Executive Summary - MPI Implementation of an Energy Landscape Visualisation Tool - Stochastic Hyperspace Embedding And Projection (SHEAP)

Supervisor: Prof. Chris Pickard
28 June 2024

Mo Ji

Energy landscape, i.e. potential energy surface (PES), is defined as a framework to describe the potential energy of a system as a function of the positions of its atoms. PES is crucial to represent the structure and thermodynamics of systems of atoms configurations, highlighting stable states (global minima), meta-stable states (local minima) and the transition states (saddle points). Visualising PES is challenging due to the high dimensionality and large quantity of data generated by simulations. Various dimensionality reduction techniques have been adapted to map high dimensional datasets into 2D or 3D plots, such as principal component analysis (PCA), multidimensional scaling (MDS), Isomap, t-distributed stochastic neighbor embedding (t-SNE), and uniform manifold approximation and projection (UMAP). Each techniques come with certain assumptions and limitations, there has not been a manifold learning dimensionality techniques that performs well for projecting high- dimensional PES data into a 2D or 3D plot. Additionally, the existing visualisation tools for PES, such as disconnectivity graphs and sketch maps, also have drawbacks. Disconnectivity graphs do not reflect the spatial relationships among these basins, or the volume of each basin . And sketch maps has been reported that the approach failed to accurately represent the pairwise distance among the data points from different basins . Therefore, SHEAP algorithm has been developed to address the issue of not being able to representing the both local and global PES structure accurately.

SHEAP combines the conditional probability, i.e. student t distribution, concept from t-SNE and the cross entropy cost function from UMAP to preserve both local and global structures of PES. The optimisation process is to minimise the conditional probability of high- dimensional and low- dimensional data. SHEAP also introduces the hard sphere radius to represent the volume of the basin, i.e. the size of a minima. The optimisation can be divided into two stages, low- dimensional position adjustment and hard sphere radius growth. Cost is calculated by the logarithm of the gradient vector norm and exponential moving average is used to smooth out the cost changes during optimisation.

The project aims to reproduce and optimise the SHEAP algorithm in FORTRAN. Message Passing Interface (MPI) will also be implemented in SHEAP to enable it to handle large data sets. It also allows the SHEAP algorithm to take advantage of he computational resources of high performance computing (HPC) facilities. The objectives of the project include:

- Reproduce the SHEAP algorithm core functions. Demonstrate the layout of the local minima and the relative volume of the basins for LJ13 and LJ38 data sets.
- Improve efficiency, maintainability and readability of the code, optimise the code to handle larger data sets.
- Implement MPI for the SHEAP algorithm to accommodate large data sets.

Optimisation strategies include optimising the initialisation and the gradient descent method, reducing the loops, data locality, hoisting and minimising branching. A master- worker arrangement has been implemented for MPI. The master rank distributed the workload among ranks, i.e. loss gradient vector calculation, while the master rank collecting the results and carrying out gradient descent.

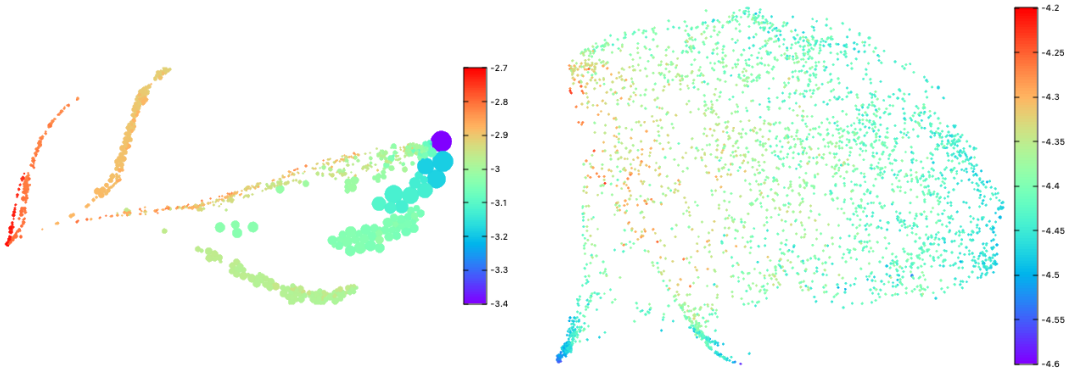
Two initialisation approaches, i.e. random and PCA have been explored. Random initialisation uses a normalised and orthogonalised random matrix, while PCA initialisation uses the eigenvalues and eigenvectors of the high-dimensional positions. Results from the LJ13 and LJ38 data sets showed that PCA

initialisation gave a more structured initial position, and PCA initialisation might have a faster convergence rate at the beginning of the optimisation. But the overall convergence rate was comparable for both initialisation approach.

The convergence rate of the optimisation methods, i.e. TPSD and ADAM, were also compared using the LJ13 and LJ38 data sets. ADAM performed better on the smaller LJ13 data set, and TPSD performed better on the larger LJ38 data set. TPSD might be more suitable for the larger PES data set as it seems to be more efficient to navigate through the local minima and saddle points of complex energy landscape.

MPI implementation has been carried out on the SHEAP algorithm, i.e. parallel computation of the loss gradient vector. It has been found that the MPI could enable the algorithm to handle larger data, the communication overhead increased with increasing number of ranks. The trials running on HPC suggested that the computational time increased significantly when the number of ranks was over 40. Running time for LJ13 data sets increased from 3.8 seconds to over 700 seconds with using 100 ranks.

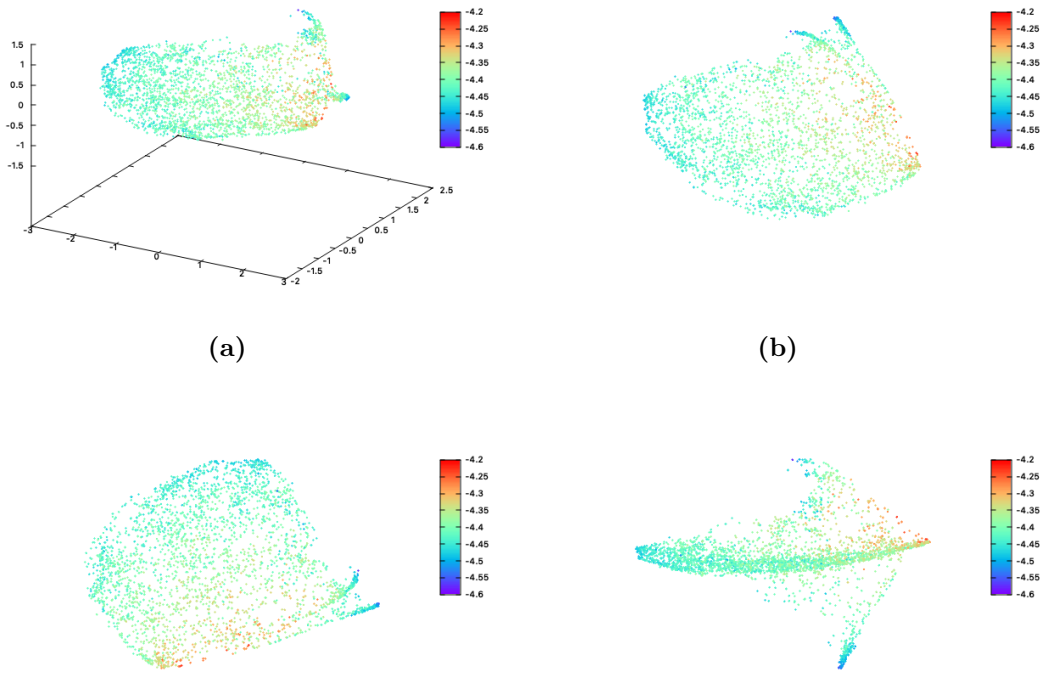
The 2D and 3D maps for LJ13 and LJ38 data sets were successfully reproduced, showing the 'single funnel' and the 'brain like' topology, as shown in Figure 1 and 2. The visualisation results suggested that the manifold learning approach is appropriate to demonstrate the complex structure of the energy landscape. Additionally, it has been found that the parameters, such as the perplexity value affected the visualisation result of LJ13 data set significantly. Therefore, careful consideration is required to select the parameters of SHEAP.



(a) **(b)**

Figure 1: (a)A 2D SHEAP map reproduced from using LJ13 data set (794 distinct minima), the projection used a perplexity of 30, with a minimum sphere radius $R_0 = 0.005$. (b) A 2D SHEAP map reproduced from using LJ38 data set (2966 distinct minima), the projection used a perplexity of 30, with a minimum sphere radius $R_0 = 0.01$The colour represents the ion energy from the data set, and the radius of each point corresponds to the size of the basin volume.

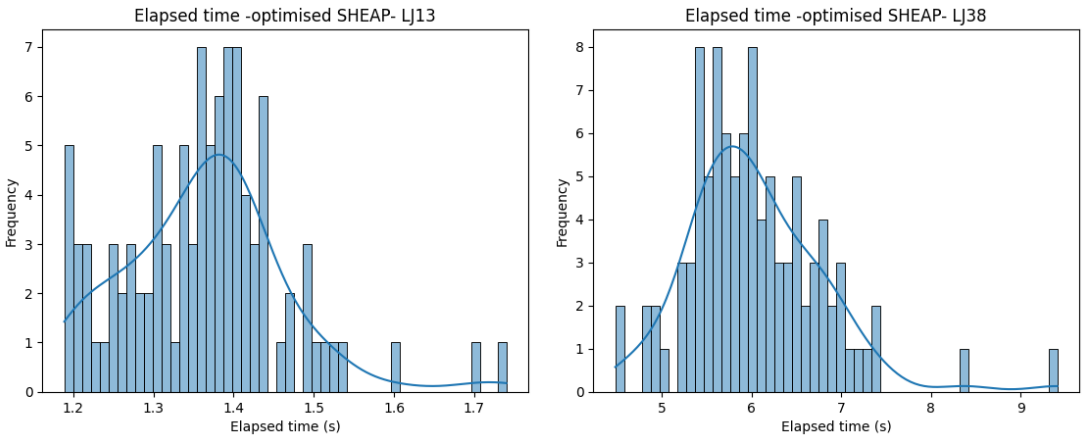
Profiling SHEAP algorithm revealed that the TPSD gradient descent function consumed most of the running time, i.e. 73.8% for LJ13 and 87.5% for LJ38 datasets.Optimisation has been carried out including reducing the number of loops (removing duplicates), converting matrices into 1D arrays, hoisting and minimising branching. The results have been shown in Figure 3. Significant improvement in performance has been achieved, reducing the LJ13 running time from 12 to 1.4 seconds, and the LJ38 running time from 12.9 to 6.49 seconds.The optimised SHEAP algorithm has achieved great speed up, gained capability of running large data set via MPI, also improved the maintainability and efficiency.



(a) **(b)**

(c) **(d)**

Figure 2: (a)A 3D SHEAP map reproduced from using LJ38 data set (2966 distinct minima), the projection used a perplexity of 30, with a minimum sphere radius $R_0 = 0.01$. (b) view angle (0,0) (c) view angle (0, 90) (d) view angle (90,0).



(a) **(b)**

Figure 3: Elaspsed time for running LJ13 (a) LJ38 (b) with the optimised SHEAP code. The figures show the time distribution of 100 runs of the programme.