TU Delft
Delft University of Technology

Minor - Computational Science and Engineering
TW3715TU - Minor Project Part A
Version 1.0 - **UNPUBLISHED**

# Global Geometry Optimisation of Atomic Clusters: a Literature Review

**Einar de Gruyl — Patrick Hibbs — Donghwi Kim — Simon Van Hulle — Marco van Veen**

*Faculty of Electrical Engineering, Mathematics and Computer Science*

*Mekelweg 4, 2628 CD Delft, the Netherlands*

January 21, 2022

## Abstract

**In the past decades, researchers have put a lot of time and effort into finding the optimal stable structures of clusters. The reason behind this is that a large variety of applications, ranging from batteries to applications in medicine, utilise the unique properties of these clusters. Efficiently finding optimal stable structures has proven to be a rather difficult task, which is why various global optimisation methods have been developed for this purpose. This review will provide a detailed description of two commonly used global optimisation methods, basin hopping and the genetic algorithm, and will give a brief overview of some other algorithms which have also been used for geometry optimisation.**

## 1. Introduction

Clusters encompass a relatively new class of material building blocks, consisting of up to hundreds of atoms. Thereby, they can be seen as an intermediate between isolated atoms or molecules on the one hand and bulk material on the other. Their unique optical, electronic and magnetic properties attract increasing interest (Kunwar et al., 2014).

These characteristics have proven to be strongly dependent on the exact geometry and size of the clusters, leading to the question of which configurations are the most energetically favourable for any given number of atoms.

A stable configuration will always have the property that small changes result in higher potential energy. This is referred to as a local minimum in the potential energy surface (Schlegel, 2003). However, when energy is temporarily added to the system, it might stabilise into a different local minimum. It stands to reason that the most stable atom arrangements correspond to the lowest local minima. Finding this global minimum for a certain number of atoms is a computationally expensive problem and requires a systematic searching procedure. In order to produce such a global optimisation algorithm, it is necessary to build a base level of knowledge concerning the state of the art in this type of algorithms.

The purpose of this article is hence twofold: firstly, to review the main properties, characteristics and applications of clusters and secondly, to give an overview of existing global optimisation approaches.

This work starts by introducing the concept of clusters and their applications in section 2. Then, the idea of global optimisation is explored in section 3, along with several of the more popular algorithms to find global minima in section 4. After that, section 5 presents the scope of this group's further research project, as well as some of the tools that will be used and finally, the main conclusions of this literature study are made in section 6, together with an outlook on future challenges.

## 2. Clusters

In the never-ending search for new materials with fitting properties for various applications, researchers have found that these properties are not only influenced by the type of atoms a material consists of. They also very strongly depend on the geometrical configuration in which those atoms are arranged in the material. It turns out that relatively small groups of atoms of the same element behave very differently compared to either individual atoms or bulk systems. This type of small group can then serve as a building block and is often referred to as a cluster (Palmer, 1997).

The properties of clusters tend to change drastically depending on the number of atoms, so adding one element to the group or removing one from it might completely change its optical, electronic and magnetic behaviour,

as well as the way it interacts with matter in its environment. Predicting this behaviour and influencing the formation of clusters therefore offers a lot of opportunities in the development of new materials.

There have been numerous studies about clusters, both in the gas phase and in situations where they are deposited onto a surface (Johnston, 2002). In both cases, the most important parameter to distinguish atomic clusters from bulk material is the ratio of surface atoms to interior atoms. In a bulk material, all the atoms are close together, resulting in a very small surface area compared to the volume. On the contrary, clusters consist of a very limited number of atoms, massively increasing their surface to volume ratio. Exactly how many atoms a cluster can contain is not formally defined, although Johnston (2002) suggests that any aggregate of less than $1 \times 10^7$ particles, can be called a cluster, regardless of its composition. However, Echt (1996) reports that this limit is very much dependent on the specific property that one is interested in.

In what follows, some of the main types of clusters are briefly introduced, after which the relevance of studies in this field is illustrated by several applications. Finally, some more focus is put on metal clusters in particular, since they are closest to the field of study for this project.

## 2.1. Types of Clusters

Depending on the type of atoms that clusters comprise and the type of atomic bonds that are present, they can be subdivided into multiple categories.

The main focus of this work will be on metal clusters. These could either consist of one metal, or multiple different elements. Many of the applications presented in subsection 2.2 involve this specific type of clusters.

When the cluster elements are semiconductors in solid state (e.g. C, Si and Ge), the bonds are typically of strong and directional covalent type (Johnston, 2002). Again, the atoms do not need to be of a single element. When the “semiconductor cluster” is hetero-atomic, the bonds might be polar covalent. This is for instance the case for clusters of gallium arsenide and aluminium phosphide (White et al., 2007).

Furthermore, when the different atoms in the cluster have a large difference in electronegativity, the bonds become ionic. This means that all the ions have a set charge and therefore, the specific composition of atoms in the ionic cluster determines the overall charge of the aggregate.

Finally, it is also possible for molecules to form clusters, using amongst others van der Waals and hydrogen bonds. These molecular clusters are mostly built up with an inorganic core and are very useful, for instance for the modelling of ozone depletion and solvation processes (Pinkard et al., 2018; Johnston, 2002).

Several of the above-mentioned cluster types are shown in Figure 1, to stress that there is a large variety of forms. The lower right image is in principle a complex molecule and maybe not strictly a cluster, but this enzyme can be viewed as a cluster of subunits, where each subunit is in fact a cluster on its own (Solov’yov, 2021).

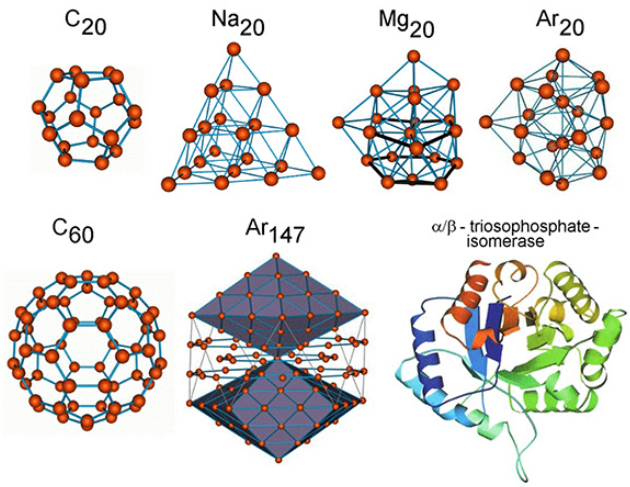


Figure 1: Several examples of atomic and molecular clusters. Retrieved from Solov’yov, 2021, p.1

## 2.2. Applications of Clusters

As mentioned earlier, there is a multitude of applications that make clusters worth investigating. These applications span a variety of different fields, which makes it an even more exciting subject for research. This section uses several of the many examples for illustration.

### *Batteries and Supercapacitors*

One very important area that could be completely revolutionised by clusters is energy storage in batteries and capacitors.

Many performance characteristics for batteries depend greatly on the effective surface area of the electrodes. After all, the surface area has an immediate influence on the efficiency of ion exchange between the electrode and the electrolyte (Rensburg et al., 2014). At the same time, a major objective in battery development is to make them smaller and lighter. As such, increasing the surface area without adding much volume or mass is the goal. This is precisely where clusters could play a role (Das et al., 2014). It was already mentioned that the main feature to distinguish clusters from bulk material is their large surface to volume ratio. Therefore, if one could deposit these clusters onto the surface of the electrodes, they would act as a coating, drastically increasing the active surface area with very limited additional volume and mass. This could lead to improvements in battery capacity, impedance and charging and

discharging rates (Paar, 2014). Of course, the process of placing the clusters onto the surface is not straightforward. This is because it is often energetically favourable for the system as a whole if the cluster atoms were incorporated into the bulk material. Therefore, when the system gains too much energy while a cluster comes into contact with the surface, it might break down and become part of the bulk material (Binns, 2001). Multiple methods have been devised for careful creation and surface deposition of these clusters (Binns, 2001), but this still remains one of the primary challenges.

### *Catalysis on Supported Nanoparticles*

Another important application could lie in the field of catalysts. According to Cheng et al. (2016), platinum-based catalysts are often considered to be the most effective electrocatalysts for many types of reactions. Nonetheless, platinum (Pt) is hardly ever used for this purpose. That is because the material is very expensive and scarce and only the part on the surface is actually useful for catalysing the reaction. It is thus desirable to size down the catalyst in order to obtain a higher platinum utilisation efficiency. The geometry of currently used Pt particles has a high concentration of atoms in the centre, where they are effectively useless, and relatively few surface atoms. Instead, it would be beneficial to use clusters or even single Pt atoms as catalysts (Cheng et al., 2016). As already mentioned above, the main challenge is to keep the clusters from concatenating after they are formed. At the same time, they must be kept close enough together, since their function as effective catalysts requires a high particle density. To this effect, a supportive nanosheet can be used, on which the clusters are deposited and where they are kept in place (Li et al., 2016; Pakkala & Putkonen, 2010). Figure 2 visualises this method of attaching the platinum to a sheet, in this case graphene, to provide structure and stabilise the particles. Note that individual atoms are used in this illustration, but the same principle could work for Pt clusters.

### *Optical Applications*

Because of the enormous amount of freely moving electrons in bulk metals, they are generally good electrical conductors and optical reflectors (Díez & Ras, 2011). However, when scaling down the particles to the nanometer level (cluster size), the energy distribution can no longer be considered continuous, and the band structure is broken down to discrete levels. Due to this fact, clusters react very differently to light. They can intensely absorb light and emit it, by storing and releasing the energy at a later time. Electrons move between energy bands to capture and release energy in discrete steps. As a result, metal clusters have strongly fluorescent behaviour (Kunwar et al., 2014).

This is a very convenient characteristic that can be applied for example in sensing and imaging (Kunwar et al., 2016).

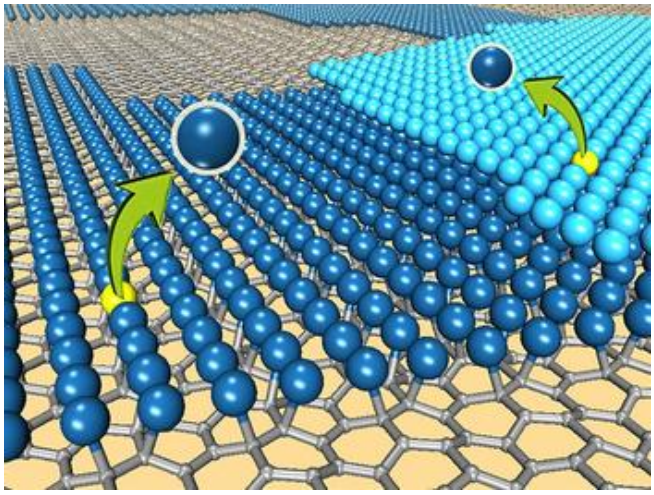

Figure 2: Graphene (gray) supports the layers of Pt (blue), resulting in very efficient usage of this costly material. Retrieved from Choi et al., 2019, p.1.

### *Biological Applications*

This fluorescent property of metal nanoclusters is for example very attractive in the field of bioanalysis. The extremely small size of metal clusters, together with the fact that they have good biocompatibility, render them ideal as fluorescent labels. Moreover, they are low in toxicity, very well soluble in water and have high emission rates; all contributing to their usefulness in detecting biologically important analytes (Shang et al., 2011).

For instance, gold (Au) clusters can be used to detect highly toxic mercury $Hg^{2+}$ ions. Huang et al. (2007) showed that $Hg^{2+}$ triggers the aggregation of these gold clusters, which extinguishes the fluorescence property. This phenomenon is illustrated in Figure 3. They found that the limit of detection for this method (i.e. the minimum concentration that can be picked up) was 5.0 nM* of $Hg^{2+}$. However, Xie et al. (2010) estimated that the limit of detection could even be reduced to about 0.5 nM, by the presence of a chelating ligand. Given that the USA standard for maximum mercury levels in drinking water is 10 nM, this is a rather impressive detection performance (Shang et al., 2011).

A similar technique using silver (Ag) clusters was reported by Zhang et al. (2015), to detect copper $Cu^{2+}$ ions, which is another significant pollutant of the environment.

However, detection and imaging are not the only biological application of clusters. They are also excellent means for drug delivery and photo-thermal treatment (Wang et al., 2011). Furthermore, researchers believe that the clusters themselves could have medicinal ac-

*M = nano-moles of solute per litre of solution. This is a unit for molar concentration.

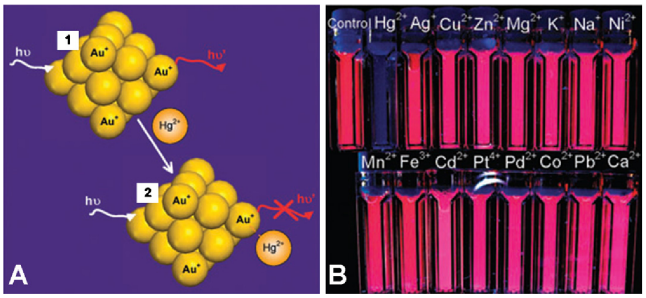


Figure 3: (a) The fluorescence of Au clusters is quenched when they aggregate, due to the presence of $Hg^{2+}$. (b) Solutions with 20 µM Au clusters and 50 µM of various metal ions under UV light. Notice how the $Hg^{2+}$ solution is not luminescent (Shang et al., 2011). Retrieved from Xie et al., 2010, p.961-962

tivity and that they could play a major role in preventing and treating challenging diseases. In one such study, Gao et al. (2019) used in vitro experiments to show that gold clusters can prevent $\alpha$-Synuclein aggregation and fibrillation. Both of these processes are believed to be a critical step in the development of Parkinson's Disease. The same group also performed experiments on mice, which indicated that gold clusters significantly ameliorate the disorders in behaviour that come with this disease. The observation that these clusters could inhibit and even reverse the Parkinson Disease syndromes opens up the options for new medicinal applications.

#### *Data Storage*

In order to store data, flash memory devices are vital for portable electronics. The principle relies on charging data cells to different levels as to represent bits equal to '1' or '0'. To store more data without requiring more physical space, these data cells need to become ever smaller. However, scaling them down to very small levels poses challenges. According to Shaw et al. (2011), there have been proposals to use molecules for these data cells, but as they often have high resistance, low electrical conductivity and limited thermal stability, they are hard to integrate into currently used technologies.

Some metal metal-oxide clusters can hold electric charge very well and can thereby act like RAM. Busche et al. (2014) suggested that clusters of polyoxometalate (POM) could act as storage nodes for this specific type of flash memory, without having problems like thermal instability. Moreover, they claim that "POM clusters can be fabricated with devices that are already widely used in industry".

# 3. Global Optimisation

The various applications described above utilise the unique properties of different types of clusters. Since these unique properties rely heavily on the geometrical configurations of the atoms within a cluster, finding different, stable configurations of atoms is a crucial task. Such stable structures can be found by optimising the geometries of these clusters.

## 3.1. Introduction to global optimisation

Optimisation refers to finding the minimum or maximum output from an objective function over a certain domain. Such optimisation processes can be divided into two categories, namely local and global optimisation. Local optimisation involves finding the optimal solution, e.g., minimum or maximum value, within a specific region of the domain. On the other hand, global optimisation is the task of finding the optimal solution over the entire domain of the objective function. This difference between local and global optimisation is illustrated in Figure 4. While local optimisation is generally much faster to perform, it can only find the global optimum if there exists only one optimum over the entire domain or if the optimisation is performed within the region of the global optimum. Since the objective functions of most applications are rather complex and their structures are not completely known, global optimisation methods are thus often required for finding global optima.

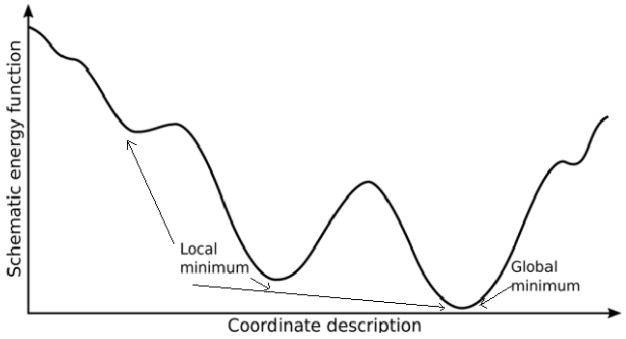


Figure 4: local minima and global minima. Retrieved from Panosetti et al., 2015, p.8044

The importance of global optimisation has led to the creation of a large number of different algorithms. Each algorithm has its strengths and weaknesses depending on the specific application. However, even though a large number of global optimisation algorithms already exist, it is still crucial to research new algorithms or to work on improving existing ones. This can be seen by looking at Levinthal's paradox about protein folding. The paradox describes that a protein would need an enormous amount of time to reach its correct native folded conformation if a random search through all possible conformations is used. Yet, many proteins can fold to their correct conformations within less than a second (Zwanzig et al., 1992). This implies that nature somehow uses a very efficient "algorithm" to perform this complex process of protein folding. This paradox shows the importance of continuous research and development of new global optimisation methods since it appears that there are still many undiscovered methods that can significantly improve on even the best currently-known algo-

rithms.

## 3.2. Global Optimisation for Clusters

In the field of nanoscience and materials science, the objective of global optimisation is to identify clear design paths from complex composite materials and nanostructures (Panosetti et al., 2015). As chemists are mostly interested in stable structures, global optimisation plays a great role in determining the structure and property of materials. Because of this, global optimisation methods have been widely applied in the field of chemistry for tasks such as revealing new structures and catalysis mechanisms, exploring new materials, and calculating thermodynamic properties.

The challenge of global optimisation in chemistry is to find a global minimum from the Potential Energy Surface $E(\mathbf{X})$ by finding the lowest stationary point in potential energy from all the minima. The Potential Energy Surface (PES) of a non-linear cluster is a complex, 3N-dimensional hypersurface that contains the coordinates of all the atoms. As mentioned in Heiles & Johnston (2013), the PES can be reduced to a (3N-6)-dimensional surface by applying a coordinate transformation which removes the translation and rotation of the entire structure, since the PES does not depend on the absolute positions of the atoms in space. A visualisation of such a PES can be seen in Figure 5. Since the function of the PES is quite complex, solving a modified version of the problem by transforming the PES to a simpler form is easier than solving the original PES function. One of the most common techniques that are used to transform a PES is smoothing. The process of smoothing transforms a PES into a set of staircases which is written as

$$\tilde{E}(\mathbf{X}) = \min_{\text{local}} \{E(\mathbf{X})\}. \tag{1}$$

This transformation can be seen in Figure 6, where the black line indicates the original PES function and the red line indicates the staircase pattern. Each step of the staircase represents the value of the local minimum in that area. A benefit of the transformation is that it is able to remove some barriers in the PES function. Barriers are higher energy areas of the PES function which can prevent a solution from moving to the correct global minimum since the solution must be able to perform large jumps over these barriers in order to escape the current local minimum. An example can be seen in the right part of Figure 6, where several tall barriers exist within the original PES function. These are nearly completely removed after the smoothing is applied. However, the smoothing process is not able to remove all barriers in a PES function, which can be seen in the middle of the figure. Efficiently dealing with such remaining barriers still poses a big challenge (Zhang & Glezakou, 2020).

The shape of the PES depends on what type of potential energy model is used. Various models for calculating

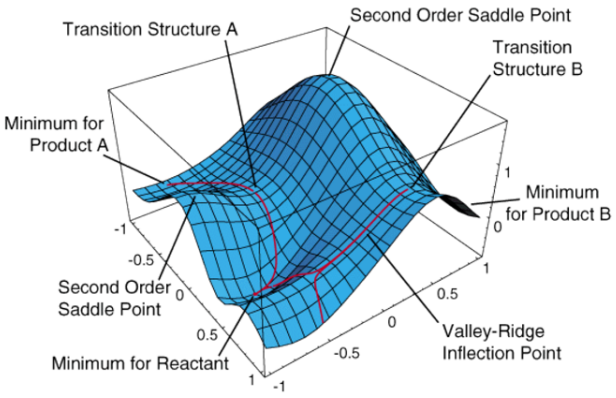


Figure 5: Potential energy surface. Retrieved from Schlegel, 2003, p.1515

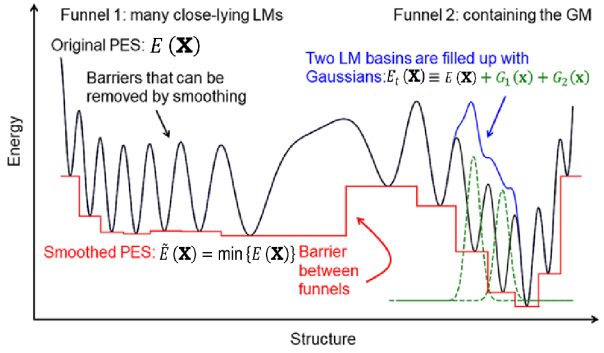


Figure 6: Removing barriers through smoothing. Transforming PES to staircase. Retrieved from Zhang & Glezakou, 2020, p.4

potential energy exist, ranging from complex quantum mechanical models to very simple models which only use a few parameters. A commonly used model for calculating the potential energies is the Lennard-Jones (LJ) potential due to its simplicity. The formula used for describing the LJ potential is

$$E = 4\epsilon \sum_{i<j} \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{2}$$

where $\epsilon$ is the pair equilibrium well depth, $2^{\frac{1}{6}}\sigma$ is the pair separation, and $r_{ij}$ is the distance between two particles (Wales & Doye, 1997). So, the LJ potential only depends on the distance between two atoms. This makes it very easy and quick to calculate, which is why it is often used when testing new global optimisation algorithms. However, even though the potential energy value of a cluster is easy to calculate, finding the globally optimal structures of clusters using LJ potentials is still a difficult problem to solve.

# 4. Methods for Global Optimisation

For the purpose of implementing a global optimisation algorithm for obtaining cluster geometries, many approaches could be considered. This work mainly first focuses on the idea of basin hopping, after which a natural selection approach is also considered. Finally, several of the multitude of other options are briefly discussed as well.

## 4.1. Basin Hopping

The basin hopping algorithm proposed by Wales & Doye (1997) introduced an unbiased method of finding the global minimum of the potential energy surface for Lennard-Jones clusters. This method was inspired by the original Monte Carlo-minimisation described in Li & Scheraga (1987). The algorithm randomly selects new cluster configurations and subsequently performs local minimisation on the generated cluster.

### *Generating the Initial Geometry*

The basin hopping algorithm requires an initial configuration in order to start the global optimisation process. One of the methods described by Wales & Doye (1997) considers a configuration $\mathbf{A}$ of $n$ atoms in 3-dimensional space by uniformly distributing atoms in a sphere defined by

$$\mathbf{A} = r\sqrt[3]{\mathbf{U}} \cdot \frac{(\mathbf{X}, \mathbf{Y}, \mathbf{Z})}{\sqrt{\mathbf{X}^2 + \mathbf{Y}^2 + \mathbf{Z}^2}}, \tag{3}$$

where $r$ is the radius of the sphere, $\mathbf{U}$ is a vector of length $n$ with uniformly distributed values between 0 and 1, and $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are vectors of length $n$ with independent normally distributed values. The radius of this sphere is set to 5.5 Å in Wales & Doye (1997) in order to prevent immediate dissociation of the atoms in the cluster.
This random generation of the initial geometry leads to an unbiased method of finding the global minimum. However, Wales & Doye (1997) also introduced a method of seeding previously found Lennard-Jones clusters in order to run the method. This has the advantage of reaching a stable solution to the global minimisation problem faster, however, this initial configuration may cause some exotic minimal configurations to not be found. This method will be discussed further in *Seeding*.

### *Generating Subsequent Geometries*

Once the initial geometry has been generated, all subsequent geometries can be defined at every next Monte Carlo step as a random displacement in the range $[-1, 1]$ of every coordinate of every atom in the previous geometry. The recurrence relation for all accepted displacements is defined by

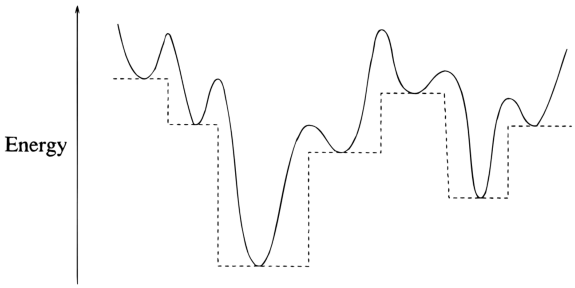


Figure 7: Example of one-dimensional energy transformation. The solid line is the energy of the original surface and the dashed line is the transformed energy $\tilde{E}$. (Wales & Doye, 1997, Figure 2)

$$\begin{cases} \mathbf{A}_0 = \mathbf{A} \\ \mathbf{A}_i = \mathbf{A}_{i-1} + \alpha_i \cdot (\mathbf{X}, \mathbf{Y}, \mathbf{Z}) \end{cases}, \tag{4}$$

where $\mathbf{A}_0$ is the initial geometry defined in Equation 3, $\mathbf{A}_i$ is the geometry at the $i$-th Monte Carlo step, $\alpha_i$ is the step size at the $i$-th Monte Carlo step, and $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are vectors of length $n$ with independent uniformly distributed values in the range $[-1, 1]$. The step size $\alpha_i$ has to be adjusted at some interval to reach to recommended acceptance ratio of $0.5$ this adjustment is necessary in order for the algorithm not to get stuck inside of a basin. Wales & Doye (1997)

### *Local Minimisation*

After generating the new geometry, one has to perform local gradient minimisation. This is necessary since most of the time if the next Monte Carlo step is located in a new basin, the energy is higher and the new configuration will be rejected. The energy for the atom configuration $\mathbf{A}$ is defined here as $E(\mathbf{A})$ and the transformed energy for the atom configuration $\mathbf{A}$ is defined as

$$\tilde{E}(\mathbf{A}) = \min\{E(\mathbf{A})\}, \tag{5}$$

where $\min$ signifies that local minimisation is performed starting from $\mathbf{A}$. This energy transformation is illustrated in Figure 7 for a one-dimensional cluster.

### *Acceptance Criterion*

Of course, not all new configurations need to be accepted. Some configurations will result in a higher energy and they could negatively affect our results. In order to resolve this, the Metropolis criterion will be used. It is defined as

$$P(\mathbf{A}) = \begin{cases} \exp(-\Delta\tilde{E}/T) & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases}, \tag{6}$$

where $P(\mathbf{A})$ is the chance of selecting the new geometry, $\Delta\tilde{E}$ is the change of the transformed energy of the

previous configuration to the new configuration, and $T$ is the temperature and decides how often a higher energy will be accepted Metropolis et al. (1953). This criterion will accept all configurations that have a lower energy than the last configuration, however, it also might accept some higher energy configurations where configurations closer to the previous configuration have a higher chance of being chosen. This selection of higher energy configurations may allow us to find more feasible energy configurations.

#### *Seeding*

As mentioned earlier, one can also use seeding as an alternative method for initialising the basin hopping algorithm. We can choose here from the two clusters with one more and one less atom for the best results. When starting from the cluster with one more atom, we can remove the atom with the highest pair energy $E(i)$ as defined by the LJ potential

$$E(i) = 4\epsilon \sum_{j \neq i} \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right], \tag{7}$$

so that the total energy is

$$E = \frac{1}{2} \sum_{i} E(i). \tag{8}$$

After the removal of this atom, the rest of the algorithm can be run as described in the previous sections.

For the cluster with one less atom, Wales & Doye (1997) took a different approach than described before. They froze all atoms in the cluster for the first 100 steps of the algorithm and added one atom to the cluster at a random position outside of the core and performed only angular moves on that atom for those first 100 steps. Each angular displacement consisted of a random displacement by spherical polar coordinates $\theta$, $\phi$ and the radius $r$ as the maximum length between atoms inside the cluster.

## 4.2. Genetic Algorithm

The genetic algorithm is a method for performing global optimisation which is inspired by the theory of evolution. In order to maintain and advance the population, the algorithm picks the fittest members of the population which will be used for mating. Mutations can also occur to any member of the population, randomly changing their properties. After these steps have been performed, natural selection will be applied where only the most suitable individuals will survive into the next generation. In the context of geometry optimisation, this algorithm will be applied to a population consisting of various geometries with the goal of finding the most energy-efficient configurations. A high-level overview of the complete algorithm can be found in Figure 8.

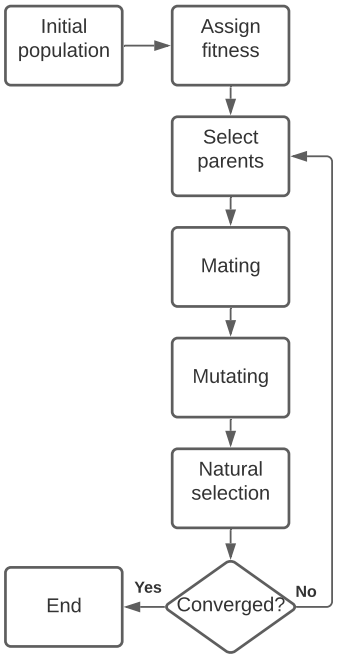


Figure 8: A high-level overview of the genetic algorithm. First, an initial population is created. Then, a number of iterations will take place where new clusters are generated until the convergence criterion is met.

#### *Generating the Initial Population*

In order to start with the genetic algorithm, an initial population must be constructed. The size of this population is denoted by $n_{pop}$. Generally, a population size between 10 and 30 is chosen for these types of problems (Johnston, 2003). Once a suitable population size has been set, the initial members of the population are generated in a randomised manner. This random generation is done by choosing the $x$, $y$, and $z$ coordinates of the atoms of each cluster at random within a suitable range depending on the type of cluster at hand. Generating initial structures like this can however lead to physically infeasible configurations where the structures may be fragmented into small, unconnected pieces. This may lead to local optimisations not converging properly, or it may cause the search to take some time before it starts to find feasible improvements. Averkiev (2009) describes an algorithm called coalescence kick which aims to combat this issue. This algorithm performs a step where all the individual fragments of connected atoms are located and then randomly pushed slightly closer to the centre of the cluster. This process is then repeated until all fragments are connected, leading to feasible starting solutions.

The randomised generation described above leads to an unbiased search for the optimal solution. However, in the case where some additional information is available about the clusters, a slightly biased approach may prove to be beneficial for the performance of the al-

gorithm, such as improved convergence of the overall algorithm or finding new optimal structures (Johnston, 2003; Heiles & Johnston, 2013). This biased search can be achieved by including some predetermined structures into the initial population with the aim of steering the search in a certain direction.

### *Selecting Parent Clusters*

Once an initial population has been generated as described above, suitable parents must be selected for mating. First, all of the population members have to be relaxed to the nearest local minimum to obtain the potential energies $E_i$. This is done by minimising the potential energy of a cluster with respect to the coordinates of all atoms belonging to that cluster. Generally, some unconstrained nonlinear optimisation algorithm using gradients is used for such energy minimisation problems. For example, Deaven & Ho (1995) makes use of the conjugate gradient method when locally minimising energies. Johnston (2003), on the other hand, suggests using the L-BFGS algorithm, which is a memory-efficient version of the original BFGS algorithm.

Next, all of the previously calculated energy values $E_i$ of each cluster are transformed as seen in Heiles & Johnston (2013) to obtain the normalised energy values $\rho_i$:

$$\rho_i = \frac{E_i - E_{min}}{E_{max} - E_{min}}. \tag{9}$$

The normalised energies $\rho_i$ can then be used to assign a fitness value $f_i$ to each cluster, which can be used to select parents. Johnston (2003) provides the three most commonly used fitness functions, which can be seen below in Equation 10, 11, and 12. Equation 11 assigns a linearly scaling fitness value to each normalised energy $\rho_i$, while the fitness values assigned by Equation 10 drop off quickly for worse clusters. Since choosing parent clusters is done randomly based on these fitness values, each fitness function has a different influence on this selection process due to the different scalings. However, even though they have some influence over the selection process, using different fitness functions does not seem to have a large influence on the success rate of finding the global minimum (Johnston, 2003).

$$f_i^{exponential} = \exp(-\alpha\rho_i), \tag{10}$$

with $\alpha$ generally being set to 3.

$$f_i^{linear} = 1 - 0.7\rho_i, \tag{11}$$

$$f_i^{hyperbolic} = \frac{1}{2}\left[1 - \tanh(2\rho_i - 1)\right] \tag{12}$$

After fitness values have been assigned to all of the clusters in the population, the parents can be selected. Two common selection methods exist which are outlined in Johnston (2003). The first selection method is the roulette wheel method. For this method, a single cluster is chosen as a potential parent. The corresponding fitness value $f_i$ is then compared to a random number between 0 and 1. If the fitness value is larger than this random number, then it is chosen as a parent. If the value is smaller, then this candidate cluster is ignored and another candidate cluster will be tested. The second method for selecting parents is the tournament method. Two or more random clusters are first picked from the entire population. Then, the cluster with the highest fitness value will be selected as a parent for the mating process. When two parents have been selected using either the roulette or tournament methods, a pair of parents can be formed. This process of selecting parent pairs is repeated until some predefined number of children can be created. The number of children required is usually set to be about 80% of $n_{pop}$ (Johnston, 2003).

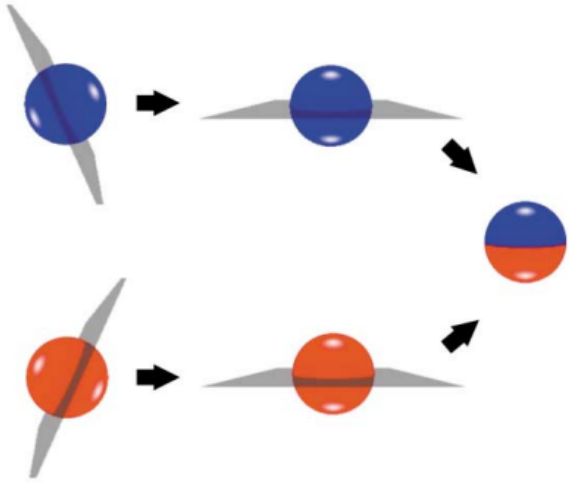

Figure 9: Visualisation of the cutting method for mating described by Deaven & Ho (1995). Retrieved from Johnston, 2003, p. 5

### *Mating*

The most commonly used technique for creating offspring from a pair of parents is outlined in Deaven & Ho (1995). In this method, a random plane through the centres of both parent clusters is chosen. The two-parent clusters are then rotated such that the chosen plane is horizontally aligned with the $xy$-plane. Then, the parent clusters are cut along this plane and a child cluster is created by combining two opposing halves of each parent. A visualisation of this technique is presented in Figure 9. This method, however, may lead to the child not having the correct amount of atoms. In order to deal with this, both parent clusters can either be moved an equal distance towards each other or away from each other until the child contains the correct amount of atoms.

Some variations of this cutting method are mentioned by Johnston (2003). One such variation is to consider us-

ing a weighted approach to cutting the clusters according to the fitness values of the parents. Instead of cutting both parent clusters through the centres which results in equally sized halves, the planes will cut the parents through different points based on their fitness values. This way, the child cluster inherits a larger amount of atoms from the fittest parent. Another method described in the article is using two planes for cutting the parent clusters. This leads to the parent clusters being split into three separate pieces, which are then again combined to create a child cluster.

#### *Mutating*

Mutation operations aim to modify clusters by making small random changes to their structures. The mating process described above does not introduce new genes to the population pool, since it only creates children from already existing structures. This may lead to population stagnation where the current best solutions get stuck around the same local minima. By randomly mutating some members of the population such stagnation can potentially be avoided. The probability of applying a mutation operation on some cluster is $P_{mut}$. This probability is set to a predefined value, which can be configured for the specific geometry optimisation problem at hand. These mutation operations can either be performed on only the newly generated children from the mating procedure (Deaven & Ho, 1995), or they can be applied to both the children and the entire current population (Heiles & Johnston, 2013).

Four possible mutation operations are described in Johnston (2003). The first type of mutation aims to randomly displace a number of atoms within a cluster. This can be done by replacing the coordinates of some atoms with new randomly generated coordinates, or by using the version from Deaven & Ho (1995) which moves the atoms a random distance into a random direction for a random number of times. The second mutation operation aims to modify the structure by rotating a part of the cluster. The cluster is divided into two halves using a horizontal plane parallel to the $xy$-plane. The top half of the cluster is then rotated around $z$-axis by a random angle. The third operation simply replaces an entire cluster with a new random cluster which is generated in the same manner as the initial clusters. Finally, the last mutation operation swaps the atom types of some pairs in a cluster. Since this operation requires clusters of multiple elements, it can only be useful in hetero-elemental clusters. One additional mutation which may be interesting is described in Paleico & Behler (2020) and is called the mirror and shift mutation. In this mutation operation, a random plane is going through the centre of the cluster is selected and the cluster is then mirrored across this plane. The mirrored half is also shifted away from the plane by a small distance. This mutation allows for the creation of symmetric clusters which tend to have lower energy values.

It is possible to either use just one of these types of mutation operations or to combine multiple different mutation operations. Combining multiple different mutation types has been shown to be very beneficial in certain applications. For example, Darby et al. (2002) found that combining both the replacement and swapping mutations for bimetallic gold-copper clusters lead to finding the global minimum much more frequently, and in some cases finding new global minima.

#### *Natural Selection*

The natural selection procedure is the final step of a genetic algorithm iteration. In this step, the fittest candidates are selected for survival into the population of the next generation. To select the fittest candidates, the newly created clusters from mating and mutating must first be locally minimised and their fitness values must again be calculated. The fitness values of the current population and of the new clusters are then combined into a list and sorted. The first $n_{pop}$ clusters with the highest fitness values (lowest energy values) are then selected to form the new population of the next generation. This selection strategy where the previous population is included in the selection procedure is known as an elitist selection strategy. An elitist selection procedure guarantees that the fitness values, and thus the best solutions, cannot get worse between subsequent generations in case the children and mutants have worse fitness values.

As mentioned before, stagnation of the population can pose a problem for the performance of the algorithm. The selection procedure described above adds another potential danger for stagnation. If various clusters end up with very similar structures, all of these nearly identical clusters might be added to the next generation. This may lead to the solutions getting stuck around just a few local minima for the subsequent generations. Thus, maintaining a variety of different structures is crucial for the performance of the algorithm. In order to do this, Deaven & Ho (1995) introduces $\delta E$ which represents a small constant. Then, for a new candidate to be allowed into the next generation, the energy value of the candidate must differ by at least $\delta E$ from all of the clusters already included within the next generation. If this is not the case, the candidate will be removed from consideration and the next fittest cluster will be considered as a candidate.

A more sophisticated method for preventing population stagnation is presented by Sierka (2010). The selection procedure in this paper makes use of geometrical information in order to maintain a structurally diverse population. This is achieved by grouping the clusters with similar structures together. To find which clusters are similar, some statistics about the geometries of the clusters are used. This may be the radial distribution of atoms, the average distance between the atoms, or

the variance and skewness of the distribution of atom distances. Once groups of similar clusters have been formed, the clusters within each group are then sorted by their fitness values. The best cluster from each group is then taken and added to a list that holds the best clusters from each group, sorted by fitness. Then, the second-best cluster from each group is added to this list in a similar fashion. This is repeated until all clusters have been added. Finally, the first $n_{pop}$ cluster are then picked from the list to form the new generation, which should contain the most structurally diverse clusters.

After the fittest $n_{pop}$ clusters have been chosen using one of the selection procedures described above, the entire process of parent selection, mating and mutating, and natural selection is repeated. This is repeated until a certain stopping criterion is met. Commonly used stopping criteria are either a maximum number of total generations or a maximum number of generations where the best solution has not changed (Heiles & Johnston, 2013).

#### *Parallelising the Genetic Algorithm*

Parallelising the genetic algorithm can decrease the amount of time used by spreading the required computations over multiple processors. Since all the operations required for producing new clusters can be performed independently for each new cluster, the genetic algorithm has a lot of room for parallelisation. Some different techniques for this are described below.

Heiles & Johnston (2013) describes the "pool-based" genetic algorithm. Here, the initial population size is much smaller than $n_{pop}$. Then, new clusters are being created and added to the population until the limit $n_{pop}$ has been reached. Once this happened, the natural selection step will take place to reduce the size of the population again. This method allows for the simultaneous creation and local minimisation of new clusters until the population limit $n_{pop}$ has been reached, where all the clusters have to be combined for sorting.

Two additional methods which provide a roughly linear speedup of the algorithm with the number of processors are outlined by Johnston (2003). The first algorithm is known as the distributed parallel algorithm. Here, the parent selection, mating, and mutating operations are spread across multiple processors while each processor holds a copy of the entire population. This approach differs from the previously described method in that it does not use a population that is smaller than $n_{pop}$. This method simply spreads the generation of new clusters across various processors, while using the same genetic algorithm steps as originally described above. The benefit of this is that new clusters are still being generated from a larger population that contains a wider variety of structures, allowing for a larger area of the solution space to be searched.

The second method for parallelisation from Johnston (2003) is the sub-population algorithm. This algorithm performs a separate genetic algorithm on each processor where each processor has its own small population. The small population sizes on each processor can lead to the problem of population stagnation. Thus, a procedure for introducing variety must be set up. This can be achieved by allowing pairs of processors to exchange genetically different clusters between each other.

### 4.3. Other Methods

Because the geometry optimisation problem appears to be quite challenging to solve in many cases, many other algorithms besides the basin hopping and genetic algorithms described above have been applied to try to solve it for various cases. A number of these algorithms are briefly described in the following sections.

#### *Particle Swarm Optimisation*

Swarm intelligence algorithms have been applied to the geometry optimisation problem a number of times in the past, with particle swarm optimisation (PSO) being the most common one. The algorithm for PSO is inspired by the behaviour of flocks of birds or schools of fish in nature. First, a population of particles is randomly generated, where each particle contains a random position and a random velocity. Next, the velocity of each particle is updated based on the particle's current velocity, the particle's own best position so far, and the best-known position within the entire swarm of particles. Additionally, some random changes may be applied to the velocity of the particle in order to prevent entrapment at a local minimum. In the final step, the particle's position is updated according to its velocity. The PSO algorithm then continues updating velocities and positions until some convergence criterion is met.

An example of PSO being applied to geometry optimisation is the work by Lv et al. (2012), which used the PSO algorithm for the optimisation of Lennard-Jones (LJ) clusters. They succeeded in finding the previously identified global minima for six LJ clusters ranging from 26 to 150 atoms. However, they managed to find the global minima much more efficiently than the previous methods, which indicates that PSO may prove to be a promising algorithm for future applications. Additionally, the PSO algorithm is also simpler to implement compared to some other algorithms such as the genetic algorithm.

#### *Artificial Bee Colony*

The artificial bee colony (ABC) algorithm is another, more recent example of a swarm intelligence algorithm. It was originally proposed by Karaboga & Basturk (2007). The ABC algorithm is based on the behaviour of

bees within a hive. It makes use of three different types of bees, namely employed, onlooker, and scout bees. These bees then search for suitable nectar locations, which in the context of geometry optimisation represent different cluster structures. In each iteration employed bees look for nectar around previously known nectar locations and nectar locations communicated to them by various other bees. Onlooker bees then communicate with the employed bees in order to find out what the best current nectar locations are and proceed to search for new nectar around these specific locations. Finally, after this has been done the worst known nectar locations are discarded. Scout bees will then randomly search new areas for nectar to replace these discarded locations.

Zhang et al. (2015) was the first to apply the ABC algorithm to geometry optimisation. Their work has shown that the algorithm manages to show very promising results for clusters using long-ranged potentials. However, the results for short-ranged potentials, used by for example metallic clusters, have been poor due to the algorithm often becoming trapped in local minima.

#### *Tabu Search in Descriptor Space*

Tabu search is a strategy that stores information about the solutions obtained throughout an optimisation process and uses this historical information to avoid generating the same solutions again in future iterations (Glover, 1990). To prevent generating such duplicate solutions, each obtained cluster is characterised using some descriptors. These descriptors are then used to compare clusters in order to find out whether these new clusters are sufficiently different from previously obtained clusters. This tabu search strategy can be used in combination with any existing global optimisation algorithm with the goal of improving the overall search process.

A separate algorithm based on the tabu search method has also been developed and used for geometry optimisation by Cheng & Fournier (2004). The algorithm used by them is known as the tabu search in descriptor space (TSDS). In this algorithm the descriptors which characterise structures are used to obtain approximated energy levels for new clusters based on the descriptors and energy levels of previously found clusters. Cheng & Fournier (2004) uses 6 different descriptors to achieve this, which use information such as mean atomic coordinates and mean nearest neighbour distance. Once the energy levels of the new clusters have been approximated, a few clusters with the best energy values are chosen and exact energy level calculations are performed on these clusters. This approach can thus greatly reduce the computational efforts since much fewer exact energy calculations are performed in each iteration. However, the descriptors must be able to provide relatively good approximations of energies for new structures. Furthermore, the performance of the algorithm also increases the longer the algorithm runs, since more clusters with new descriptor and energy values are located which can be used to provide better approximations for future clusters.

Cheng & Fournier (2004) has only tested the TSDS algorithm using LJ clusters up to 40 atoms. Since the energy calculations for LJ clusters are rather simple, a TSDS iteration requires more time than simply calculating all the energy values of newly found LJ clusters. However, it managed to find the same global minima as other algorithms while requiring significantly less exact energy calculations. This indicates that the TSDS algorithm may be very promising when analysing other types of clusters that require computationally expensive energy calculations.

#### *Minima Hopping*

Another algorithm that uses historical information during the search is the minima hopping (MH) algorithm described by Goedecker (2004). The MH algorithm also makes use of two dynamically adjusted parameters, namely the kinetic energy $E_{kinetic}$ and energy difference threshold $E_{diff}$. First, the cluster is changed by running a short molecular dynamics simulation which starts from the current minimum and uses the $E_{kinetic}$ parameter. The result of the simulation is then relaxed to the nearest local minimum. If this new local minimum is the same as the current local minimum or if it has already been visited before, then the molecular dynamics simulation is run again with a slightly higher $E_{kinetic}$ value. Once a new local minimum has been found, it is accepted as the new current solution if it is at most $E_{diff}$ larger than the current minimum. The value of $E_{diff}$ is dynamically changed throughout the algorithm such that about half of the new minima are accepted and half of them are rejected.

Goedecker (2004) tested the performance of the algorithm on LJ clusters and compared the performance to the basin hopping method. For small 19-atom LJ clusters, the performances were very similar. However, for 38-atom LJ clusters the MH algorithm was significantly more efficient than the basin hopping method since it only had to visit about 34000 minima compared to the 75000 minima of the basin hopping method. This performance increase is likely due to the fact that the 38-atom LJ cluster has two deep lying minima and the MH algorithm was able to escape the wrong minimum much faster, while the basin hopping method got stuck for a large number of iterations.

#### *Neural Networks*

Machine learning methods have received a large amount of interest from many different fields in the past years, and this is the case for the field of geometry op-

timisation as well. Within geometry optimisation, such machine learning methods can be applied to find the energy values of various clusters much more efficiently compared to the existing traditional methods. Portman & Tamblyn (2017) has compared various machine learning algorithms for this purpose and concluded that the algorithms utilising neural networks (NN) were the most accurate and scalable compared to the other machine learning algorithms.

Paleico & Behler (2020) has used NNs to obtain the neural network potentials (NPP) which can be used instead of normal potential energy calculations. The NPP is first trained on a dataset containing a huge amount of varying structures and energy values. A genetic algorithm is then used which applies these NPPs for energy calculations. They have managed to find several important structural trends and structural families appearing at low energy levels for small copper clusters on large ZnO$(10\bar{1}0)$ surfaces. This was achieved in a fraction of the computational costs required by traditional first principle methods.

Another recent example of NNs being used is the work of Mitra et al. (2021), which combined convolutional neural networks (CNN) with a PSO algorithm. The PSO algorithm using the CNNs for energy calculations managed to find the same global minima for $C_5$ clusters significantly faster compared to simulated annealing or advanced BH algorithms. However, for $N_4^{2-}$ and $N_6^{4-}$ clusters only structures close to the known global minima were able to be located. The structures at the true global minima were then obtained by applying some post-processing techniques to the best structures found by the CNN-based PSO algorithm.

The algorithms utilising energy calculations based on NNs have shown varying success depending on the application. However, NNs have only gathered a larger amount of interest from the field of geometry optimisation in recent years. This indicates that such NN-based algorithms may still show significant improvement in the coming years.

#### *Predatory Genetic Algorithm*

The traditional genetic algorithm (GA) as previously described is a widely applied global optimisation algorithm. Due to its popularity, a large number of variations have also been created for specific problems. One such variation used in geometry optimisation is the predatory genetic algorithm (PGA) developed by Manby et al. (1998). The PGA takes the idea from natural selection used in the traditional GA and adds predators to the system which remove undesired genes from the population. Manby et al. (1998) has applied this by first running a normal GA algorithm to identify the best known global minimum. Then, a PGA is run which is similar to the GA, but while it runs it removes candidates which have structures very similar to the previously identified global minimum. For the next PGA, both previously found global minima are now consistently deleted from the population. This process continues until a sufficient number of PGAs have been run, or the desired amount of low-lying isomers have been obtained. While Manby et al. (1998) uses the PGA to remove the previously found global minima, it is also possible to adapt the algorithm to remove other undesired or unrealistic structures for the population with the goal of finding higher quality solutions.

The PGA used by Manby et al. (1998) is able to effectively find a number of different low energy structures which can be of interest in some applications, such as carbon clusters (von Helden et al., 1991). The algorithm is also able to identify new global minima in some cases, such as for $Al_{10}$ clusters. This may be possible due to other algorithms potentially becoming stuck in some very low, but not minimal section of the PES. The PGA approach removes this deep-lying dead end, allowing the search to continue in other directions.

## 5. Project Plan and Software Tools

In the next quarter, the group will take action in implementing its own global optimisation algorithm. The project plan containing the schedule will be discussed in subsection 5.1. An overview of the Atomic Simulation Environment (ASE) package, which will be used during the project, is given in subsection 5.2.

### 5.1. Project Plan

The main plans for the further project in the next quarter are visible in Table 1, along with when they are planned to be finished.

Table 1: Project Timeline

| Task | Deadline |
|---|---|
| Familiarizing with ASE | 12 November 2021 |
| Developing the global optimisation algorithm | 10 December 2021 |
| Making the code parallelizable and building the Raspberry Pi cluster | 14 January 2022 |
| Writing the final report | 28 January 2022 |

The first task relies on getting familiar with the ASE library. Starting with the basics is crucial in order to develop skills for more complicated problems.

The second task is the most important task for which the group is given roughly one month time: making the global optimisation algorithm. The group chooses to make two global optimisation algorithms, the first one being the basin hopping algorithm and the second one

a genetic algorithm. Two algorithms will be developed instead of one, because developing a single algorithm should take less than one month and the group is large enough to be split into two separate teams. Furthermore, the geometries obtained by running both global optimisation algorithms can be analysed by, for example, checking whether there are differences between the resulting structures. Does one algorithm perform better than the other and, if that is the case, why? Is one of the algorithms able to find different types of structures compared to the other? These are the types of questions that the group can try to answer. The end goal is finding stable, low-energy geometries in the PES and both global optimization algorithms should aid in efficiently finding those structures. Due to their simplicity, Lennard-Jones clusters will be used to test the algorithms. If needed, more complicated structures can be used by simply using a different model for calculating potential energies.

After the initial two algorithms have been created, the next task is to parallelise each algorithm with the goal of improving their performances. Not only is parallel code the standard for computers these days, but it will also make calculations noticeably faster. The group will get in contact with the supervisor to arrange a Raspberry Pi workshop where the group can get familiar with Raspberry Pis. Then, the pyMPI library will be used to implement the code run in parallel and make it ready to be used in high-performance computing (HPC) environments. To test that, the group will set up a small Raspberry Pi cluster and run example optimisation tasks on it.

Finally, a report will be delivered which contains the outcomes and achievements of the project.

## 5.2. ASE Package

The ASE library will help in solving the research question of this project. This section starts with a short introduction to the ASE library, followed by a description of the procedure of making initial structures of the clusters in ASE. Then, some algorithms for local and global optimization that are implemented in ASE are discussed.

### *Introduction to ASE*

The Atomic Simulation Environment (ASE) is a collection of Python modules intended to set up, control, visualise, and analyse atomic simulations. There is for example an Atom class in where information such as its position, its symbol, its mass and so on is stored. Running a simulation in ASE means calculating the forces and energies that rest in a cluster. That is done with so-called 'calculators': some atomic numbers and positions are inputted and it outputs the energy and forces. An example of a calculator is shown below.

```
>>> from ase import Atoms
>>> from ase.optimize import BFGS
>>> from ase.calculators.nwchem import NWchem
>>> from ase.io import write
h2 = Atoms('H2', positions = [[0, 0, 0], [0, 0,
    0.7]])
>>> opt = BFGS(h2)
>>> opt.run(fmax = 0.02)
BFGS:   0   19:10:49  -31.435229  2.2691
BFGS:   1   19:10:50  -31.490773  0.3740
BFGS:   2   19:10:50  -31.492791  0.0630
BFGS:   3   19:10:51  -31.492848  0.0023
>>> write('H2.xyz', h2)
>>> h2.get_potential_energy()
-31.492847800329216
```

Interactive session with calculator. Retrieved from Larsen et al. (2017), p. 4

The Atoms object plays a central role in ASE. It is a collection of all atoms and is thus used to create any atomic structure in ASE. These structures can be manipulated by rotations and translations on the individual atoms themselves.

The calculators also operate on the Atoms object. There are many kinds of calculators that all have different implementations and it is the programmer's job to assign calculators to their Atoms objects. Only then, energies and forces can be calculated. Apart from the calculators, there are also other algorithms that are provided by ASE for other tasks, such as (global) structure optimization. This is further described in one of the following sections.

### *Generating Clusters in ASE*

ASE provides two methods for generating atomic structures: one via the GUI and one via manual coding. Here, the focus will lie on coding the structures.

Making simple structures is straightforward. The structures can be created manually, but ASE also provides some databases which already have certain molecules ready. A code example of both scenarios is shown below:

```
from ase import Atoms
h2 = Atoms('H2', [(0, 0, 0), (0, 0, 0.74)])

from ase.build import molecule
water = molecule('H2O')
```

Creating structure of molecules. Retrieved from Larsen et al. (2017), p. 4

### *Local Structure Optimization in ASE*

Local structure optimization algorithms start from an initial guess for the atomic positions and use the forces acting on the atoms to find structures of lower energy in an iterative way until convergence. There are several methods available in ASE, which will be outlined below. All the algorithms can be found in the `ase.optimize` module.

- **BFGS:** uses the forces and the Hessian matrix to decide where to move the atoms on each step. Because the Hessian matrix is computationally expensive to calculate, it estimates the matrix by using an initial guess which is then corrected with every step. The involvement of the (approximate) Hessian matrix makes it a computationally heavy algorithm to perform and the initial guess at the start of the algorithm makes it sensitive to false starts.
- **L-BFGS:** a low-memory version of BFGS.
- **MDMin:** solves Newton's second law numerically and after each time step checks the dot product between the forces and the momenta. If it is zero, it passed through a local minimum. In general, this algorithm performs very well because it uses the laws of physics to optimize the cluster.
- **FIRE:** adds an additional force which steers the atoms gradually towards the direction of the steepest descent. The FIRE algorithm usually requires more iterations than BFGS to converge. That is due to the step size of the FIRE algorithm being smaller, which also decreases the cost per step since fewer calculations need to be made.

#### *Global Structure Optimization in ASE*

As mentioned above, the group will implement its own global optimisation algorithms throughout the further project. The ASE library provides several features that will assist in implementing the group's algorithms. One feature that ASE provides is a few global optimisation algorithms, such as Basin hopping and Minima hopping, which could be used to test the performance and efficiency of the algorithms made by the group for comparison. Another interesting feature that ASE provides is that it has libraries that make the parallelisation of the algorithms easier. For example, with 'PBSQueueRun' it is possible to make genetic algorithms that work together using queuing systems where each candidate is relaxed in a separate job.

## 6. Conclusion and Outlook

The purpose of this article was to review the main properties, characteristics and applications of clusters and to give an overview of existing global optimisation approaches.

It was found that clusters are a relatively new class of material building blocks, which can mainly be distinguished from bulk systems by their large surface to volume ratio. This results in unique optical, electric and magnetic properties, which make clusters applicable in many fields. Amongst others, clusters can be deposited onto electrodes, to drastically increase the performance of batteries. Furthermore, clusters show great potential in the field of catalysis and even data storage for flash memory. Finally, it was discussed that metal clusters are useful for medical imaging, drug delivery and that they could even aid in the prevention or treatment of several diseases.

Since the characteristics of clusters depend greatly on the exact number of atoms and their geometrical arrangement, being able to find energetically favourable configurations is key to predicting and manipulating clusters for their intended application. Two global optimisation approaches were addressed in detail.

The basin hopping algorithm is an unbiased approach to global optimisation. Due to each generation being randomly generated, the lowest energy configurations can be found with this algorithm. The basin hopping algorithm performs best for smaller clusters since the algorithm converges slower than the genetic algorithm due to its random nature.

The genetic algorithm is a commonly used method for global optimisation, because it can be applied to a wide variety of problems and generally seems to perform well. Since the algorithm uses a population of solutions, it is able to effectively search multiple areas of the search space at the same time. Numerous methods for combining and mutating clusters exist, which can improve the performance of the algorithm in different applications. Additionally, since these processes of combining and mutating clusters can be performed independently for each cluster, the genetic algorithm has a natural ability for being parallelised. For this purpose, two methods have been described which can provide roughly linear speedup in the execution time of the genetic algorithm.

Some additional algorithms, such as Particle Swarm Optimisation and Tabu Search in Descriptor Space, have also briefly been described. These algorithms have shown to have different strengths and weaknesses depending on the application.

This group will continue the project by implementing the described basin hopping and genetic algorithm in an attempt to propose enhancements to the existing implementations. The primary goal is to create a working code that can find global minima for very simplified and idealised situations. However, there is a multitude of options, nuances, and specific applications to explore, motivating more research in the future.

First of all, it would be interesting to investigate more accurate potential energy estimations while still minimising computational costs. As discussed above, this might be achieved by applying machine learning methods like neural networks to find the energy of large clusters. Moreover, neural networks might also be employed to determine the steps for global optimisation itself. If some pattern could be found in the global minima

for clusters of varying sizes, then the complexity of the search for new minima might be drastically reduced.

Another exciting idea is to investigate the possibility of adding constraints to the energy minimisation. When creating new materials, one might be looking for a set of very specific characteristics. Perhaps a certain symmetry or polarity is needed for the application of interest, or the shape might affect the cluster's biocompatibility and so on. In that case, the search is not for the absolute global minimum, but rather for the lowest local minimum with all the preset characteristics. This approach could lead to a systematic procedure for using clusters in new materials.

Finally, the objective of building ever more efficient algorithms is to be able to take more atoms and more scenarios into account. For instance, it seems intuitive that large bulk systems are more energetically favourable than clusters, but this does not necessarily mean that one large cluster is more stable than multiple smaller clusters. Considering more unlikely arrangements might lead to a better general understanding of complicated systems that are formed in nature.

From the few possible directions for further research mentioned above, it can be seen that there are still many areas within geometry optimisation that have room for additional improvements and discoveries. However, considering the amount of research effort being put into this field and the importance of the subject, the future of geometry optimisation seems to be very promising.

## Acknowledgements

We thank Dr. Dennis Palagin for the interesting discussions, valuable input and critical feedback that made this work possible.

## References

Averkiev, B. (2009). *Geometry and electronic structure of doped clusters via the Coalescence Kick method*. Utah State University. https://digitalcommons.usu.edu/etd/963

Binns, C. (2001). Nanoclusters deposited on surfaces. *Surface Science Reports*, *44*, 1–49. https://doi.org/10.1016/S0167-5729(01)00015-2

Busche, C., Vilà-Nadal, L., Yan, J., Miras, H. N., Long, D.-L., Georgiev, V. P., Asenov, A., Pedersen, R. H., Gadegaard, N., Mirza, M. M., Paul, D. J., Poblet, J. M., & Cronin, L. (2014). Design and fabrication of memory devices based on nanoscale polyoxometalate clusters. *Nature 2014 515:7528*, *515*, 545–549. https://doi.org/10.1038/nature13951

Cheng, J., & Fournier, R. (2004). Structural optimization of atomic clusters by tabu search in descriptor space. *Theoretical Chemistry Accounts*, *112*(1), 7–15.

Cheng, N., Stambula, S., Wang, D., Banis, M. N., Liu, J., Riese, A., Xiao, B., Li, R., Sham, T.-K., Liu, L.-M., Botton, G. A., & Sun, X. (2016). Platinum single-atom and cluster catalysis of the hydrogen evolution reaction. *Nature Communications 2016 7:1*, *7*, 1–9. https://doi.org/10.1038/ncomms13638

Choi, J. I., Abdelhafiz, A., Buntin, P., Vitale, A., Robertson, A. W., Warner, J., Jang, S. S., & Alamgir, F. M. (2019). Contiguous and atomically thin pt film with supra-bulk behavior through graphene-imposed epitaxy. *Advanced Functional Materials*, *29*. https://doi.org/10.1002/ADFM.201902274

Darby, S., Mortimer-Jones, T. V., Johnston, R. L., & Roberts, C. (2002). Theoretical study of cu–au nanoalloy clusters using a genetic algorithm. *The Journal of Chemical Physics*, *116*(4), 1536–1550. https://doi.org/10.1063/1.1429658

Das, S., Li, J., & Hui, R. (2014). Impact of electrode surface/volume ratio on li-ion battery performance. *Cosmol*. Accessed on 22 Oct 2021. https://www.comsol.nl/paper/impact-of-electrode-surface-volume-ratio-on-li-ion-battery-performance-18611

Deaven, D. M., & Ho, K. M. (1995). Molecular geometry optimization with a genetic algorithm. *Phys. Rev. Lett.*, *75*, 288–291. https://doi.org/10.1103/PhysRevLett.75.288

Díez, I., & Ras, R. H. A. (2011). Fluorescent silver nanoclusters. *Nanoscale*, *3*, 1963–1970. https://doi.org/10.1039/C1NR00006C

Echt, O. (1996). Convergence of cluster properties towards bulk behavior: How large is large? *Large Clusters of Atoms and Molecules*, (pp. 221–239). https://doi.org/10.1007/978-94-009-0211-4_8

Gao, G., Chen, R., He, M., Li, J., Wang, L., & Sun, T. (2019). Gold nanoclusters for parkinson's disease treatment. *Biomaterials*, *194*, 36–46. https://doi.org/10.1016/J.BIOMATERIALS.2018.12.013

Glover, F. (1990). Tabu search - part i. *INFORMS Journal on Computing*, *2*, 4–32.

Goedecker, S. (2004). Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. *The Journal of Chemical Physics*, *120*(21), 9911–9917. https://doi.org/10.1063/1.1724816

Heiles, S., & Johnston, R. L. (2013). Global optimization of clusters using electronic structure methods. *International Journal of Quantum Chemistry*, *113*, 2091–2109. https://doi.org/10.1002/QUA.24462

Huang, C., Yang, Z., Lee, K., & Chang, H. (2007). Synthesis of highly fluorescent gold nanoparticles for sensing mercury(ii). *Angewandte Chemie (International ed. in English)*, *46*, 6824–6828. https://doi.org/10.1002/ANIE.200700803

Johnston, R. (2003). Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries. *Dalton Transactions - DALTON TRANS*, *22*. https://doi.org/10.1039/b305686d

Johnston, R. L. (2002). Atomic and molecular clusters. *Atomic and Molecular Clusters*. https://doi.org/10.1201/9780367805814

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *Journal of Global Optimization*, *39*, 459–471. https://doi.org/10.1007/s10898-007-9149-x

Kunwar, P., Hassinen, J., Bautista, G., Ras, R. H., & Toivonen, J. (2014). Direct laser writing of photostable fluorescent silver nanoclusters in polymer films. *ACS Nano*, *8*, 11165–11171. https://doi.org/10.1021/NN5059503

Kunwar, P., Hassinen, J., Bautista, G., Ras, R. H. A., & Toivonen, J. (2016). Sub-micron scale patterning of fluorescent silver nanoclusters using low-power laser. *Scientific Reports*, *6*. https://doi.org/10.1038/SREP23998

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C., Jensen, P. B., Kermode, J., Kitchin, J. R., Kolsbjerg, E. L., Kubal, J., Kaasbjerg, K., Lysgaard, S., & ... (2017). The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, *29*, 273002. https://doi.org/10.1088/1361-648X/AA680E

Li, S., Wang, W., Liu, X., Zeng, X., Li, W., Tsubaki, N., & Yu, S. (2016). Nitrogen-doped graphene nanosheets as metal-free catalysts for dehydrogenation reaction of ethanol. *RSC Advances*, *6*, 13450–13455. https://doi.org/10.1039/C5RA20471B

Li, Z., & Scheraga, H. A. (1987). Monte carlo-minimization approach to the multiple-minima problem in protein folding. *84*(19), 6611–6615. https://doi.org/10.1073/pnas.84.19.6611

Lv, J., Wang, Y., Zhu, L., & Ma, Y. (2012). Particle-swarm structure prediction on clusters. *The Journal of Chemical Physics*, *137*(8), 084104. https://doi.org/10.1063/1.4746757

Manby, F., Johnston, R. L., & Roberts, C. (1998). Predatory genetic algorithms. *Match*, *38*, 111–122.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Mitra, A., Jana, G., Pal, R., Gaikwad, P., Sural, S., & Chattaraj, P. (2021). Determination of stable structure of a cluster using convolutional neural network and particle swarm optimization. *Theoretical Chemistry Accounts*, *140*. https://doi.org/10.1007/s00214-021-02726-z

Paar, A. (2014). Application note: Surface area determination of battery cathode and anode materials |. *Covalent Metrology Analytical Labs*. Accessed on 25 Oct 2021. https://covalentmetrology.com/surface-area-determination-of-battery-cathode-and-anode-materials/

Pakkala, A., & Putkonen, M. (2010). Atomic layer deposition. *Handbook of Deposition Technologies for Films and Coatings*, (pp. 364–391). https://doi.org/10.1016/B978-0-8155-2031-3.00008-9

Paleico, M. L., & Behler, J. (2020). Global optimization of copper clusters at the zno(101¯0) surface using a dft-based neural network potential and genetic algorithms. *The Journal of Chemical Physics*, *153*(5), 054704. https://doi.org/10.1063/5.0014876

Palmer, R. (1997). Welcome to the clusterworld. *NewScientist*. Accessed on 14 Oct 2021. https://www.newscientist.com/article/mg15320705-200-welcome-to-clusterworld/

Panosetti, C., Krautgasser, K., Palagin, D., Reuter, K., & Maurer, R. J. (2015). Global materials structure search with chemically motivated coordinates. *Nano Letters*, *15*, 8044–8048. https://doi.org/10.1021/ACS.NANOLETT.5B03388

Pinkard, A., Champsaur, A. M., & Roy, X. (2018). Molecular clusters: Nanoscale building blocks for solid-state materials. *Accounts of Chemical Research*, *51*, 919–929. https://doi.org/10.1021/ACS.ACCOUNTS.8B00016

Portman, N., & Tamblyn, I. (2017). Sampling algorithms for validation of supervised learning models for ising-like systems. *Journal of Computational Physics*, *350*, 871–890. https://doi.org/10.1016/j.jcp.2017.06.045

Rensburg, A. J. V., Schoor, G. V., & Vuuren, P. A. V. (2014). Active surface area approximation in a lead-acid cell for optimal performance in renewable energy systems. *IFAC Proceedings Volumes*, *47*, 9450–9455. https://doi.org/10.3182/20140824-6-ZA-1003.01642

Schlegel, H. B. (2003). Exploring potential energy surfaces for chemical reactions: an overview of some practical methods. *Journal of computational chemistry*, *24*(12), 1514—1527. https://doi.org/10.1002/jcc.10231

Shang, L., Dong, S., & Nienhaus, G. U. (2011). Ultra-small fluorescent metal nanoclusters: Synthesis and biological applications. *Nano Today*, *6*, 401–418. https://doi.org/10.1016/J.NANTOD.2011.06.004

Shaw, J., Zhong, Y.-W., Hughes, K. J., Hou, T.-H., Raza, H., Rajwade, S., Bellfy, J., Engstrom, J. R., Abruna, H. D., & Kan, E. C. (2011). Integration of self-assembled redox molecules in flash memory devices. *IEEE Transactions on Electron Devices*, *58*, 826–834. https://doi.org/10.1109/ted.2010.2097266

Sierka, M. (2010). Synergy between theory and experiment in structure resolution of low-dimensional oxides. *Progress in Surface Science*, *85*, 398–434. https://doi.org/10.1016/j.progsurf.2010.07.004

Solov'yov, I. A. (2021). Growth mechanisms and stability of atomic clusters | quantbiolab. *QuantBio*. Accessed on 18 Oct 2021. https://quantbiolab.com/research/growth-mechanisms-and-stability-of-atomic-clusters

von Helden, G., Hsu, M., Kemper, P. R., & Bowers, M. T. (1991). Structures of carbon cluster ions from 3 to 60 atoms: Linears to rings to fullerenes. *The Journal of Chemical Physics*, *95*(5), 3835–3837. https://doi.org/10.1063/1.460783

Wales, D. J., & Doye, J. P. K. (1997). Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110

atoms. *Journal of Physical Chemistry A*, *101*, 5111–5116. https://doi.org/10.1021/JP970984N

Wang, C., Li, J., Amatore, C., Chen, Y., Jiang, H., & Wang, X.-M. (2011). Gold nanoclusters and graphene nanocomposites for drug delivery and imaging of cancer cells. *Angewandte Chemie International Edition*, *50*, 11644–11648. https://doi.org/10.1002/ANIE.201105573

White, W. E., Bushey, A. H., Holtzclaw, H. F., & Hengeveld, F. W. (2007). Aluminum phosphide. *Inorganic Syntheses*, *4*, 23–25. https://doi.org/10.1002/9780470132357.CH7

Xie, J., Zheng, Y., & Ying, J. Y. (2010). Highly selective and ultrasensitive detection of hg2+ based on fluorescence quenching of au nanoclusters by hg2+–au+ interactions. *Chemical Communications*, *46*, 961–963. https://doi.org/10.1039/B920748A

Zhang, J., & Glezakou, V.-A. (2020). Global optimization of chemical cluster structures: Methods, applications, and challenges. *International Journal of Quantum Chemistry*, *121*. https://doi.org/10.1002/qua.26553

Zhang, X. H., Zhou, T. Y., & Chen, X. (2015). Applications of metal nanoclusters in environmental monitoring. *Chinese Journal of Analytical Chemistry*, *43*, 1296–1305. https://doi.org/10.1016/S1872-2040(15)60856-8

Zwanzig, R., Szabo, A., & Bagchi, B. (1992). Levinthal's paradox. *Proceedings of the National Academy of Sciences*, *89*(1), 20–22. https://doi.org/10.1073/pnas.89.1.20