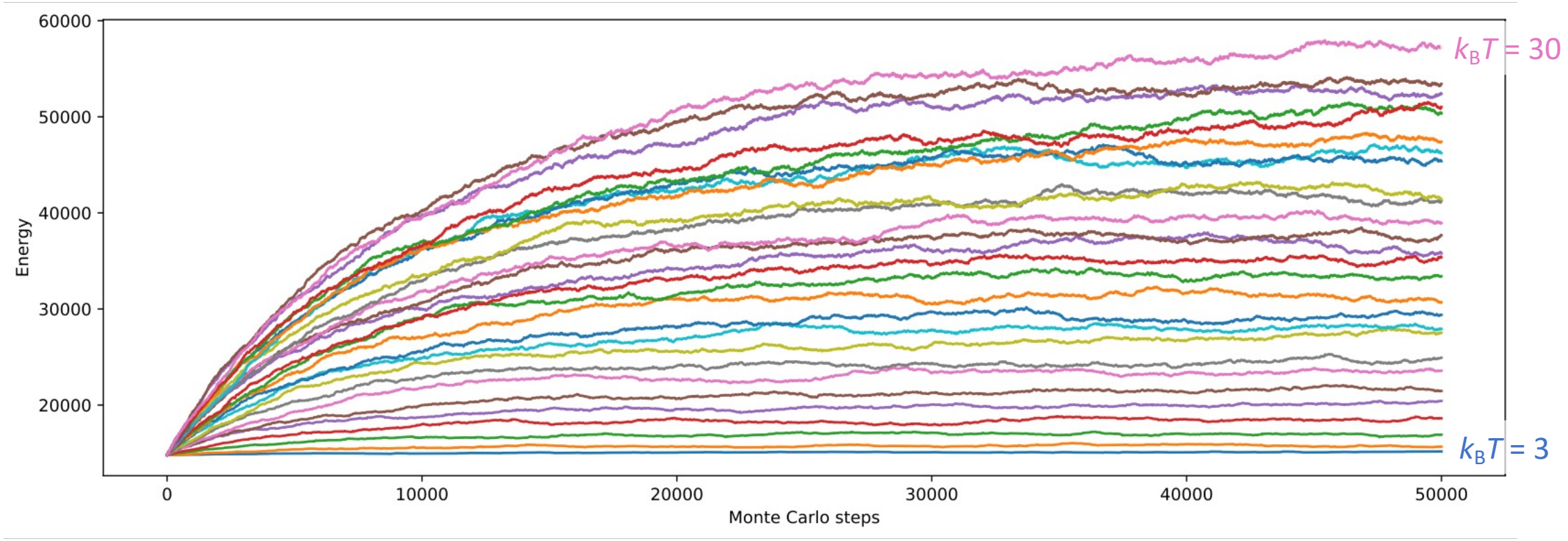
$$= \frac{\pi^2\hbar^2}{2mL^2}[(n_x \pm 1)^2 - n_x^2] = \frac{\pi^2\hbar^2}{2mL^2}(\pm 2n_x + 1)$$

- Note: Reject moves that try to make $n < 1$

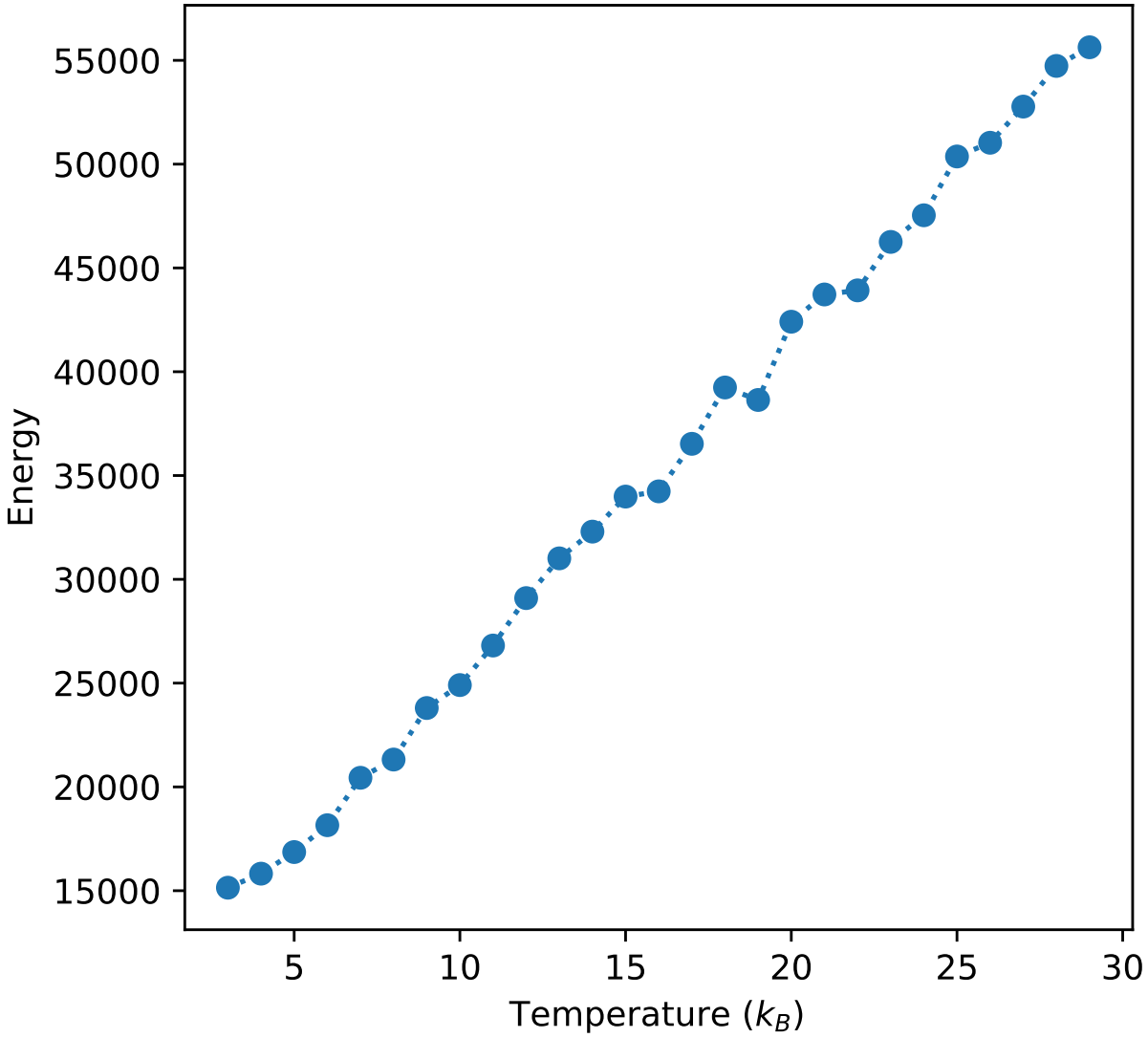
# Review: Monte Carlo simulation of ideal gas



# Review: Monte Carlo simulation of ideal gas: Dependence on T



# Review: Monte Carlo simulation of ideal gas: E vs. T



# Review: The Ising model

- The Ising model is a classic model in statistical physics for describing magnetic systems

- Describe a magnetic material as classical spins on a lattice
  - Spins can only point up or down
  - Energy is given by:

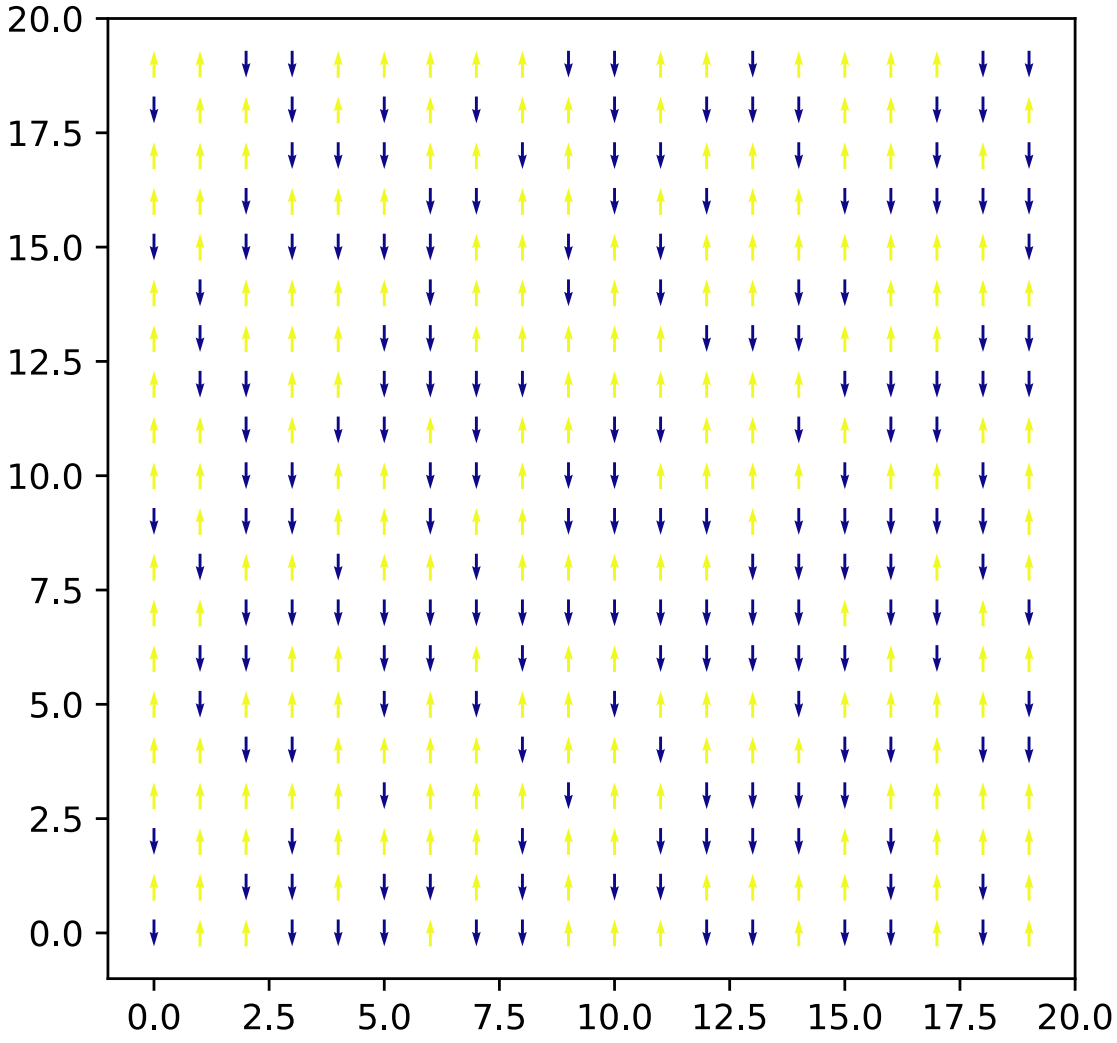
$$E = -J \sum_{\langle i,j \rangle} s_i s_j$$

  - Where $\langle i, j \rangle$ indicate neighboring spins
  - *J* is the interaction strength
    - If J > 0 aligned spins are preferred
    - If J < 0 antialigned spins are preferred

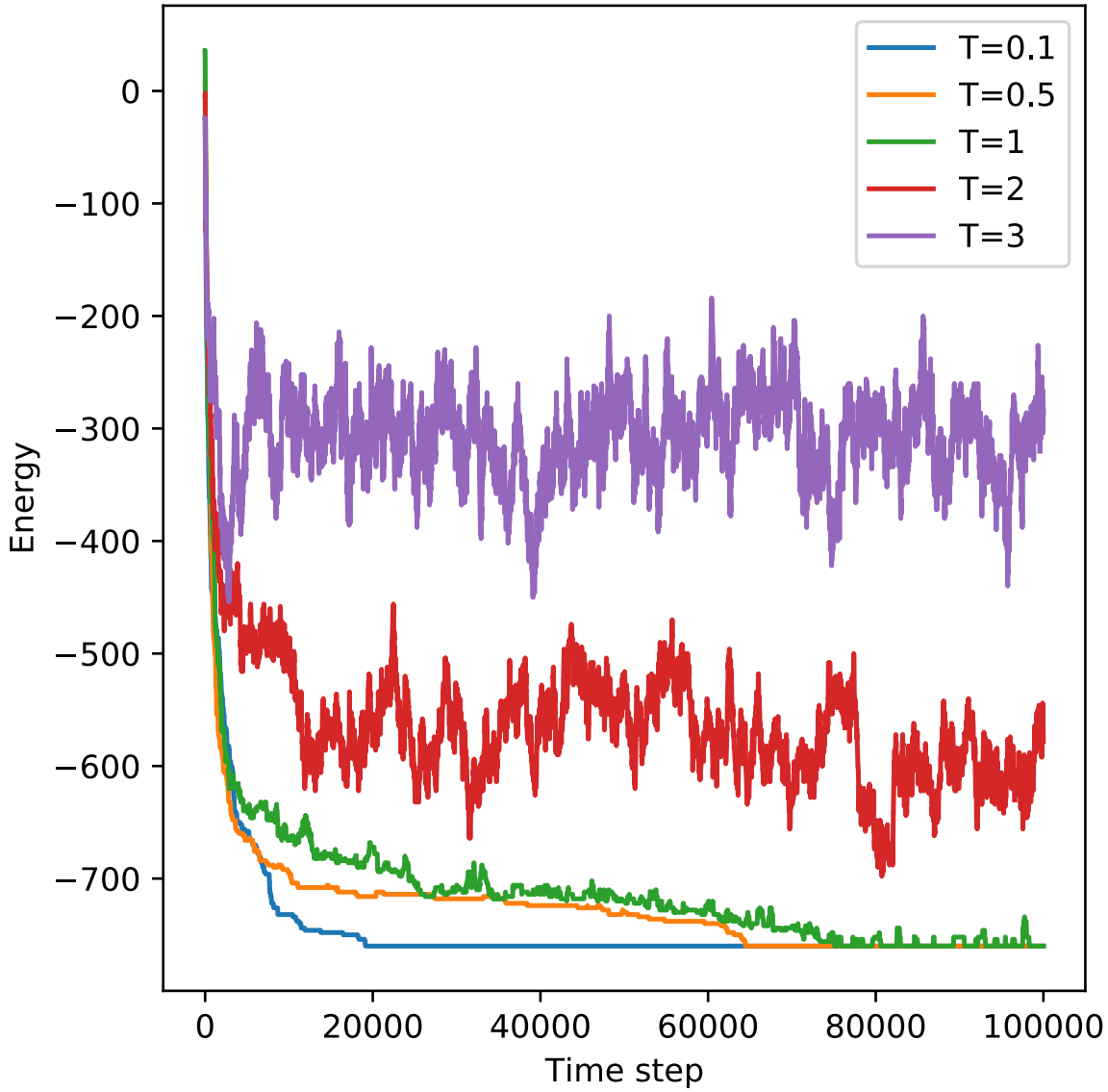
# Review: Monte Carlo simulation: Ising model on square lattice

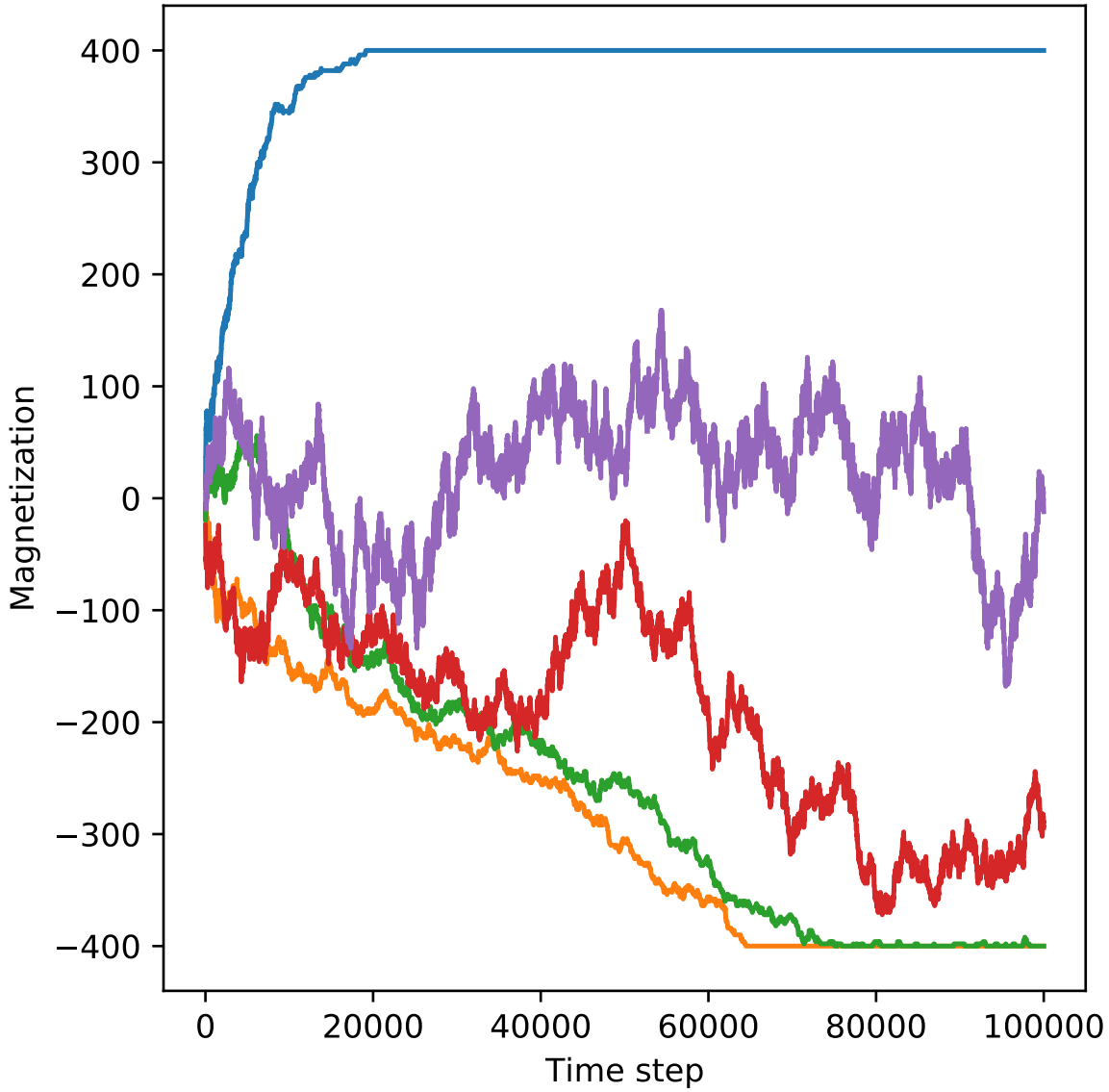
- 20 x 20 square lattice of spins
- Initialized randomly
- Move set: Flip a random spin
  - If it lowers the energy, accept
  - If it raises the energy, accept with Boltzmann probability
- Can also monitor magnetization

$$M = \sum_i s_i$$

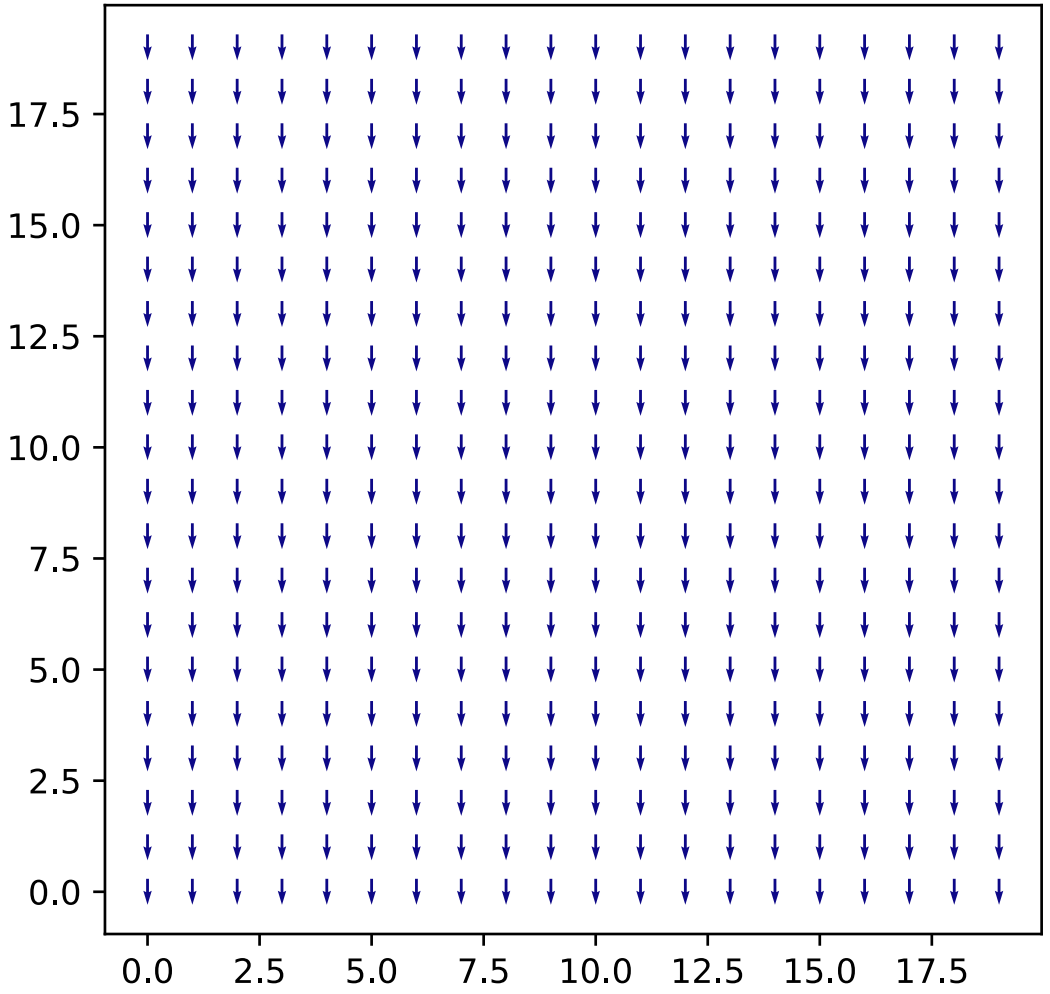


# Review: Ising model on square lattice versus T

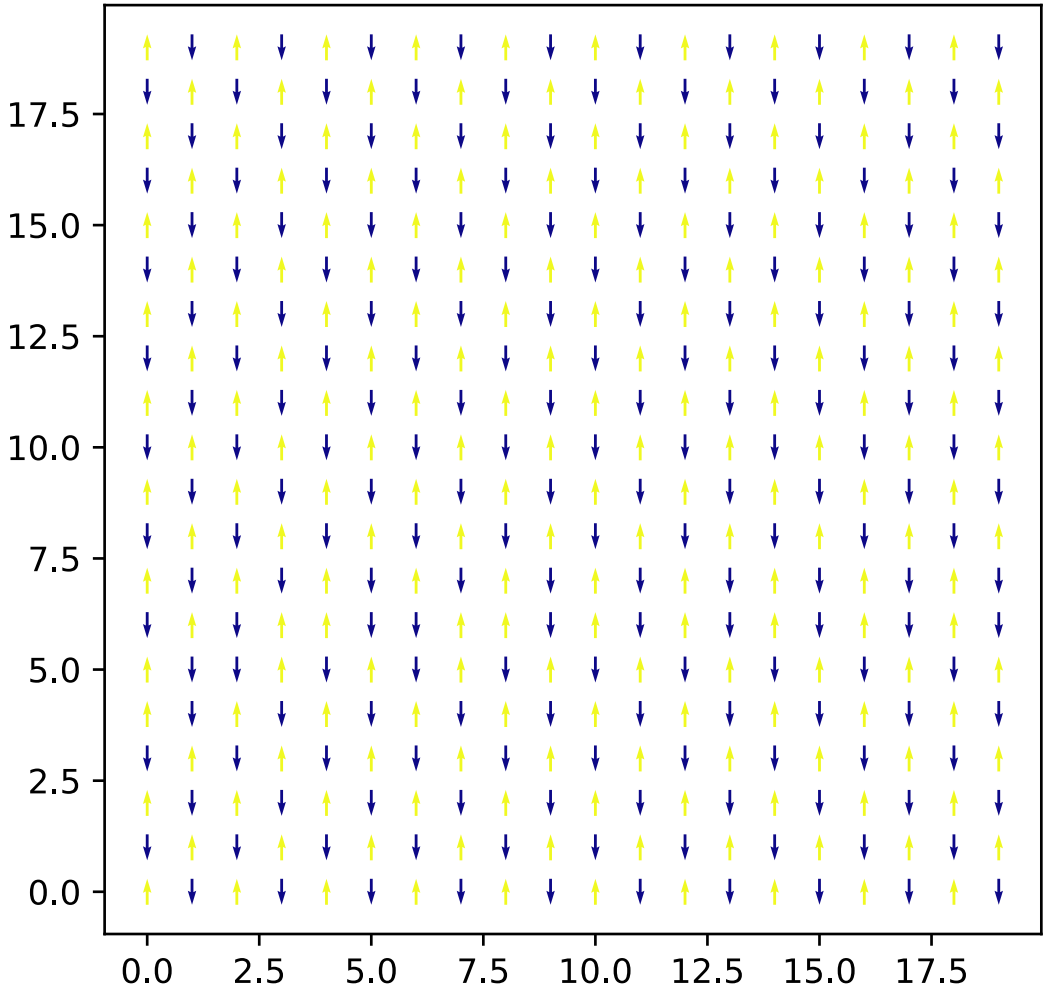


# Ferromagnetic for J > 0,




# Antiferromagnetic for J < 0



# Today's lecture:
# Simulated annealing, genetic algorithms

- Simulated Annealing
  - Travelling salesman problem

- Genetic algorithms

# Simulated annealing (Newman Sec. 10.4)

- Monte Carlo methods can also be used for numerical optimization
- Optimization methods discussed so far only give local minima
- Global optimization problems are very challenging
- Simulated annealing borrows ideas from statistical physics/materials science to tackle this problem
  - Annealing: Heat treatment of materials to remove defects by allowing atoms to move to their equilibrium structure

# Statistical mechanics for optimization

- Recall the Boltzmann probability:

$$P(E_i) = \frac{e^{-\beta E_i}}{Z}, \qquad Z = \sum_i e^{-\beta E_i}$$

- Assume we have single, unique ground state
- Choose energy scale so that the ground state configuration is 0 energy
- If we cool the system to $T = 0$, then the probability distribution is:

$$P(E_i) = \begin{cases} 1 & \text{if } E_i = 0 \\ 0 & \text{if } E_i > 0 \end{cases}$$

- By cooling the system, we can find the ground state

# Statistical mechanics for optimization

- We can use the same strategy (cooling the system) for finding the minimum of a function
  - Take the value of the function to be the “energy”
  - Take the values of independent variables to define a state of the system

- But how can we avoid getting trapped in a local minima?
  - Energy of all nearby states are higher in energy, will not accept moves for low *T*

- Solution: “Anneal” by cooling slowly so system can find its way to the global minimum
  - Guaranteed to converge to global minimum if we cool slowly enough (often not possible)

# Simulated annealing approach

- Choose $k_B T$ to be significantly greater than the typical energy change from a singe Monte Carlo move
  - Then:

$$\beta(E_j - E_i) \ll 1 \quad \Longrightarrow \quad P_a \simeq 1$$

  - Most moves accepted, state of the system rapidly randomized

- Make a cooling "schedule," e.g.:

$$T = T_0 e^{-t/\tau}$$

- Choice of $\tau$ require some trial and error, slower cooling is more likely to find ground state, but simulation takes longer

# Example: Travelling salesman problem

- Find the shortest route that visits a given set of locations on a map
- One of the most famous optimization problems (NP hard)
- We will assume the salesman can travel between the *N* points on the map in straight lines (i.e., the world is flat)
  - *N* cities are chosen at random in a 2D square of unit length
- Want to minimize total distance travelled over the tour:

$$D = \sum_{i=0}^{N-1} |\mathbf{r}_{i+1} - \mathbf{r}_i|$$

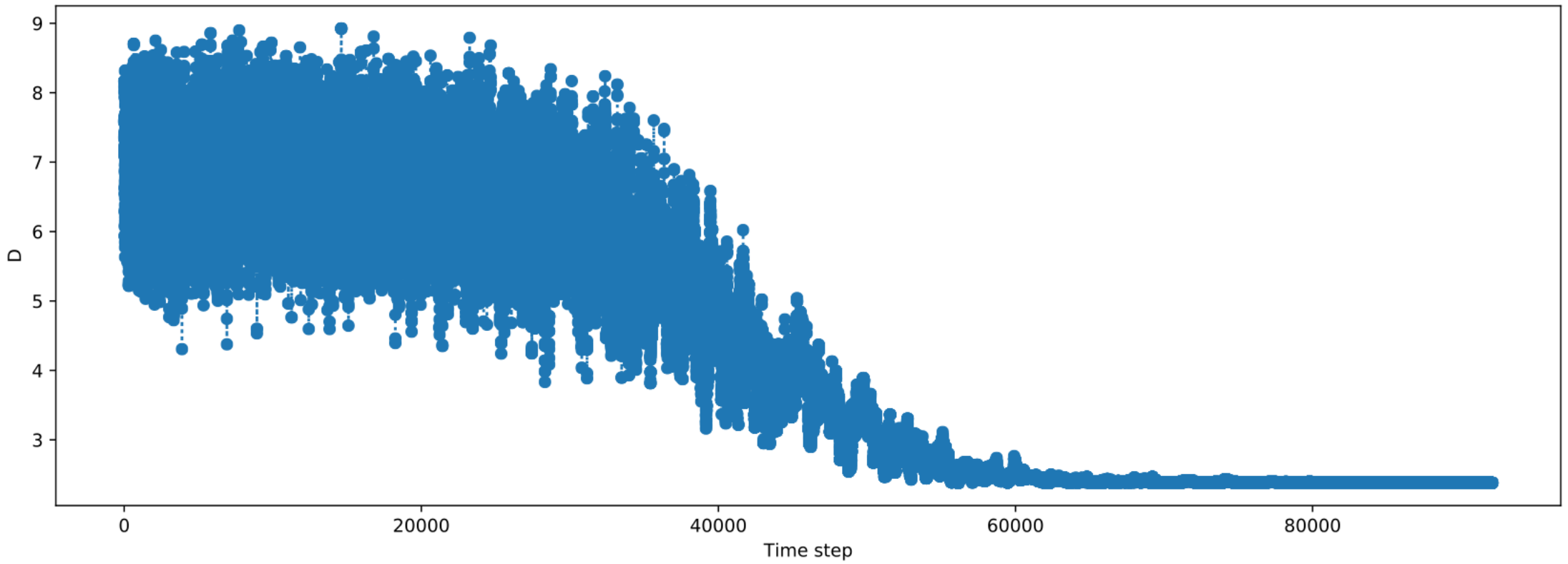
# Markov chain Monte Carlo for traveling salesman

$$D = \sum_{i=0}^{N-1} |\mathbf{r}_{i+1} - \mathbf{r}_i|$$

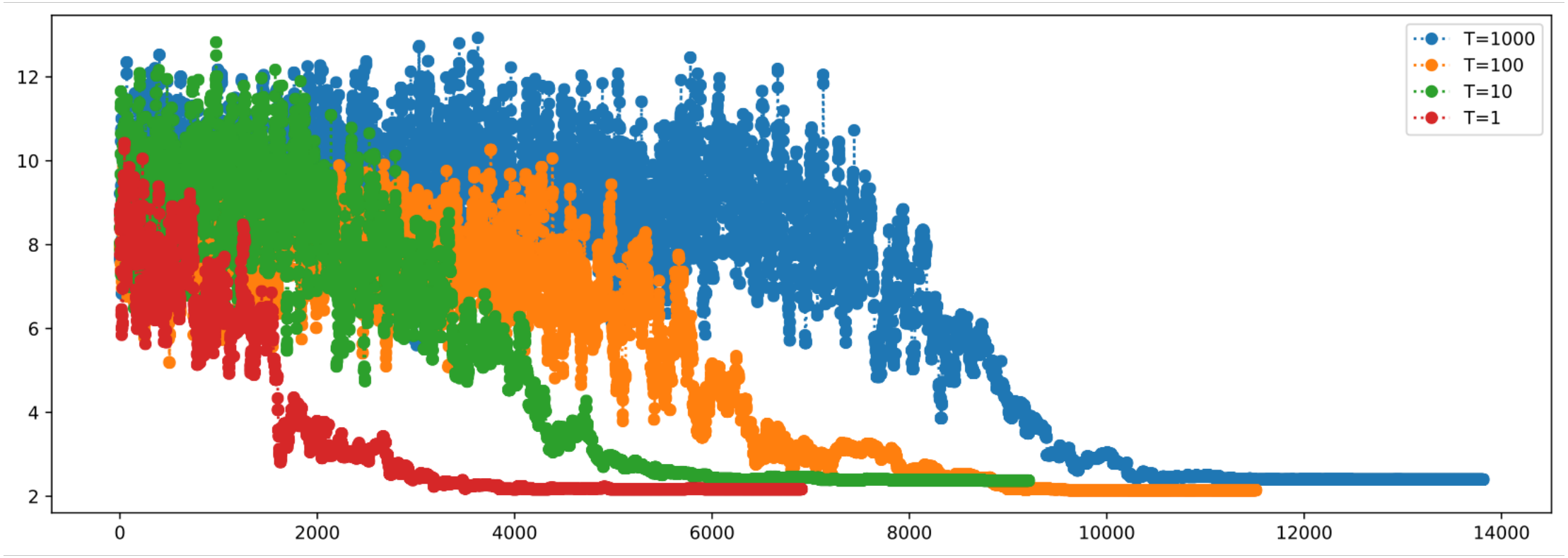
- Minimize *D* over set of all possible tours

- First set up an initial tour
- Then choose from set of moves: Swap pairs of cities
  - Accept if swap shortens the tour
  - If it lengthens the tour, accept with Boltzmann probability, energy replaced by distance D

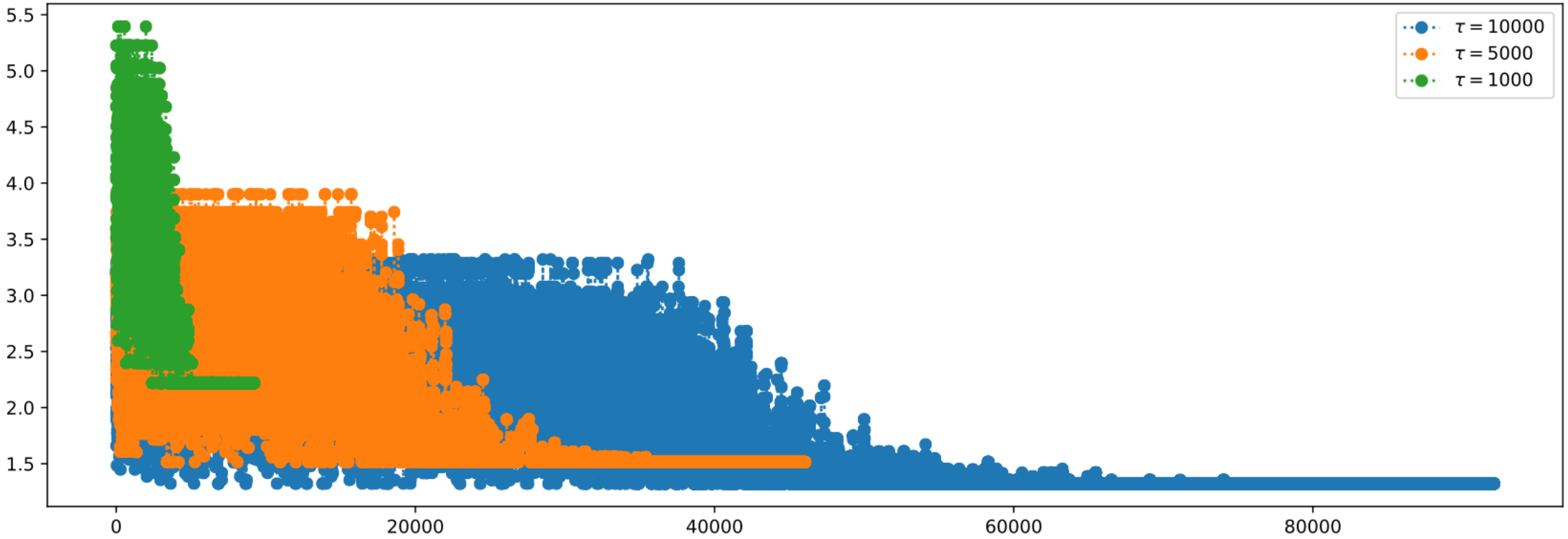
# Simulated annealing for traveling salesman



# Simulated annealing for traveling salesman: Different starting temperatures



# Simulated annealing for traveling salesman: Different cooling rates temperatures



# Today's lecture:
# Simulated annealing, Genetic algorithms

- Simulated Annealing
  - Travelling salesman problem

- Genetic algorithms

# Genetic Algorithms (Pang Ch. 11)

- We saw in the case of simulated annealing:
  - Finding global minima is difficult
  - We can use inspiration from physics in solving unrelated problems in optimization

- Genetic algorithms are techniques for optimization inspired by biology
  - Create “organisms” that store a set of chromosomes
  - Create new organisms by mixing the genes of parents, and allowing for mutations

# Steps for the genetic algorithm

- The problem: find the global minimum of multi-variable function $g(r_1,r_2,\ldots,r_n)$

- 1. Create a gene pool, i.e., an initial population of configurations
  - Configurations are values of variables
  - Can be binary or continuous
- 2. Selection: Choose members to be parents
- 3. Crossover: Produce offspring by mixing their genes
  - Parent chromosomes are cut into segments, exchanged, and joined together
- 4. Mutation: Create random changes to the chromosomes
- 5. In all of the steps above, make sure the configurations with lowest cost (evaluation of $g$) survive

# Example: the Thomson problem

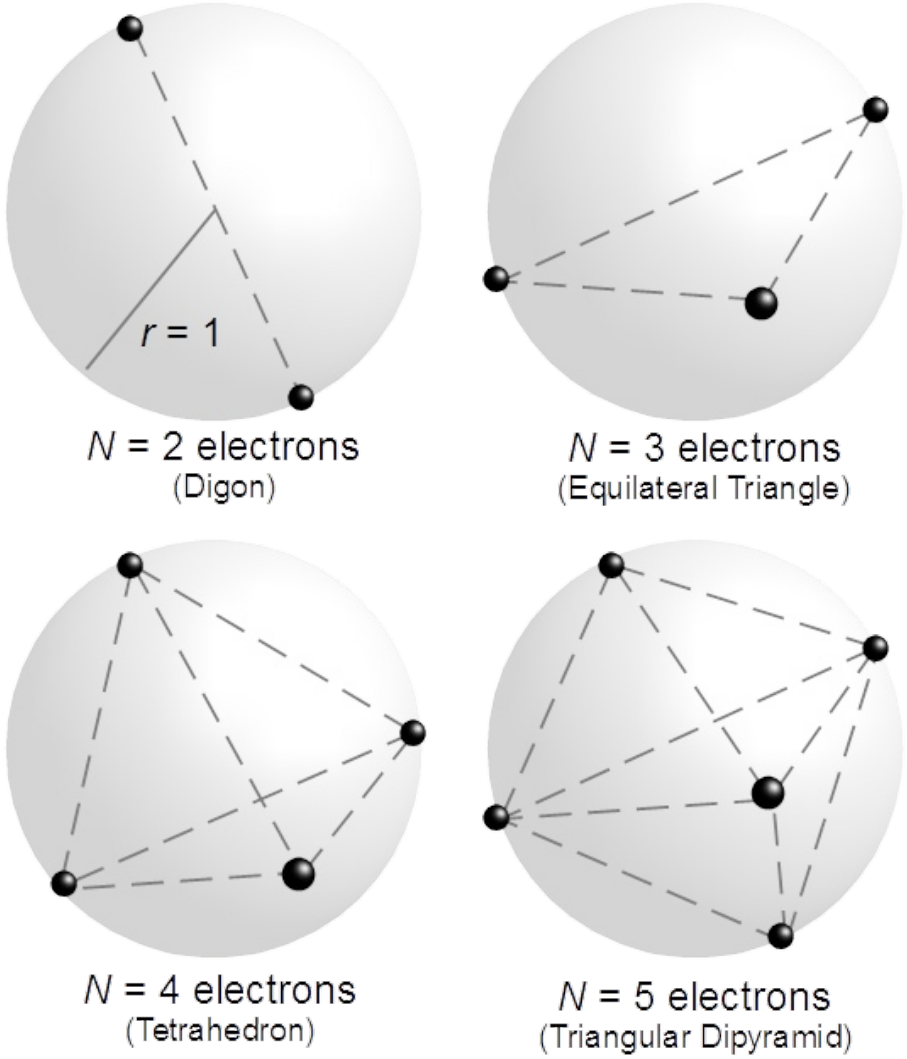
- Consider placing like charges on a unit sphere
- What is the optimal arrangement to reduce the electrostatic energy:

$$U = \frac{q^2}{4\pi\epsilon_0} \sum_{i>j=1}^{N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}$$

- Inspired by J.J. Thomson's "plum pudding model" for atoms

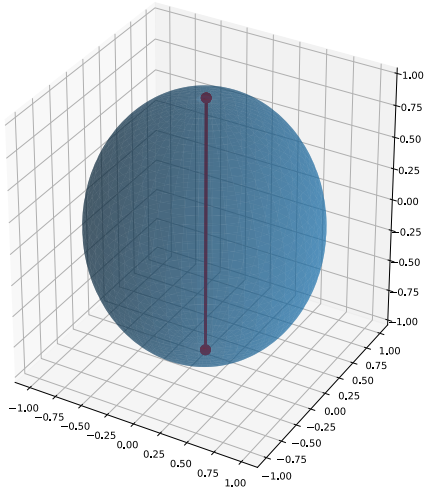
# Solutions to the Thomson problem

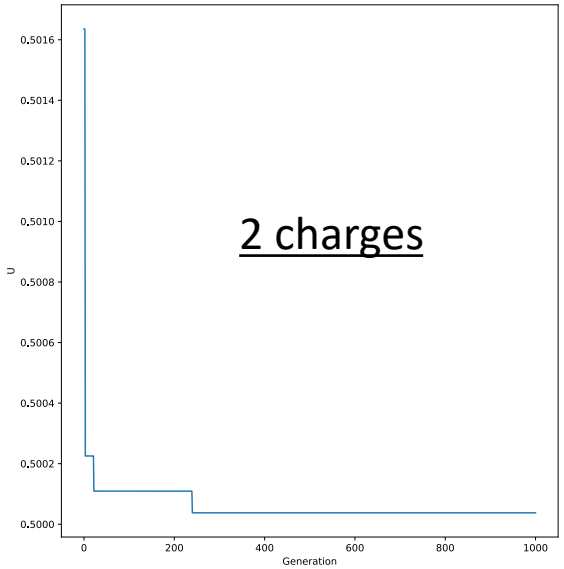
| N | $E_1$ | Symmetry | $\left\|\sum \mathbf{r}_i\right\|$ | $v_3$ | $v_4$ | $v_5$ | $v_6$ | $v_7$ | $v_8$ | $e$ | $f_3$ | $f_4$ | $\theta_1$ | Equivalent polyhedron |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 0.500000000 | $D_{\infty h}$ | 0 | – | – | – | – | – | – | 2 | – | – | 180.000° | digon |
| 3 | 1.732050808 | $D_{3h}$ | 0 | – | – | – | – | – | – | 3 | 2 | – | 120.000° | triangle |
| 4 | 3.674234614 | $T_d$ | 0 | 4 | 0 | 0 | 0 | 0 | 0 | 6 | 4 | 0 | 109.471° | tetrahedron |
| 5 | 6.474691495 | $D_{3h}$ | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 9 | 6 | 0 | 90.000° | triangular dipyramid |
| 6 | 9.985281374 | $O_h$ | 0 | 0 | 6 | 0 | 0 | 0 | 0 | 12 | 8 | 0 | 90.000° | octahedron |
| 7 | 14.452977414 | $D_{5h}$ | 0 | 0 | 5 | 2 | 0 | 0 | 0 | 15 | 10 | 0 | 72.000° | pentagonal dipyramid |
| 8 | 19.675287861 | $D_{4d}$ | 0 | 0 | 8 | 0 | 0 | 0 | 0 | 16 | 8 | 2 | 71.694° | square antiprism |
| 9 | 25.759986531 | $D_{3h}$ | 0 | 0 | 3 | 6 | 0 | 0 | 0 | 21 | 14 | 0 | 69.190° | triaugmented triangular prism |
| 10 | 32.716949460 | $D_{4d}$ | 0 | 0 | 2 | 8 | 0 | 0 | 0 | 24 | 16 | 0 | 64.996° | gyroelongated square dipyramid |
| 11 | 40.596450510 | $C_{2v}$ | 0.013219635 | 0 | 2 | 8 | 1 | 0 | 0 | 27 | 18 | 0 | 58.540° | edge-contracted icosahedron |
| 12 | 49.165253058 | $I_h$ | 0 | 0 | 0 | 12 | 0 | 0 | 0 | 30 | 20 | 0 | 63.435° | icosahedron (geodesic sphere $\{3,5+\}_{1,0}$) |
| 13 | 58.853230612 | $C_{2v}$ | 0.008820367 | 0 | 1 | 10 | 2 | 0 | 0 | 33 | 22 | 0 | 52.317° | |
| 14 | 69.306363297 | $D_{6d}$ | 0 | 0 | 0 | 12 | 2 | 0 | 0 | 36 | 24 | 0 | 52.866° | gyroelongated hexagonal dipyramid |
| 15 | 80.670244114 | $D_3$ | 0 | 0 | 0 | 12 | 3 | 0 | 0 | 39 | 26 | 0 | 49.225° | |
| 16 | 92.911655302 | $T$ | 0 | 0 | 0 | 12 | 4 | 0 | 0 | 42 | 28 | 0 | 48.936° | |
| 17 | 106.050404829 | $D_{5h}$ | 0 | 0 | 0 | 12 | 5 | 0 | 0 | 45 | 30 | 0 | 50.108° | double-gyroelongated pentagonal dipyramid |
| 18 | 120.084467447 | $D_{4d}$ | 0 | 0 | 2 | 8 | 8 | 0 | 0 | 48 | 32 | 0 | 47.534° | |
| 19 | 135.089467557 | $C_{2v}$ | 0.000135163 | 0 | 0 | 14 | 5 | 0 | 0 | 50 | 32 | 1 | 44.910° | |
| 20 | 150.881568334 | $D_{3h}$ | 0 | 0 | 0 | 12 | 8 | 0 | 0 | 54 | 36 | 0 | 46.093° | |
| 21 | 167.641622399 | $C_{2v}$ | 0.001406124 | 0 | 1 | 10 | 10 | 0 | 0 | 57 | 38 | 0 | 44.321° | |
| 22 | 185.287536149 | $T_d$ | 0 | 0 | 0 | 12 | 10 | 0 | 0 | 60 | 40 | 0 | 43.302° | |



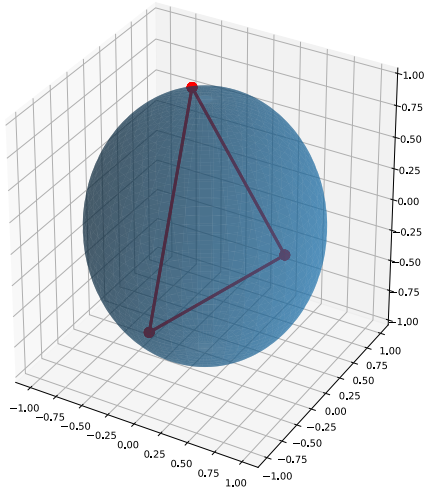

https://en.wikipedia.org/wiki/Thomson_problem

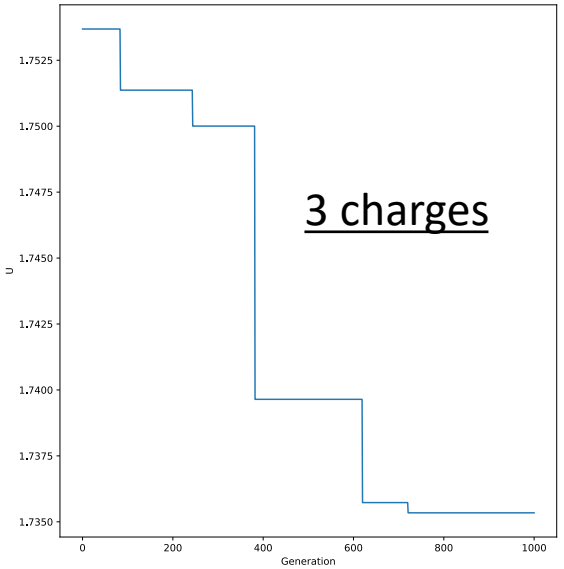
# Solutions to the Thomson problem



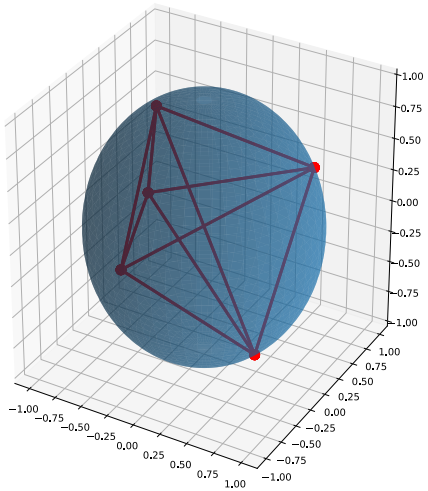


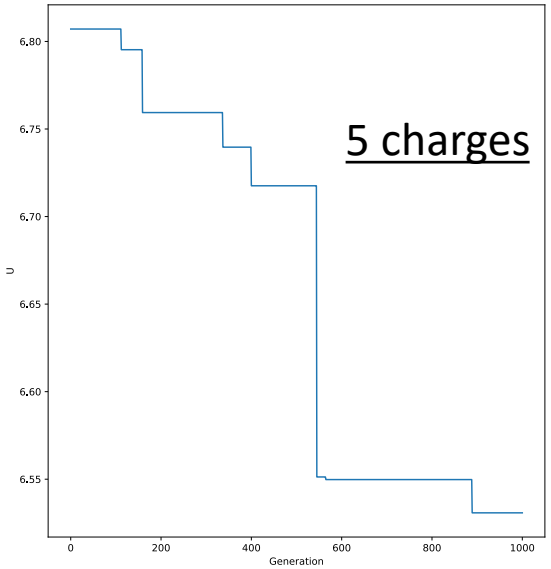

2 charges



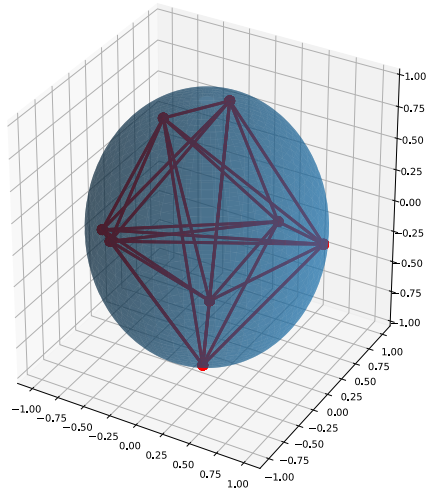


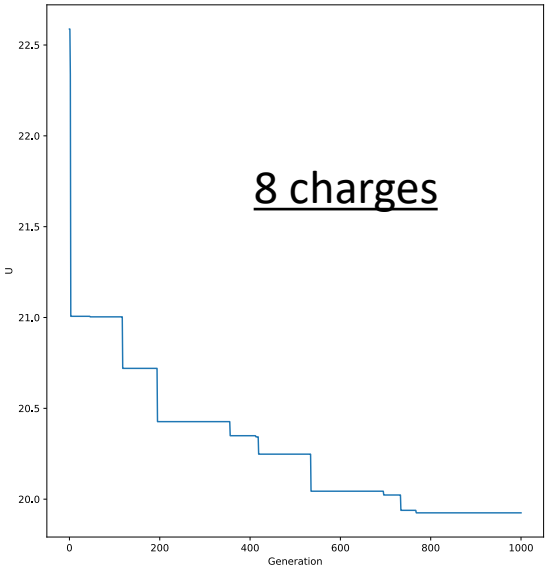

3 charges






5 charges






8 charges

# After class tasks

- Homework 5 due today
- Homework 4 graded
- Final project presentations Dec 12 11:15am-1:45pm

- Readings:
  - Simulated annealing: Newman Sec. 10.4
  - Genetic algorithms:
    - https://en.wikipedia.org/wiki/Thomson_problem
    - Pang Ch. 11