Basin-Hopping Algorithm

An Overview of its Applications to

Global Optimization and Ground State Energy

Between Particles

Ronaldo Ascano

Computational Physics, Fall 2018

University of Nevada, Las Vegas

Abstract

The study of global optimization can applied to a wide variety of problems coming from different scientific fields such as chemistry and physics. With the help of the *basin-hopping algorithm* from the scipy library in Jupyter Python, we are able to find the best global minimum or "ground state energy" between N-amount of atoms (12, 17, 20) with their interatomic/ molecular forces considered. This algorithm's stochastic nature allows it to be much more efficient in finding the global minimum simply by changing some parameters already built-in the algorithm.

Background

Lennard-Jones Potential (LJP)

The LJP equation:

$$V = 4\epsilon\left[\left(\frac{\delta}{r}\right)^{12} - \left(\frac{\delta}{r}\right)^{6}\right]$$

describes the interactive potential energy between two (or more) distinct "non-bonding atoms or molecules reliant upon their distance of separation, *r*, forces of repulsion and attraction. According to chem.libretexts.org, these forces have been taken into account in the equation. The first term $(\delta/r)^{12}$ represents repulsive forces, and oppositely, the second term $(\delta/r)^{6}$ describes the attractive forces.

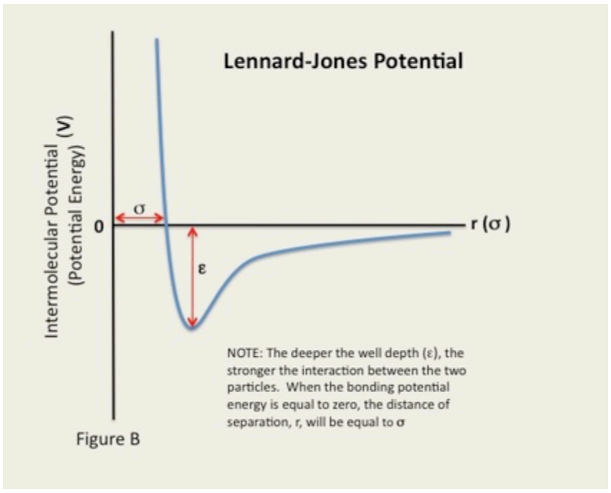


Figure B

The ε in the equation designates the potential well depth. It is a measure for the how strong the frequency between two particles with intermolecular forces in mind. While δ denotes the van Der Waals radius which describes the distance of how close two particles can be, or where V (intermolecular potential energy) is 0. For simplicity's sake, we used $\varepsilon = \delta = 1$. With this we produced a plot similar to the one above that have indicators of where ε and δ are in the graph (LibreTexts.org).

In this research, an important part is to compute the LJP, "ground state," or the global minimum between multiple atoms through the stochastic method behind the basin-hopping algorithm.

Global Optimization and the Basin-Hopping Algorithm

Global optimization is an applied mathematics and numerical analysis branch that deals with the global minimization or maximization of a function or a set of functions to follow under some criteria. Some fields that benefit from this type of study is chemistry (protein-folding prediction by minimizing the free energy function), physics (atomic/molecular dynamics simulations), and some other more (e.g. biology, medicine, chemical engineering). (Wikipedia)

One of the known methods to perform global optimization on functions is simulated annealing algorithm. This mimics a heating and slow cooling technique done on solid metals (Zhu) like steel to soften the material and increase its ductility. This algorithm allows a search for a random best solution of the global minimum in a multistep manner. One of its strengths is that it can easily avoid remaining in the maxima. Though it is imperfect as it tends to get stuck at a local minima as it merely uses the local minimization package that incorporates the conjugate gradient method. This problem can be averted using the basin-hopping algorithm within the scipy package in the Jupyter Notebook.

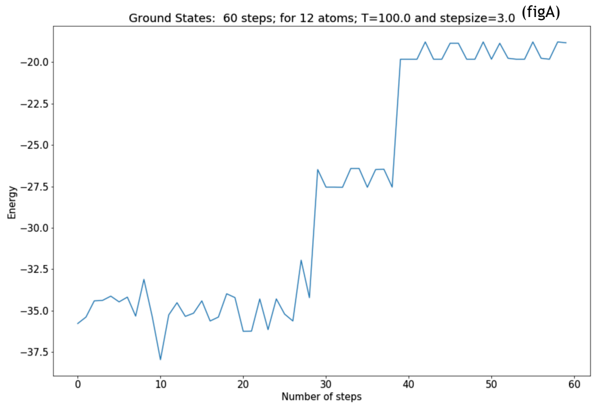
In this overview, we explore the global minima using the Lennard-Jones Potential equation together with the basin-hopping algorithm in scipy to encapsulate the dynamics of the energy landscape of multi-particle structures.

Research Methods and Results

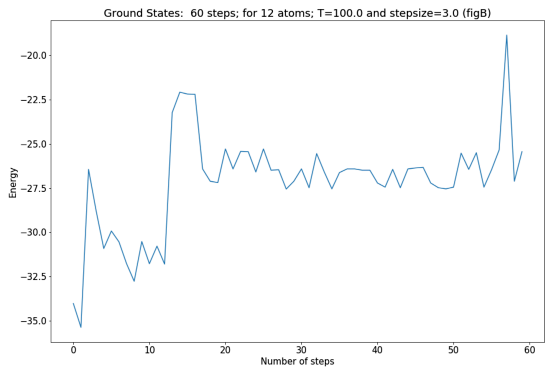
All the calculations were computed using the Jupyter Notebook running Python 3. There were multiple functions to consider to compute the ground states for N-amount of atoms. First, we need the LJP function to calculate the potential energy between two atoms. Secondly is to calculate the total energy between all other atoms considered. With that said, we generated random initial positions for each atoms in a 3d space. These parameters are then fed through the basin-hopping algorithm in scipy. The algorithm intrinsically has a function to perform the search with multiple iterations and setting different parameters such as the temperature and step-size as its acceptance criteria. For this project, I chose 12, 17, and 20 atoms to search the global minima of. When these values are found, these represent the amount of energy in the system or atoms when in a relaxed state. Along with this is the shape or conformation of the atoms in this relaxed state.

N=12

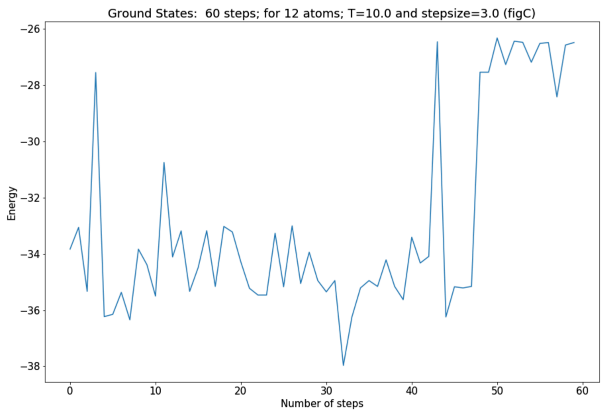
In this research, the found global minimum for 12 atoms is -37.9676. Verifying from this website: *http://doye.chem.ox.ac.uk/jon/structures/LJ/tables.150.html*, this result suffices. Changing the parameter(s) around, with high temperature — at T=100.0 and stepsize=3.0 — it produced a seemingly odd, but familiar shape iterating it 60 times. (See figA.)



When the temperature was raised, it allowed the algorithm to find the ground state at only 10 steps, but as it continues, it exits and deviates away from the true ground state. This resembles a "hop" from a "basin" of local minima. With a high temperature the algorithm will accept even the biggest of jumps with the values. This occurrence is depicted at around steps 30 and 40 in the figure above.



Running the algorithm again to see if it behaves differently, in figB, it searched for the minimum very early but it did not actually find the proper ground state according to the given link. Although, it continued to accept huge jumps with the recorded values as it is still under high temperature. With more tries under high T, there seems to be several instances where I do not correctly find the global minimum but it maintains the basin shape. So it is safe to say that there are some inaccuracies with the algorithm when the temperature is up to 100.

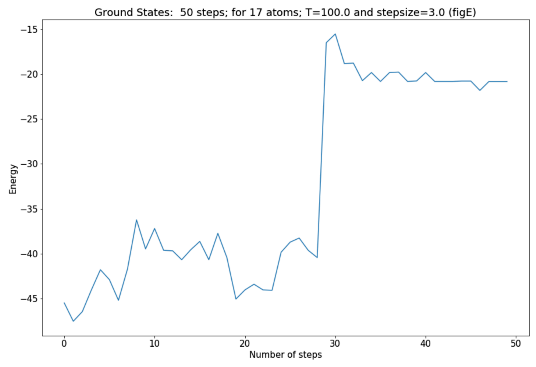


At low temperatures, tuning it down to T=10.0 and stepsize=3.0 (see figC), the inaccuracies still ensues but it finds the minimum in under 40 steps and it is still accepting drastic climbs but losing its frequency. Taking a closer look during other trials in T=10.0, the global minimum can be found twice or more times in one run, and the probability of estimating the proper minimum is much better being on these parameters settings. Also the spread of the data is becoming more localized to fewer "basins" accepting fewer jumps in value.

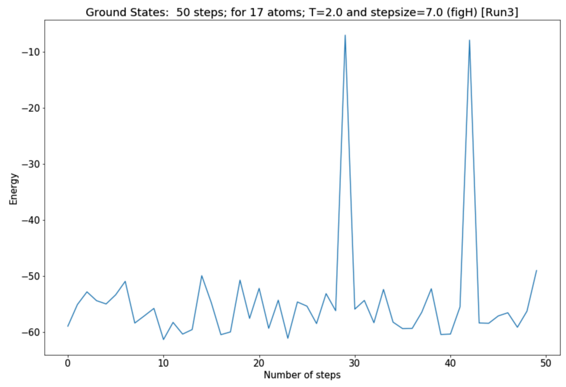
Furthermore, dropping the temperature to T=2.0, the search for the global minimum resulted in some consistency around the general area where the true minimum is at. In other words, it avoided the "hop" and stayed within the “basin.”

N=17

For 17 atoms, the recorded ground state is -61.317995 from the website mentioned website above. Running the basin-hopping program in Jupyter, the best resulting ground state it could produce was -61.307146. Using the parameters T=100.0 and stepsize=3.0, the program provided a plot with the same basin



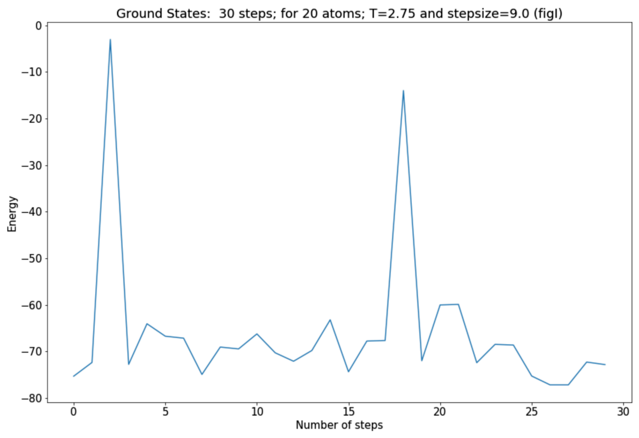
shape (as figE) but not achieving the proper minimum. After many attempts and trying different parameters, still the algorithm could not seem to hit the correct global minimum.



Some days after of last running the program, I tried the parameters T=2.0 and stepsize=7.0 (increasing the step-size by a few notches), the target minimum was finally found; to be exact, it is -61.31799466010683. And then actually found it again on the next run, producing 61.31799466009076 as the global minimum. After a few more runs, it seems that these parameters are consistent in producing the global minimum. In the 3rd run, the algorithm found the global minimum in under 15 steps which is a huge improvement from the previous runs (see figH).

N=20

I wanted try and find the ground state of 20 atoms and see if I could find the right parameters to calculate the global minimum. I assumed that having more atoms it is harder to find the optimal values to help search for the global minimum. After entering many different combinations of parameters, using T=2.75 and stepsize=9.0 (see figI) in under 30 steps, the algorithm successfully discovered the indicated minimum that satisfies the value in the webpage above (-77.177043). From running the algorithm in Jupyter, the best solution found for the global minimum is -77.17704256829511. This graph is an excellent example of how the energy evolutions between each step "hops" down to a basin of global minimum. Huge jumps such as on step 2 and 18 are quickly avoided using the basin-hopping algorithm.

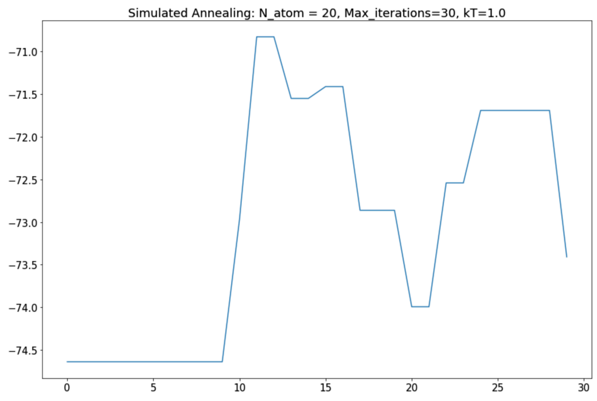


Discussion

The basin-hopping algorithm performs more efficiently in handling multi-iterated and multivariable global optimization computing than just the ordinary minimization under the conjugate descent method from the scipy kit.

It seems to be that the higher number of atoms, the harder it takes as it needs more computing power to calculate the global minimum due to more iterations of simply having more atoms. Having to consider more atoms requires the algorithm to require a higher temperature and step-size to find the ground state energy.

Comparing it to regular minimization tool in scipy that I used through simulated annealing, the objective behaves in such a way that it remains in a local minimum for several steps and only finds the other local minima and deems the smallest the best solution. Here I tested it on N=20 atoms and in 30 steps. (see fig. below)



Also in the trials data that I included below, I aim to validate the parameters I found above in my results. I ran them in 10 trials under niter=10 while changing and keeping parameters constant.

Trials Data

Running multiple trials on a few set of parameters to validate the data found above:

N=12 (niter=10)

	T=1.0, S=3.0	T=2.0, S=3.0	T=3.0, S=3.0
Run 1	Not found	Found	Not found
Run 2	Found	Found	Not found
Run 3	Not found	Not found	Found
Run 4	Not found	Not found	Not found
Run 5	Not found	Found	Not found
Run 6	Not found	Not found	Not found
Run 7	Not found	Not found	Not found
Run 8	Not found	Not found	Not found
Run 9	Found	Not found	Not found
Run 10	Not found	Not found	Found

N=17 (niter=10)

	T=2.0, S=5.0	T=2.0, S=7.0	T=2.0, S=9.0
Run 1	Not found	Not found	Not found
Run 2	Not found	Not found	Not found
Run 3	Not found	Not found	Not found
Run 4	Not found	Not found	Not found
Run 5	Not found	Not found	Not found
Run 6	Not found	Not found	Not found
Run 7	Not found	Not found	Not found
Run 8	Not found	Not found	Not found
Run 9	Not found	Not found	Not found
Run 10	Not found	Found	Not found

N=20 (niter=10)

	T=2.0, S=9.0	T=2.5, S=9.0	T=3.0, S=9.0
Run 1	Not found	Not found	Not found
Run 2	Found	Not found	Found
Run 3	Not found	Not found	Not found
Run 4	Not found	Not found	Not found
Run 5	Not found	Found	Not found
Run 6	Not found	Not found	Not found
Run 7	Not found	Not found	Found
Run 8	Not found	Not found	Not found
Run 9	Not found	Not found	Not found
Run 10	Not found	Not found	Not found

References

Naeem, Rabia. (2017). *Chemistry.LibreTexts.Org*. Lennard-Jones Potential. Retrieved Decemeber 2, 2018, from https://chem.libretexts.org/Textbook_Maps/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Physical_Properties_of_Matter/Atomic_and_Molecular_Properties/Intermolecular_Forces/Specific_Interactions/Lennard-Jones_Potential

Wales Group. (n.d.). Table of Lennard-Jones Clusters Minimum Values. Retrieved December 2, 2018, from http://doye.chem.ox.ac.uk/jon/structures/LJ/tables.150.html

Wikipedia. (2018). Global Optimization. Retrieved December 2, 2017 , from https://en.wikipedia.org/wiki/Global_optimization

Zhu, Qiang. (2018). Global Optimization (III). Retrieved December 3, 2018, from https://github.com/qzhu2017/2017-cmp