A Genetic Algorithm for Crystal Structure Prediction

Nathan Luke Abraham

Submitted for the Degree of
Doctor of Philosophy

Department of Physics
University of York

December 2006

Abstract

In this thesis I will present a method of crystal structure prediction which is based upon Genetic Algorithms, which allows for truly *ab initio* crystal structure prediction. This method is outlined by presenting results on systems using the Lennard–Jones potential. An extension to this method is then derived which improves convergence to the minimum enthalpy structure using the Structure Factor in a method which allows the discrimination of structures as the calculation is progressing.

The technique is then extended to systems where the cell of the structures is not known, in the study of 4–atom carbon polymorphs using Density Functional Theory. This technique is then applied to study the high–pressure phases of the Dzugutov potential, finding three new phases that have not previously been reported. Future avenues of research and extensions to this method are then discussed.

Contents

List of Figures

List of Tables

List of Algorithms

Declarations

I declare that the work presented in this thesis, except where otherwise stated, is based on my own research and has not been submitted previously for a degree in this or any other university. Parts of the work reported in this thesis have been published in:

N.L. Abraham and M.I.J. Probert, "A Periodic Genetic Algorithm with Real-Space Representation for Crystal Structure and Polymorph Prediction", *Physical Review B* **73**, 224104

N.L. Abraham and M.I.J. Probert, "An Improved Genetic Algorithm for Polymorph Prediction using the Structure Factor", (in preparation)

N.L. Abraham and M.I.J. Probert, "A study on the High–Pressure Phases of the Dzugutov Potential using a Genetic Algorithm", (in preparation)

Signed

Nathan Luke Abraham

Acknowledgements

This thesis would not be possible without the help and support of my supervisor, Matt Probert, who has helped me developed the work presented here over the past three years. I would also like to thank Rex Godby who originally suggested the use of a periodic over a planar cut in a group seminar two years ago, Peter Main who suggested the Structure Factor as a suitable function for use in this work, and Mohammed Babiker for his support in writing letters of recommendation and giving extremely helpful comments on my first paper. Moira Penny, former Graduate Administrator and now Departmental Administrator, has also helped make this thesis possible.

I would also like to mention the invaluable assistance of the following members of this department who have helped me over the past three years by listening to my problems and making helpful suggestions. They are, in no particular order: Phil Hasnip, Sam Skipsey, Simon Horsley, Chris Eames, Nicola Booth, Josh Uretski, Martin Stankovski, Matthieu Verstraete, Hector Mera, Nuno Carneiro, Peter Bokes, Tony Patman, Maff Glover, Matt Farrow, Andy Carter, Tom Deakin, Stewart Stirk, Edmund Spracklen, Iain Hall, Paul Kent, Chris Bunce, Julia Linke, Mike Taylor, Jeremy Coe, Geoff Pert, Paddy Barr and Stephanie Miller.

There are also a number of people who are not members of the department that I would also like to thank for their personal support: My Parents, Ruth and Mark Abraham, my Grandmother Zena Lebow, my Aunt Bea Lebow and my cousin Zuli Lebow, along with Christine Wilson, Irene Walton, Lee Moorhead, Jon Ingold, Francesca Raphaely, Geneva Melzack, Tim Seagrave, Paddy Brinded, Simon Pickering, John Squires, Claire Shannon, Daniel Lizar, Tom Murphy, Tony Geard, Lizzie Hasnip, Cheryl Alexander, Marthe Tholen, Miranda Jackson and Esther Morcowitz.

I would also like to thank the members of Big Bad Voodoo Daddy and The Squirrel Nut Zippers, who helped me keep my sanity during the writing of this thesis. I'm certain that there are other people who I have forgotten, and so I would like to thank them as well.

“Use the Force, Luke.”

– Alec Guinness, *Star Wars (1977)*.

Chapter 1

Introduction

1.1 What is a Potential Energy Surface?

There are many problems for which it may be easy to determine a good solution, but in where it is difficult to find the *best* solution. What is the best configuration of atoms? What is the best route for an encyclopedia salesman to take between cities? What is the optimal configuration of transistors on a microchip? These problems will have many solutions that will exist, only one (or a small number) of which will be optimal. For each solution there will be a measure of how good that solution is. This measure can be used to define a potential energy, $V(\mathbf{r})$, of the system which is a function of the system co-ordinates. For a system of N atoms the *potential energy surface* is a $3N$–dimensional object embedded in a $3N+1$–dimensional space, where the extra dimension is the value of the potential energy at each point [Wales, 2003].

Genetic Algorithms fall into a class of optimisation algorithms that are designed for the *global* optimisation of systems. There are a large number of very efficient strategies for *locally* optimising a structure (some of which will be described in section 2.2). In local optimisation the system is "relaxed" until there are no net forces on any of the atoms. The resulting configuration will be a valid "solution" which exists in the space defined by the potential energy function, however there is no guarantee that it is the lowest energy (and hence the "best") configuration of all the possible configurations. Global optimisation methods are techniques in which the lowest energy solution can be found. Some methods are *deterministic* and will search the whole of the potential energy surface exhaustively, and others are *stochastic* and search in a random manner. Genetic Algorithms fall into this latter category – although they retain a memory of previous parts of the potential energy surface that have already been searched.

1.2 Application to Crystallography

Maddox [1988] stated that "one of the continuing scandals in the physical sciences is that it remains in general impossible to predict the structure of even the simplest crystalline solids from a knowledge of their chemical composition". Eighteen years later the situation is much the same. While there are many efficient techniques that can be used to relax a configuration of atoms, there is no guarantee that this is the most stable configuration. A systematic search of the potential energy surface in not feasible in the $3N$–dimensional energy landscape for more than a handful of atoms.

For larger systems, a stochastic approach is useful. Simulated Annealing methods (see section A.2) are very good for small systems, involving the determination of the minimum energy configuration of atomic clusters. However, simulated annealing rapidly becomes unfeasible as the system size increases (see section A.2.2). Comparable results have been shown in similar systems for Genetic Algorithms (see section A.3), but in this technique as the system size increased the method was still able to determine the global minimum solution (see section A.3.1). If this method gives good results for non–periodic atomic systems, perhaps it would show similar scaling and results in periodic systems as well.

Chapter 2 describes the work done on solid state systems using Genetic Algorithms and chapter 3 outlines some different techniques for calculating the energy of the system. After the basics of these previous Genetic Algorithm methods have been explained the new method developed that is suitable for use on periodic systems will be outlined in chapter 4. The work described in chapter 6 will explain how this method can be applied to systems where the cell of the crystal of interest is unknown, and apply this method within an *ab initio* framework. I will also explain some subtleties that need to be considered when using Genetic Algorithms in this first principles approach.

Following on directly from the work in chapter 4, I will derive in, chapter 5, an expression that can aid the differentiation of structures during the course of a calculation. I have also applied this method to search for phases of materials at different pressures and the results of these calculations can be seen in chapter 7. I will make conclusions on the technique that I have developed in chapter 8, and in chapter 9 I will describe some ways in which the technique could be extended. In appendix A I present results I obtained early on in my PhD when I compared the Simulated Annealing method with Genetic Algorithms. The contents of appendix B are left to the reader to digest at their own leisure.

Chapter 2

The State of the Art

2.1 Introduction

Before I can describe what developments to genetic algorithm (GA) techniques have been made in the field of solid–state physics and chemistry, I shall first describe what a genetic algorithm is, both as it has been developed historically and in its application to solid–state systems.

I will then describe the previous advances made in this area, and the systems that have been studied by these techniques. Following this, I will also outline some other global optimisation techniques that have been utilised to study similar problems to which genetic algorithms have also been applied. Due to some computational issues that must be addressed when performing calculations using this method I must also outline the basics of density functional theory (DFT).

Firstly however, I will describe some methods of local optimisation. Some GA techniques require a local optimiser to be used within the GA framework.

2.2 Local Optimisation

The potential energy surface can be thought of as a multi–dimensional system of hills and valleys with saddle points connecting them. If you placed a ball on one of the slopes at random it would roll downhill until it reached the bottom of a valley. This valley is the "local basin of attraction" for the ball on the potential energy surface. A direct minimiser will minimise the energy of the structure to the minimum of the local basin of attraction, which may or may not be the global minimum (see figure 2.1).

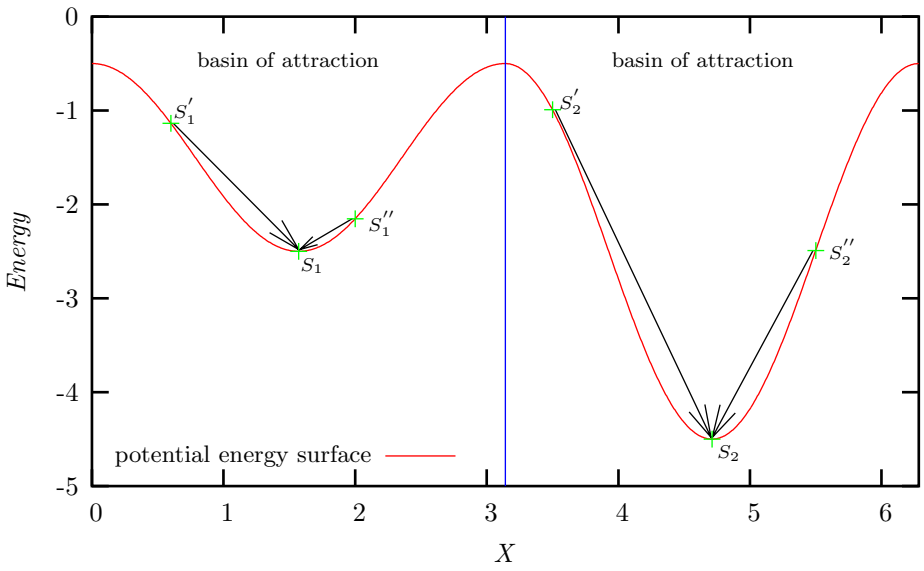

Figure 2.1: A 2–Dimensional representation of a potential energy surface, with two basins having minimum energy solutions of S_1 and S_2. Four initial structures exist on the PES, S_1' and S_1'' (which minimise to S_1), and S_2' and S_2'' (which minimise to S_2). After minimisation this reduces the number of structures to two.

2.2.1 Line Minimisation – Golden Section Search

The line minimisation step is very important in local minimisation techniques. The local minimisation step chooses the search direction (different methods choose this direction more intelligently than others), but the line minimiser finds the minimum in the direction chosen. I will describe the golden section search algorithm Polak [1971], which uses the Fibonacci fractions to improve efficiency over the similar method of bisection search. There are other line minimisation methods that can be used.

The procedure is described in Algorithm 1. The energy tolerance ε and the end–point of the search ρ must be supplied from the local minimiser. This

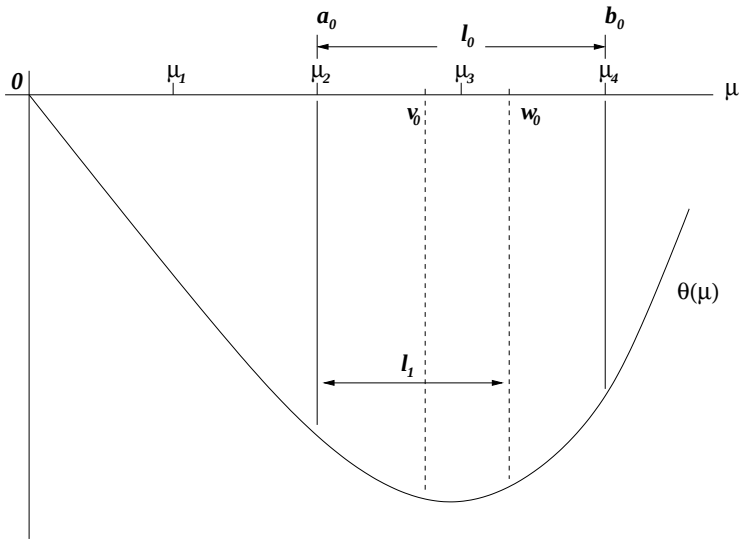

Figure 2.2: Golden Section search

method makes use of two important parameters, $F_1 = (3-\sqrt{5})/2 \approx 0.38$, and $F_2 = (\sqrt{5}-1)/2 \approx 0.618$. These values of these numbers are chosen such that they give F_1 and F_2 the properties that $F_2 = 1 - F_1$ and $F_1 = (F_2)^2$. The value $\overline{\mu}$ is returned, which minimises the function $\theta(\mu) = f(\mu + \lambda h(\mu)) - f(\mu)$, where $h(\mu)$ is the search direction (see figure 2.2 for a graphical description).

2.2.2 The Method of Steepest Descents

One of the simplest direct minimisation techniques is steepest descents. The minimisation starts at the initial configuration $\mathbf{x}_{(0)}$, and a step direction is chosen in which $f\left(\mathbf{x}_{(i)}\right)$ decreases the quickest, which is opposite to $\nabla f\left(\mathbf{x}_{(i)}\right)$. The function in this direction is minimised by a line minimisation search (see previous section), and then $-\nabla f\left(\mathbf{x}_{(i)}\right)$ is recalculated and the function is again minimised in the search direction. This continues until (by a convergence criterion being reached, usually on the change in energy of the system, or a change in forces on the system) it is decided that the function is at the bottom of the basin of attraction.

2.2.3 The Conjugate Gradient Algorithm

The method of conjugate gradients is a direct minimisation method which chooses the search direction more intelligently than steepest descents. I will describe the Polak–Ribière method as described in Polak [1971] (as outlined in Algorithm 2).

Algorithm 1 Golden Section Search

1: calculate $\theta(\rho)$ and $\theta(0)$
2: **if** $\theta(\rho) \geq \theta(0)$ **then**
3: set $a_0 = 0$, $b_0 = \rho$ and go to step 13
4: **end if**
5: set $i = 0$, $\mu_0 = 0$
6: set $\mu_{i+1} = \mu_i + \rho$
7: calculate $\theta(\mu_{i+1})$
8: **if** $\theta(\mu_{i+1}) \geq \theta(\mu_i)$ **then**
9: set $a_0 = \mu_{i-1}$, $b_0 = \mu_{i+1}$ and go to step 13
10: **else**
11: set $i = i + 1$ and go to step 6
12: **end if**
13: set $j = 0$
14: set $l_j = (b_j - a_j)$
15: **if** $l_j \leq \varepsilon$ **then**
16: go to step 24
17: **end if**
18: set $v_j = a_j + F_1 l_j$, $w_j = a_j + F_2 l_j$
19: **if** $\theta(v_j) < \theta(w_j)$ **then**
20: set $a_{j+1} = a_j$, set $b_{j+1} = w_j$, set $j = j + 1$ and go to step 14
21: **else**
22: set $a_{j+1} = v_j$, set $b_{j+1} = b_j$, set $j = j + 1$ and go to step 14
23: **end if**
24: set $\overline{\mu} = (a_j + b_j)/2$ and STOP

This method may need restarting when the method has difficulty choosing a new direction. If so, setting $\gamma_i = 0$ will achieve this, and is the equivalent of performing one steepest descent step. It should be noted that another similar method, the Fletcher–Reeves conjugate gradient method, is identical to the Polak–Ribière method, except γ_i has the form

$$\gamma_i = \frac{\langle g_{i+1}, g_{i+1} \rangle}{\langle g_i, g_i \rangle}.$$

where the $\langle a_i, b_i \rangle$ notation describes taking the dot–product between vectors **a** and **b**.

2.2.4 The BFGS Algorithm

The method that shall be used in this study is that of Pfrommer *et al*. [1997] which has advantages over other methods because the cell symmetry is preserved, whilst allowing the relaxation of both the atomic co–ordinates and the cell shape. This is an extension to the original BFGS algorithm.

Algorithm 2 Polak–Ribière Conjugate Gradient Method

1: select a $z_0 \in \mathbb{R}^n$
2: **if** $\nabla f(z_0) = 0$ **then**
3: STOP
4: **end if**
5: set $i = 0$ and set $g_0 = h_0 = -\nabla f(z_0)$
6: calculate $\lambda_i > 0$ such that

$$f(z_i + \lambda_i h_i) = \min\{ f(z_i + \lambda h_i) \mid \lambda \geq 0 \}$$

using a line minimisation technique such as described in section 2.2.1.
7: set

$$z_{i+1} = z_i + \lambda_i h_i$$

8: calculate $\nabla f(z_{i+1})$
9: **if** $\nabla f(z_{i+1}) = 0$ **then**
10: STOP
11: **else**
12: set

$$g_{i+1} = -\nabla f(z_{i+1})$$
$$\gamma_i = \frac{\langle g_{i+1} - g_i, g_{i+1} \rangle}{\langle g_i, g_i \rangle}$$
$$h_{i+1} = g_{i+1} + \gamma_i h_i$$

then set $i = i + 1$ and go to step 6
13: **end if**

The BFGS algorithm is a quasi–Newton method that accumulates information about $\underline{\underline{H}}$, the inverse of the Hessian matrix $\underline{\underline{A}}$, and uses this information to determine the search direction on the potential energy surface.

The full state of the system is defined by the matrix of lattice vectors, $\underline{\underline{h}} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$ and the fractional co–ordinates of the atoms in the cell, s_i, $i = 1, \ldots, N$ which are defined in terms of the $\underline{\underline{h}}$ matrix. The volume of the cell is $\Omega = \det\left(\underline{\underline{h}}\right)$. Pfrommer *et al*. [1997] chooses the finite strain tensor $\underline{\underline{\varepsilon}}$ as a free variable instead of $\underline{\underline{h}}$. It stretches a reference configuration $\underline{\underline{h}}_0$ to $\underline{\underline{h}} = \left(\mathbb{1} + \underline{\underline{\varepsilon}}\right) \underline{\underline{h}}_0$. This method allows $\underline{\underline{\varepsilon}}$ to be asymmetric.

With an applied pressure, p, it is the enthalpy, $\mathcal{H} = E + p\Omega$, in the $(3N+9)$–dimensional space which we are attempting to optimise $\left(\text{where } \mathcal{H} = \mathcal{H}\left(\underline{\underline{\varepsilon}}, s_i, \ldots, s_N\right)\right)$. A point in configuration space is given by a column vector X where the first nine components are the strain tensor, and the rest are the fractional atomic co-ordinates. The force vector of the system is defined as

$$F = -\left.\frac{\partial \mathcal{H}}{\partial X}\right|_p \tag{2.1}$$

The first nine components in F from equation 2.1 are

$$f^{(\varepsilon)} = -\left(\sigma + p\Omega\right)\left(\mathbb{1} + \underline{\underline{\varepsilon}}^{\mathrm{T}}\right)^{-1} \tag{2.2}$$

where

$$\sigma = \left(\frac{\partial E\left(\left(\mathbb{1} + \underline{\underline{\varepsilon}}'\right)\underline{\underline{h}}\right)}{\partial \underline{\underline{\varepsilon}}'}\right)_{\underline{\underline{\varepsilon}}'=0} \tag{2.3}$$

The other $3N$ components of F are obtained by multiplying the forces on the atoms, $f_1, \ldots, f_N$, with the metric tensor $\underline{\underline{g}} = \underline{\underline{h}}^{\mathrm{T}}\underline{\underline{h}}$ so that

$$F = \left(f^{(\varepsilon)}, \underline{\underline{g}} f_1, \ldots, \underline{\underline{g}} f_N\right) \tag{2.4}$$

If we are sufficiently close to the minimum X_{min} then the change in enthalpy can be approximated by

$$\delta\mathcal{H} = \frac{1}{2}\left(X - X_{min}\right)^{\mathrm{T}} \cdot \underline{\underline{A}} \cdot \left(X - X_{min}\right) \tag{2.5}$$

If we were close to X_{min} then the knowledge of the Hessian matrix, $\underline{\underline{A}}$ would mean that the exact minimum of the local basin of attraction would be found in one step. However $\underline{\underline{A}}$ is unknown, and the basis of quasi–Newton schemes is to start with an initial guess for $\underline{\underline{A}}$ and improve on this guess as knowledge of the potential energy surface is determined as the algorithm proceeds. To aid computational matters it is in fact the inverse of the Hessian matrix that is approximated, $\underline{\underline{H}} = \underline{\underline{A}}^{-1}$. The update step is

$$X_{i+1} = X_i + \lambda \Delta X_i \tag{2.6}$$

$$\Delta X_i = \underline{\underline{H}}_i F_i \tag{2.7}$$

where F_i is F evaluated at X_i and λ is the step length along the step direction ΔX_i that could have been determined by any line minimiser such as that described in section 2.2.1.

The BFGS scheme only requires an initial guess, $\underline{\underline{H}}_0$, for $\underline{\underline{H}}$ and updates this guess by

$$\begin{aligned}\underline{\underline{H}}_i =& \underline{\underline{H}}_{i-1} - \frac{(X_i - X_{i-1}) \otimes (X_i - X_{i-1})}{(X_i - X_{i-1}) \cdot (F_i - F_{i-1})} \\ & - \frac{\left(\underline{\underline{H}}_{i-1}(F_i - F_{i-1})\right) \otimes \left(\underline{\underline{H}}_{i-1}(F_i - F_{i-1})\right)}{(F_i - F_{i-1}) \cdot \underline{\underline{H}}_{i-1}(F_i - F_{i-1})} \\ & + \left[(F_i - F_{i-1}) \cdot \underline{\underline{H}}_{i-1}(F_i - F_{i-1})\right] U \otimes U \end{aligned} \tag{2.8}$$

where

$$U = \frac{(X_i - X_{i-1})}{(X_i - X_{i-1}) \cdot (F_i - F_{i-1})} - \frac{\underline{\underline{H}}_{i-1}(F_i - F_{i-1})}{(F_i - F_{i-1}) \cdot \underline{\underline{H}}_{i-1}(F_i - F_{i-1})} \tag{2.9}$$

If the enthalpy was perfectly quadratic in $X - X_{min}$ then $\underline{\underline{H}} \rightarrow \underline{\underline{A}}^{-1}$ after the number of steps is the same as the number of degrees of freedom in the system. Pfrommer *et al*. [1997] provides details as to how to initialise $\underline{\underline{H}}_0$. There is a debate as to the advantage of conjugate gradient methods (section 2.2.3) over quasi–Newton methods (a disadvantage is the amount of storage required in quasi–Newton methods). Quasi–Newton methods have been around for longer and, as such, are more developed.

Algorithm 3 The BFGS algorithm

1: Initialise Hessian Matrix $\underline{\underline{H}}_0$
2: Evaluate forces F_0 of initial positions X_0 from equation 2.4
3: Update positions using equations 2.6 and 2.7 and line minimisation procedure such as one described in section 2.2.1
4: Evaluate forces on new positions X_i
5: Update the Hessian matrix $\underline{\underline{H}}_i$ using equation 2.8
6: Continue until a convergence criterion is reached. If not, goto step 3.

2.3 Genetic Algorithms

Genetic Algorithms (GAs) were first suggested by John H. Holland in his book *Adaption in Natural and Artificial Systems* [Holland, 1992], first published in 1975. He had noticed that simple representations (in his case, bit strings) could encode complicated structures, and that simple transformations could improve these structures. A basic outline of a genetic algorithm is shown in figure 2.3, where "crossover" defines performing a mating operation.

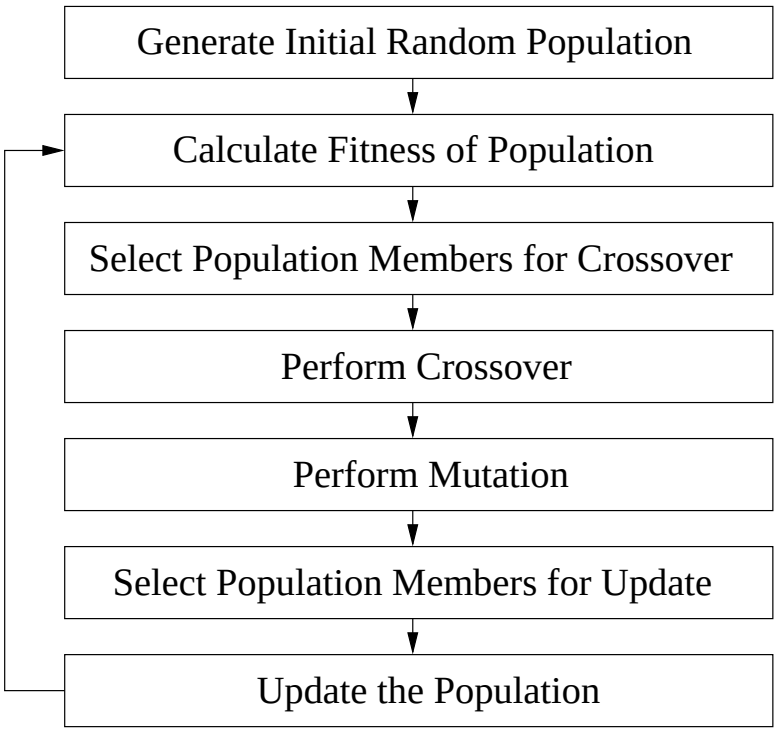

Figure 2.3: A basic outline of a genetic algorithm. One passage though the loop represents one generation.

It is a stochastic global optimisation method based on "survival of the fittest". This is a computational technique which is used to solve problems in which there are many potential solutions, only one of which is optimal.

A GA has five main parts, *representation*, *fitness*, *selection*, *crossover* and *mutation*. The method proceeds through *generations* until it is determined that the global minimum has been reached. The size of the population is usually fixed, and each member of the population contains a complete set of parameters for the function being searched.

A set of candidate solutions to the problem being studied are generated *at random* and are grouped together into a *population*, with each solution being a *member* of this population. This randomness is important, since we do not wish to bias the solution in any way, and it is also this randomness that makes

this method a truly *ab initio* search technique.

There needs to be a way of determining the *fitness* of each member, i.e. some way of telling which members are better or worse solutions to the problem than the other members. It is this fitness function which defines the problem. For some problems the fitness function is quite complicated, but for others it can be quite a simple form. This function may have some form of absolute measure of the fitness of a member, but usually in solid–state physics problems it is a relative measure between all the members of the population.

Population members will be chosen or *selected*, based on their fitness, to become *parents* to produce *offspring* in a breeding procedure known as *crossover*. Crossover involves creating one or more offspring which are a combination of features from their parents. The selection process is quite important in preventing the system *stagnating* through the generations.

Stagnation occurs when one good but not optimal structure will be found which biases the system away from the global minimum solution and the population becomes "trapped" in this local minima. I will discuss solutions to prevent stagnation in more detail in chapter 5. The form of the fitness function is important in this, but the way that members are selected is also important and will be discussed in more detail in section 2.3.3.

Each population member needs to be *encoded* or *represented* in some way such that crossover can be performed in a systematic way. The form of this representation is very important since it can affect how stable salient features of members are during the crossover procedure. A representation that is most efficient for crossover steps may be less efficient when considering fitness calculation. Representation will be discussed further in section 2.3.1.

The offspring may be *mutated* after crossover, which involves making changes to offspring in a random way which could introduce new, and possibly beneficial, aspects into the population.

Using each member's fitness the population is *updated* by only allowing some population members to survive into the next generation; the rest of the population members will be discarded. After this step the process begins again. The system evolves as new members are created and bred. While there is no guarantee of reaching the global minimum solution, good local minimum solutions should be found, and there are some tricks that can be played to aid convergence to the minimum energy solution which shall be discussed later in chapter 5.

While I describe the methodology of GAs I will describe the general binary (bit–string represented) approach, and also the approach as applied to real

space systems, which is my area of interest. There are not enough pages allowed for this thesis to do a full study of the minutiae involved in genetic algorithms in particular, or evolutionary algorithms in general, so I will focus on how genetic algorithms have been used in solid–state problems, in particular to show the logical progression of concepts that resulted in the technique that was developed for this thesis.

2.3.1 Representation

Traditionally, the values of GA population members have been represented by bit–strings directly, although this is no longer necessary for all systems, such as the solid–state physics systems that I shall describe. This representation method is analogous to that of DNA, and gives many advantages in terms of crossover and mutation (which shall be discussed below). However, it is often complicated to decode these strings to determine the fitness of the structure.

A bit string representation is very simple; strings of ones and zeroes would be randomly generated, e.g. 11010011, 01011001 etc., and these would form the initial population. The strings may be of fixed length or, more rarely, be of variable length [Harvey, 1992]. This string is also termed a *chromosome* with each individual bit also being called a *gene*.

A different representation system, a real–space approach suggested by Deaven and Ho [1995], was applied to clusters of atoms. In this method, the members were defined by the Cartesian co–ordinates of their atoms. This approach made fitness calculation very straightforward, as described below, but made crossover much more difficult. When using the real–space representation, a population member would be initially generated by randomly placing the required atoms inside a given volume. The system would then be locally minimised so that the structure is relaxed. Wales and Doye [1997] showed that working within this structurally relaxed space simplifies the problem by simplifying the search space without changing the relative positions of the global minima, or the relative positions of the local minima, and the search method proceeds by moving from one basin of attraction to another.

2.3.2 Fitness

This step evaluates the function being searched, and assigns a measure of how "fit" the individual member of the population is, upon which selection for reproduction is based. There are a large number of ways of scaling the fitness function to attempt to prevent early good (but not global) structures from

dominating.

The fitness is entirely dependent on the system being studied. In the real–space approach of Deaven and Ho [1995] the fitness was purely determined by the relative energy of the cluster, the fittest member having the lowest energy in that generation. This meant that the fitness of each member would need to be re–calculated in each generation and the population evolved.

There is an excellent review article by Johnston [2003] which covers the field of atomic cluster optimisation using GAs. I will give a brief overview of his approach.

Since the global minimum structure (and hence the global minimum energy) is not known (this is what the algorithm will be searching for), dynamic scaling of the fitness is required. Any cluster with a potential energy ≥ 0 will have its fitness set to zero. All potential energies given are therefore negative, and the more negative the value, the fitter the population member. A normalised value of the energy is used in the fitness calculations:

$$\rho_i = \frac{V_i - V_{min}}{V_{max} - V_{min}} \tag{2.10}$$

It should be noted that in this equation it doesn't matter if V_{max} is positive or negative, however setting these positive potential energy values to zero reduces the likelihood of the corresponding population member being selected for crossover or update.

There are a number of different scaling functions that can then be used to map this normalised energy value to fitness, three possible forms are exponential, linear and hyperbolic tangent, as suggested by Johnston [2003]. The fitness of each population member, f_i, can then be calculated:

$$\textbf{Exponential:} \quad f_i = \exp(-3\rho_i) \tag{2.11a}$$

$$\textbf{Linear:} \quad f_i = 1 - 0.7\rho_i \tag{2.11b}$$

$$\textbf{Hyperbolic Tangent:} \quad f_i = \frac{[1 - \tanh(2\rho_i - 1)]}{2}. \tag{2.11c}$$

A "better" structure (i.e. one with a low energy) will have a ρ_i close to 0. When its fitness is calculated it will have f_i close to 1. Conversely a high energy (or "bad") structure will have f_i close to 0. The different schemes dictate how much emphasis is placed on very fit or very unfit structures.

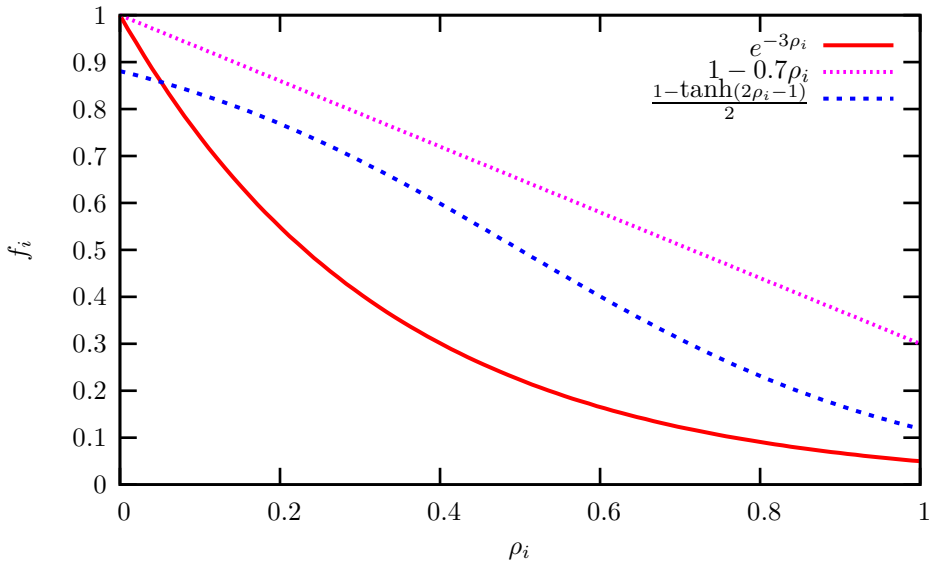

Figure 2.4: Comparison of the three different fitness schemes mentioned

As can be seen from figure 2.4 these scaling functions give different weight to different members of the population. The exponential function will bias strongly towards members that have a fitness close to the lowest in that generation. Linear scaling has a nice regular fall–off, and hyperbolic–tangential scaling weights the lowest section of members more equally, and so increases the spread of energies of members chosen, so increasing the probability that less fit members would proceed though to the next generation. This may help prevent stagnation.

2.3.3 Selection

The members of the population are selected for reproduction or update based on their fitness. Again, there are a large number of methods of selection that vary in complexity. A method with low selectivity accepts a large number of solutions, while high selectivity will allow a few or even one to dominate. However, a balance needs to be reached to try to prevent the solution from becoming trapped in a local minimum.

The simplest method of selecting members is through *elitist* selection. The "super–population" of all parents and offspring are sorted by fitness and then the fittest half is chosen, all of which would then become parents in the next generation. This process may lead very quickly to stagnation.

One simple selection method that should help prevent stagnation is *roulette* selection: the probability of selection is now proportional to fitness. The probability of an individual i being selected from its fitness f_i is

$$P_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{2.12}$$

A uniform random number would then be generated in the range 0 to 1. If this number is between the cumulative probabilities of the i^{th} and $(i+1)^{th}$ individuals, then the i^{th} individual is selected. This means that even though the fitter members are more likely to be selected, less fit members can still make it into the next generation. There is no guarantee that the fittest member will proceed through to the next generation using this method.

There are further selection schemes that can be used, such as *tournament* selection. In this case the population is broken up by randomly picking members whose fitnesses are directly compared, the winners from each group being selected for crossover.

There is also a *hybrid* scheme which involves picking m_{fit} members by elitist selection, and the remaining $M - m_{fit}$ members by roulette selection, where M is the size of the population. This method can be used to ensure that the fittest member will proceed to the next generation while keeping the benefits of roulette wheel selection.

2.3.3.1 Selection for Reproduction

The above sections describe general methods for selecting a number of members that can be used for a number of purposes. If you are selecting members for reproduction then you would usually be picking only two members for each crossover procedure, and would want a range of different parents to better explore the potential energy surface, so selection using the roulette method (or similar) is to be preferred. The choice of fitness function is also important in this step. There is no requirement to use the same selection method or fitness function in selection for crossover and selection for update.

2.3.4 Crossover

Crossover in the binary string formulation is very simple, two parents, 11010011 and **01011001** may be split in half and recombined to make two new offspring 1101**1001** and **0101**0011 (where the boldness of the text is merely an aid to the eye). They may be split at any point on the string, or may be split in a number of places. This form of crossover is very quick to calculate.

2.3.4.1 The Schema Theorem

The Schema Theorem [Holland, 1992] goes some way to explain why GAs are such a powerful technique. It demonstrates how crossover (and to an extent, mutation) forces the system to find robust and fit solutions in a binary representation.

A string can be thought of as being made up of building blocks, or "sub–strings". In this bit–string, a *schema* (plural schemata) is defined as the subset of strings with similarities over a given number of positions, and a *similarity template* is a string taken over the underlying string alphabet together with a wild–card character (e.g. $*$) that matches any of the string characters.

So, the similarity template $000**$ has schema 00000, 00001, 00010 and 00011. In general for binary strings of length l there are 2^l unique strings and 3^l schemata.

The number of fixed positions, H, in any schema is known as its *order*, $o(H)$, so $o(*011*1*) = 4$, $o(****1**) = 1$ etc.

If we have a string A=0111000, the schema $H_1 = *1****0$ and $H_2 = ***10**$ are represented within A. However, if a random crossover were to take place between positions 3 and 4 (defined by the notation |), such that

$$\begin{aligned} A &= 0\,1\,1\,|\,1\,0\,0\,0 \\ H_1 &= *\,1\,*\,|\,*\,*\,*\,0 \\ H_2 &= *\,*\,*\,|\,1\,0\,* \,* \end{aligned}$$

In this case H_1 would have been destroyed in the crossover step, while H_2 will have survived.

The defining length, $\delta(H)$ is defined as the position of the last fixed bit minus the position of the first fixed bit, i.e. $\delta(H_1)$=5 and $\delta(H_2)$=1. We can use the defining length to determine the probability of any schema being destroyed in crossover

$$P_d = \frac{\delta(H)}{(l-1)} \tag{2.13}$$

From the example above, $P_d(H_1) = 5/6$ and $P_d(H_2) = 1/6$. The survival rating will then be $P_s = 1 - P_d$, or, if the crossover itself is made by random choice, with probability P_c, then

$$P_s \geq \frac{1 - P_c \cdot \delta(H)}{(l-1)} \tag{2.14}$$

where the "$\geq$" symbol takes into account the fact that H could be re–created after crossover from components of both parents.

If we consider that at a given generation, t, there are $m(H,t)$ examples of schema H in the population $\mathbf{A}(t)$, we known the probability of any string A_i being selected from (2.12), so we can calculate the number of representatives expected in generation $(t+1)$

$$m(H,t+1) = \frac{m(H,t) \cdot N \cdot f(H)}{\sum f_j} \tag{2.15}$$

where $f(H)$ is the average fitness of the strings representing H in generation t. If the average fitness of the entire population is $\overline{f} = \sum f_j / N$ then the above equation can be simplified

$$m(H,t+1) = \frac{m(H,t) \cdot f(H)}{\overline{f}} \tag{2.16}$$

i.e. a schema grows as the ratio of its fitness to the average fitness of the population. If a schema has a fitness greater than the average fitness of the population it will have more samples in the next generation. If its fitness value is below the population average then it will receive a decreasing number of samples. Above–average schema will propagate and below–average ones will die off.

2.3.4.2 The Hamming Cliff and Gray Codes

Binary codes have a number of disadvantages. One of these is the so–called *Hamming cliff* named for mathematician Richard W. Hamming.

Say we have two binary numbers, $A = 011$ and $B = 100$ (with decimal values 3 and 4 respectively). When considering the use of binary strings in a genetic algorithm the values of A and B may represent two solutions that are quite similar – B could be the global minimum, and A could be a local minimum that is very close in the potential energy landscape to this global minimum. However, there is no way in the crossover or mutation steps to move between these minima. A mutation on the most significant bit of A would turn 3 into 7, while a mutation on B would turn 4 into 0. In this last case, the value has fallen off the Hamming cliff [Hayes, 2001].

One solution is to use Gray codes, named after Frank Gray of Bell Labs but

have been used since Emile Baudot, an early pioneer of French telegraphy [Hayes, 2001]. In this coding system, which is still of a binary nature, the Gray–encoded numbers differ by only one bit in the sequence (see Table 2.1).

In this way 7 can also cycle back to 0 again. This method of counting is slightly counter–intuitive, but Gray codes are easy to set up for a particular size. When combined with the Schema Theorem (section 2.3.4.1) Gray codes seem to be a more efficient way of encoding most problems in a genetic algorithm context given the perils associated with the Hamming cliff.

2.3.4.3 Real–Space Crossover

However, the binary–string (or even Gray–encoded string) method of encoding and crossover is not practical for use with continuous variables such as the spatial positions used in the atomic cluster problem. In this case there may be some slight numerical error in position which, while small, when the system is switched from real–space co-ordinates to a binary–encoded system for crossover may become larger. Binary–encoded strings also have a finite resolution for real space problems. One solution is to fix the atoms onto grid–points, and these points have an absolute value in the binary–encoded system. However, fixing atoms to grid–points could restrict the system in a number of ways. It may impose certain symmetries which will put an unintended bias into the system. It also reduces the accuracy in which the positions of the atoms are defined. While these real numbers are eventually binary encoded for storage purposes, this is at the machine precision, rather than a precision defined by the algorithm.

A solution to these issues was proposed by Deaven and Ho [1995], and is a much more "physical" way of dealing with the problem. The clusters are represented by an array of the atomic position vectors in *xyz* format, and crossover is done by taking a random plane through the centre of mass of the two parent

Number	Binary	Gray
0	000	000
1	001	001
2	010	011
3	011	010
4	100	110
5	101	111
6	110	101
7	111	100

Table 2.1: Comparison of binary and Gray codes.

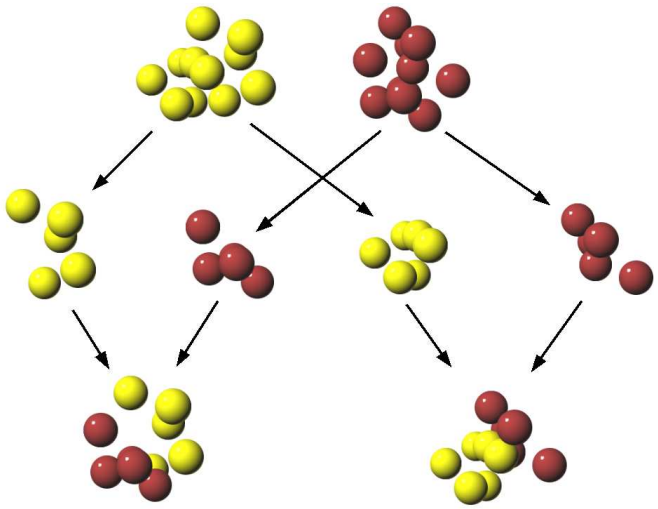

Figure 2.5: Crossover

clusters, and then swapping the halves. This is represented diagrammatically in figure 2.5.

The simplest way of defining this plane is to generate a random unit vector on the surface of a sphere [see, for example, Allen and Tildesley, 1987, appendix G.4, p349.]; this vector defines the normal to the plane, which always passes through the origin. If the centre of mass of the cluster is the same as the origin of the atomic position vectors, taking the dot product of the position vectors with the unit sphere vector will yield a scalar number α. If α is positive then the atom is defined as being above the plane, if α is negative then the atom is defined as being below the plane. The number of atoms in the offspring clusters must be the same as those in the parent clusters, so the plane may have to be moved to conserve atom number, the number of atoms in each half being recalculated each time the plane is moved.

There is a problem with this method as it stands. When one cluster is bred with another cluster, two or more atoms may find themselves very close to each other, and as such the offspring might have a large potential energy. A solution to this is to minimise every new structure that is generated *at any stage* so that it is now at the bottom of its local minimum, and as such all crossovers and mutations would produce viable structures [Wales and Doye, 1997; Johnston, 2003].

This method can be compared to the Schema Theorem. Due to the fact that the cut is being made in real–space, it is less likely to disrupt the structure of the population member, thereby allowing the local structure to be maintained during crossover, but also allowing good localised structure to be shared between different members.

2.3.5 Mutation

In the binary string case mutation usually involves changing a small percentage of the bits on the string. This can actually have quite a large effect – if the most significant bit is mutated it could affect the member a great deal. Mutation rates for binary strings are quite low because of this, typically $0.07\% \rightarrow 1\%$.

Mutation can be thought of as being either *static*, where the value of the quantity being mutated is assigned a completely new value independently (i.e. the new value of the bit could either be 0 or 1, with no reference to the old value), or *dynamic*, where the original value is altered by a random amount (i.e. if the bit were a 0 it would become a 1 and *vice versa*) [Johnston, 2003]. Mutation is also important in preventing stagnation by introducing new features into the population that may have a beneficial effect.

Mutation in a real–space atomic cluster sense involves the random changing of some, or all atomic positions. For each population member a random number is generated and is compared against the mutation rate (which, in this framework, are typically on the order of 0.1, i.e. 10%). If the random number is less than the mutation rate then that population member is mutated. Two possible mutations are:

- A reflection of one atom through the origin (centre of mass)
- A random "vibration" of atomic positions.

The mutation rates of real–space encoded GAs are generally an order of magnitude higher than those of a bit–string encoded GA. The low rate for bit–strings is due to the fact that changing the most significant bit on the string will have a dramatic effect on the population member. Mutation in real–space systems is usually dynamic, meaning that the new location of an atom is dependent on its old location. This change in location will typically affect a volume of $\approx 10\text{Å}^3$ of the population member; the member itself would typically have a volume $\gtrsim 100\text{Å}^3$. This means that a mutation would only effect $\lessapprox 10\%$ of the structure of the member when it is performed. A 10% mutation rate introduces approximately the same amount of disruption as in the bit–string approach. The use of a local minimiser in the real–space case also means that mutations themselves may have very little effect, since the local minimiser may "undo" any mutation.

The way that the vibration is performed is also important. This is controlled by two quantities, m_R and m_A. The mutation rate is $m_R \in [0, 1]$, and this determines the probability that each atom will be mutated or not after crossover, before the

local structure minimisation procedure. In this study once an atom has been selected for mutation then it is randomly placed in a cubic box with sides of length $2m_A$ (where $m_A > 0$ Å is the mutation amplitude) which has been centred on the atom's original position.

After mutation the cluster is minimised using a direct minimisation method. Also, because the local structure is well maintained during crossover, mutation is essential to aid the search of the potential energy surface.

2.3.6 Updating the Population

After the crossover/mutation steps have been performed the population needs to be updated. There are now at least double the number of members that were in the original population and so a cull of members needs to be performed. As in the selection stage, the updating process can be as simple or as complicated as required.

The use of elitist selection tends to lead to stagnation. While it will work well for small systems with relatively simple potential energy surfaces (see appendix A, section A.3), for larger systems the GA will find it increasingly difficult to move out of local minima. It is better to allow some variation in the update procedure.

If the ground state is being sought then hybrid selection is a good method to use. It means that the fittest individual is maintained as breeding stock, but that the other members of the new generation are allowed more variation. The fitness function used will also affect this, with an exponential fitness function still weighting the fitter members highly.

In contrast, pure roulette selection might be a good method if you were not too interested in the global minimum structure, but merely wanted to drive the population to a number of fit but different viable structures.

After update the knowledge of which population members were parents and which were offspring is lost and some members may survive for many generations. After the population is updated, the process starts again with selection for reproduction, onto crossover, mutation and finally updating. This cycle constitutes one generation.

2.3.6.1 Convergence

The loop shown in figure 2.3 could technically go on indefinitely – so some form of convergence criterion must be included. This is one of the trickier parts

of a GA. How can you tell if the solution is converged, i.e. that the solution has found the global minimum? In testing the GA the global minimum solution may be known, but in practise the global minimum solution for the system being studied will be unknown, so the GA must have a way of judging that the population is sufficiently converged to end the calculation.

In the original formulation, the GA not only found the global minimum basin in the potential energy surface, but also minimised within the basin. It was realised that locally minimising the structure simplifies the work that the GA (or any global minimisation technique that operates in this search space) has to do [Wales and Doye, 1997].

With no minimiser, the algorithm would bounce around in the bottom of the global minimum basin, but might never reach the minimum structure. Genetic Algorithms are very good at homing in on the global minimum basin, but will have difficulty in getting to the bottom of it. The use of the local minimiser means that the GA is only searching viable, relaxed, solutions to the problem.

However, determining if the population has converged is not as easy as it sounds. Due to the direct minimiser, all structures will be local minima, if not the global minimum. It may make sense to have the program check and see how long the lowest minimum has been the lowest energy structure found, and when it has been the lowest structure found for, say, t generations. This could then define that the GA is converged.

However, if t is too low, then the program will exit early and will be stuck in a local minimum. If t is too large then the algorithm may stay in the loop too long and waste resources. The length of time required may also depend on the complexity of the system – a simpler system would only need a small t, whereas a system with a complex potential energy surface may require a larger value.

Often during a GA calculation the population will remain with the same fittest member for a number of generations before falling into another basin. It may appear that the population has converged, when in fact it has just stagnated. In chapter 5, I will demonstrate one method that can be used to help prevent stagnation.

I will now describe the different flavours of genetic algorithm that have been used in solid–state physics problems. I have split these into four sections, as defined by the periodicity of the systems being studied. The reason for this should be made obvious when I outline the extension made to this method in chapter 4.

2.4 Zero–Dimensional Systems

2.4.1 Binary Encoded Solutions

Binary encoded GA studies of clusters have focused both on single–species systems [Hartke, 1993, 1996] and on conformations of polymers [Brodmeier and Pretsch, 1994], molecules [McGarrah and Judson, 1993] or relative orientations of larger clusters such as benzene [Xiao and Williams, 1993].

The work by Bernd Hartke initially used a binary encoded approach to study silicon clusters [Hartke, 1993, 1996]. Later, as the method proposed by Deaven and Ho [1995] grew in popularity, he also used real–space encoded methods.

In the case of polymers and molecular clusters, the binary string was made up of a binary representation of the torsion angles in the polymer [Brodmeier and Pretsch, 1994], or the (x,y,z) positions and (θ,ϕ,ψ) angles of one molecule with reference to another fixed molecule at the origin [Xiao and Williams, 1993]. Local minimisation was also used in these cases.

There are also studies which used lattice sites to define the positions of atoms, and these sites are encoded into the chromosome, with a 0 indicating an empty site, and a 1 an occupied one. In the study of Xiang *et al*. [2004] on large Lennard–Jones clusters, there was no crossover involved, but instead new members were found by pure mutation alone. Searches had to be performed on a number of different lattice types, and after new structures had been found the structure was locally minimised to relax the surface of the cluster.

2.4.2 Real–Space Encoded Solutions

There are two classes of real-space encoded solutions: pure real–space formulations where the crossover procedure is also performed in real–space, with reference to the relative difference between the atoms in space, and one in which the positions of the atoms are encoded in real space, but the crossover operation is performed by swapping sections of an array of atomic positions (and possibly angles, in the case of molecular clusters).

2.4.2.1 Real–Space Crossover

The original Deaven and Ho [1995] paper focused on the optimisation of various fullerene clusters. Having investigated the above methods they instead opted to develop their own method, which is essentially the real–space approach described in section 2.3.4.3.

This method has been applied to a large number of different systems. In a follow–up to the above paper, Lennard–Jones clusters were also investigated [Deaven *et al.*, 1996]. Lennard–Jones clusters are an ideal test system due to the computational ease in which the potential can be calculated, and the fact that they have a complicated potential energy surface [Wales and Doye, 1997]. Pullan [1997] gives a good summary as to why binary encoding for the cluster problem is inefficient. Binary strings are ideal when the parameters defining a particular problem can be defined as short, low–order schema which are also unrelated to schema at other fixed positions [Goldberg, 1989]. Pullan [1997] calculated that for a 100–atom Lennard–Jones cluster the potential energy surface would have of the order of 10^{140} local minima; however for binary strings a 30–atom cluster with a precision of 3 decimal places would require a binary string of 1260 bits, which has a search space of 10^{420} elements.

Wolf and Landman [1998] used an etching technique to improve the searching of the potential energy surface. They combined the Deaven and Ho [1995] method with an add–and–etch technique which involved adding atoms to the cluster size that they were interested in, and then removing them in subsequent generations. This study also used a seeding technique where instead of beginning from a totally random start, small idealised clusters were used to seed larger ones to reduce the time taken in the local minimisation step. While the use of a seeding technique may improve convergence, it also biases the answer from the outset.

Hartke has switched from using binary strings to using a Deaven and Ho [1995] based algorithm, with some changes, such as a fixed plane of crossover, rather than a random plane. He has published results on Lennard–Jones clusters [Hartke, 1999], mercury clusters [Hartke *et al.*, 2001] and water clusters [Hartke, 2002].

The review article by Johnston [2003] covers a large number of different systems studied using these methods. It also covers the general formalism for the real–space encoded GA.

2.4.2.2 Array Crossover

The work of Niesse and Mayne [1996] uses real–space co–ordinates for the positions of the cluster, stored in an array $X_i = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n)$. At each position is a molecule, defined by the $(x, y, z, \theta, \phi, \psi)$ values corresponding to the location of a fixed point on the molecule and the rotation angles of the molecule defined with respect to that point. A second string, $Y_i = (\theta_1, \phi_1, \psi_1, \ldots, \theta_n, \phi_n, \psi_n)$ contains these angles. Operations are carried out on these strings separately.

Zeiri [1995] defined six operators that could be used for updating the population when using this representation:

1. *n*–mutation: One parent producing one offspring.
   One of the fitter chromosomes is mutated by a dynamic mutation using a random number evaluated from a Gaussian distribution.

2. Inversion: One parent producing one offspring.
   The order of a random part of the parent chromosome is inverted.

3. 2–point cross link: Two parents producing two offspring. Standard crossover – a single crossover point is chosen and array elements are swapped, similar to binary crossover.

4. *n*–point cross link: Two parents producing two offspring.
   Each array element from each parent is either given to offspring 1 or offspring 2 depending on a uniform random number.

5. Arithmetic average: Two parents producing one offspring.
   The arithmetic average of each of the array elements in the two parents is used to form the offspring.

6. Geometric average: Two parents producing one offspring.
   The geometric average of each of the array elements in the two parents is used to form the offspring.

This set of operators is more flexible than the Deaven and Ho [1995] approach. The study of Niesse and Mayne [1997] did a comparison of the Deaven and Ho [1995] method with that of Zeiri [1995] and found that the Deaven and Ho [1995] method was comparable to inversion and 2–point cross link, but was inferior to the averaging operators for small Lennard–Jones clusters.

2.5 One–Dimensional Systems

There have been some studies done on nanowires, using the Deaven and Ho [1995] formalism. Studies have been done on gold [Wang *et al.*, 2001b], titanium [Wang *et al.*, 2001a] and zirconium [Wang *et al.*, 2002]. The relationship of the periodicity of the system to the use of the crossover operation is not discussed. These results show the appearance of a bulk–like character within the wire as its thickness is increased.

2.6 Two–Dimensional Systems

There have been a number of surface studies that have utilised the GA formalism, using both binary and real–space encoded methods.

2.6.1 Binary Encoded Solutions

The work of Fu *et al*. [1997] used a tight binding scheme to study the Si(001) surface. Only the top layer undergoes the GA process and the system was periodic in x and y. The co–ordinates were mapped onto binary strings for crossover and mutation and the system was relaxed into the local minimum through quenched molecular dynamics (600 time-steps covering 0.6 ps). The minimum energy configuration found was from the 21st generation, and remained so until the 65th (the length of the calculation).

Miyazaki and Inoue [2002] used a GA to study the deposited structure of atoms in a Lennard–Jones potential. The system was set up as a $2^{10} \times 2^{10} \times 2^{10}$ mesh, and used 30 bit long strings. A seeding technique was used to determine structures, and the LJ_{13} cluster was used as a test method. They also used a low mutation rate of 0.0003, and recombination rate (a binary analogue of n–point cross link as defined above) of 0.2. The substrate was 10 layers thick with a FCC(001) surface. No periodic boundary conditions were imposed.

GAs have also been applied to surface structure determination from surface diffraction data [Landree *et al*., 1997]. In this case the fitness function was a comparison to the experimental data, and the genes of the chromosome represented phase information.

The parameters used in the GA chromosome need not be atomic positions or torsion angles. The work of Doll and Vanhove [1996] used layer spacings as the values to be optimised in a GA study of Ir $(110) - (1 \times 2)$ surface by comparing with low energy electron diffraction (LEED) data. The fitness function in this case used the Pendry R–factor, R_p, in its calculation.

2.6.2 Real–Space Encoded Solutions

Sun *et al*. [2004] used a Deaven and Ho [1995] based GA to study the structure of adatom clusters on a Ag (111) surface. The GA component only operates on the adatom clusters, and not the surface itself. The local–minimisation step also only takes place on the frozen surface. Elitist selection was used in the update stages.

The work of Chuang *et al*. [2004] was the first real–space represented GA designed for surface structure calculations. The surface of Si(105) was investigated using an empirical potential. The system was periodic in *x* and *y*, with no periodicity in the *z*–direction. Only atoms in the top 5 Å of the slab undergo the genetic algorithm.

Crossover was performed with a plane perpendicular to the surface, and then the whole of the structure (15-20 Å deep) is relaxed. This study allowed the number of atoms to vary during the crossover stage, and did not hold a constant atom number during the course of the calculation. Less than 200 mating operations were required to give the lowest structure. This group have continued their work on GAs for surface studies and have also published a study on the Si(114) surface [Chuang *et al*., 2005]. There is some excitement about this method from other surface scientists [Zandvliet, 2005].

2.7 Three–Dimensional Systems

There have been a number of publications attempting to do truly *ab initio* crystal structure determination using genetic algorithms, as well as those that are comparing results to powder diffraction data. Studies based on powder diffraction data have been encoded in both binary string and real–space (array–strings). Rather than cataloguing these by the type of representation used, I will instead categorise by the application.

2.7.1 Binary Alloys

Smith [1992] used a GA to create starting configurations of atoms in binary alloys for use in Molecular Dynamics (MD) calculations. He used a binary encoded approach of a 50–50 mixed configuration of atoms to get the minimum enthalpy configuration before starting the MD run.

The more recent work of Johannesson *et al*. [2002] used an encoding scheme which used the atomic type as the gene, and its position in the chromosome determined its position in a FCC or BCC lattice.

2.7.1.1 Cluster Expansion

A recent approach developed by Hart, Blum, Walorski, and Zunger is a binary encoded approach which combines the method of cluster expansion [de Fontaine, 1994] with those of GAs [Hart *et al*., 2005; Blum *et al*., 2005]. The chromosome in this case represents a chosen subset of 1–, 2–, 3–, 4–, 5– and 6–body interaction terms that exist within the chosen lattice structure. The search space of the system is a set of n–body terms; the optimisation determines which terms are important. One draw–back of this method is that the lattice type of the system needs to be fixed in this method.

The problem with these binary alloy studies is the assumption of the lattice type of the system. Unless all lattice types are considered (which will be time consuming) the system is biased in a particular direction. It is better if the lattice could be also determined *ab initio*, i.e. during the course of a calculation.

2.7.2 Powder Diffraction Studies

2.7.2.1 Binary Encoded Solutions

The work of Woodley and others has also presented a method of crystal structure determination using powder diffraction data using a binary string approach by mapping the real–space atomic coordinates onto a lattice [Woodley *et al.*, 1999; Woodley, 2004b]. He has also investigated zeolite structures by incorporating exclusion zones into his method and preventing atoms from occupying sites on the lattice in these areas [Woodley *et al.*, 2004b,a; Woodley, 2004c]. A review article on this method can be found in Woodley [2004a].

2.7.2.2 Real–Space Encoded Solutions

Research by Harris, Johnston and others uses a real–space array–string encoded representation similar to that described in section 2.4.2.2 [Kariuki *et al.*, 1997]. Local minimisation is also used in this technique, and a large number of systems have been studied using this method. Their work has involved studies on *para*–methoxybenzoic acid, formylurea and *ortho*–thymotic acid [Harris *et al.*, 1998a,b]; L–glutamic acid [Turner *et al.*, 2000]; and peptides [Tedesco *et al.*, 2001; Cheung *et al.*, 2002; Harris *et al.*, 2004].

2.7.3 Other Systems studied using Real–Space Encoding

Bazterra *et al.* [2004] used a similar encoding method to that described in section 2.4.2.2, but in this case the six degrees of freedom that defined the unit cell $(a, b, c, \alpha, \beta, \gamma)$ were also included in the chromosome. This allowed the cell to be optimised at the same time as the atomic co-ordinates. The structures of molecular crystals of L–alanine and DL–alanine were investigated using this method.

There is also the work by Artem Oganov and Colin Glass at Zurich, who are developing a code called USPEX [Oganov *et al.*, 2006; Oganov and Glass, 2006]; however, no details of their method have been published as yet, although they have presented some results. They appear to be a using real–space method which uses planar cuts and fractional co–ordinates, and is capable of optimising the unit cell using a local minimiser. However, they have not yet published a detailed description of their method or comparison with current methods and so no further comment can be made on the validity of their method.

2.8 A Brief Overview of Other Global Optimisation Techniques

While there are a large number of global optimisation techniques I will list some that have been applied to similar problems to those discussed earlier.

2.8.1 Stochastic Methods

Early work was done on the Lennard–Jones cluster problem by Hoare and Pal [1971a,b, 1972] where a seeding technique, in combination with relaxation of the cluster, was used to grow new minima. Other work on Lennard–Jones clusters was done using simulated annealing [Wille, 1987] (for a full description of this method, please see section A.2) for $N \leq 25$. A related technique, Gaussian Density Annealing [Tsoo and Brooks, 1994] also gave good results for this range of cluster sizes. In the Hoare and Pal [1971a] technique it is hard to see how this method could be extended effectively to bulk systems, and the work on Simulated Annealing becomes less effective as the dimensionality of the potential energy surface increases (see section A.2). This may happen partly because the simulated annealing algorithm also has to perform the local optimisation as well as the global.

The work of Wales and others has mainly used the technique of Basin Hopping [Wales and Scheraga, 1999], which is a Monte–Carlo based method but explores the same potential energy surface as a real–space encoded GA using local minimisation. This is similar to the Simulated–Annealing method described above (except where the temperature, T, is merely an adjustable parameter and is not used for cooling) with a very large atomistic perturbation combined with local optimisation. The perturbation on the system needs to be larger, so as to move it from the current basin of attraction into a neighbouring one. This work found all the minima up to 110 atoms for the Lennard–Jones cluster problem. This group has also made use of genetic algorithms in the Deaven and Ho [1995] formulation [Wales, 2003].

Doye and Wales [1997] have also used the technique of Eigenvector Following to explore the potential energy surface. The Hessian matrix (the matrix of second–derivatives) of the system is determined and is used to choose search directions; once a new basin has been found a Metropolis [Metropolis *et al.*, 1953] criterion is used to determine if moving to that basin should be accepted or rejected.

A similar technique to Basin Hopping is Minima Hopping which has shown

promising results for bulk systems [Goedecker, 2004]. In Minima Hopping the accept/reject ratio is continually adjusted to ensure that the potential energy surface is explored more thoroughly than it might otherwise be in a thermodynamically based method that is dependent on a "temperature".

Good results can also be found by simply creating a large number of random, highly disrupted structures and then locally minimising them. Pickard and Needs [2006] have used this method to examine the high–pressure phases of silane. This method will soon fall into problems as the number of atoms in the structures increases due to the exponentially increasing size of the potential energy surface, but for smaller numbers of atoms it is a useful technique.

Metadynamics, a technique developed by Laio and Parrinello [2002] has also shown good results for a number of different molecular systems [Laio and Parrinello, 2002; Laio *et al.*, 2005]. This method works by filling in areas of the potential energy landscape with Gaussian functions centred on areas previously searched, which then prevents the system returning to a state that has been previously searched. It is also able to work at finite temperature, which GA methods are currently unable to do.

Molecular Dynamics (MD) methods have been used to determine the structure of Lennard–Jones [Honeycutt and Andersen, 1987] clusters. MD can also be used to determine the bulk phases of other materials [Quigley and Probert, 2005a,b].

There has also been some work on Differential Evolution, another evolutionary technique, which was used to determine crystal structures from powder diffraction data whilst also intelligently constraining the search space [Chong and Tremayne, 2006]. Differential Evolution works by adding the difference of two population members onto a third, producing offspring [Becerra and Coelloa, 2006]. In this study the members were represented as described in section 2.4.2.2.

2.8.2 Systematic Methods

Systematic methods are not widely used in this field due to the exponentially increasing size of the potential energy surface with the number of atoms. Graph theory has been used for small numbers of atoms to determine the possible polymorphs of up to 6 atoms of sp^2– [Winkler *et al.*, 1999, 2001] and 4 atoms of sp^3– [Strong *et al.*, 2004] hybridised carbon crystals. It is not possible for both sp^2 and sp^3 structures to be found in the course of a single search using this method.

It should be noted that this list is not exhaustive, and that there are many other methods that can be employed in the field of crystal structure determination.

2.9 Conclusions

I hope that I have been able to show in this chapter how powerful Genetic Algorithms are, especially when they are combined with local minimisation methods. Genetic algorithms are flexible and can learn about and exploit features of the potential energy surface, and are suitable for use in solid–state physics problems.

There has been a large amount of study of Lennard–Jones systems, mostly clusters due to the plethora of minima that exist in those systems, and due to the ease of calculation with the Lennard–Jones potential. The GA method has been shown to be a very useful technique in studying these zero–dimensional systems and recently real–space encoded methods for tackling this problem have been expanded into more dimensions to study nanowires, surfaces and bulk crystals. However, there are a large number of competing methods, none of which allow for truly *ab initio* crystal structure determination, except possibly that of Oganov *et al*. [2006]; Oganov and Glass [2006] whose method has yet to be described in detail (and also for the non–GA method of Pickard and Needs [2006], but this method has yet to be proven for large systems).

While the Deaven and Ho [1995] approach is very "physical" in its approach to the crossover operation, the methods proposed for array–represented clusters are more flexible. However, some of these methods may not be as suitable in higher–dimensional systems. These methods have shown good results for molecular crystals where the rotation of the molecules is also important, but they are not as successful in pure, single species systems. I suspect that a mixture of approaches may be the best when optimising systems made up of molecules (i.e. the centroid positions of the molecules is mixed in a Deaven and Ho [1995] crossover procedure, and the angles of the molecules are mixed as in Zeiri [1995]).

I hope to describe in the following chapters a method that allows for *ab initio* crystal structure determination both by using the empirical potentials and by using the framework of DFT described in the next chapter. I will also describe an extension to the fitness function which allows for the comparison of different structures during the course of a calculation and will therefore improve convergence to the global minimum and will help prevent stagnation in local minimum structures.

Chapter 3

Methods of Energy Calculation

3.1 Introduction

As was discussed in the previous chapter, any genetic algorithm method requires some method of calculation of fitness, and in solid–state physics problems this measure is usually associated with the energy of the system. The methods themselves have no bias as to how this energy is calculated. In addition to the energy, the local minimisers also require the forces on the system when they relax a structure down to the bottom of its local basin of attraction.

In this chapter I will discuss the methods of energy calculation used in this study. I will start with empirical potentials and then move on to the *ab initio* method of density functional theory.

3.2 Empirical Potentials

3.2.1 Super–cells and the Minimum Image Convention

Due to computational necessity it is impossible to study a large enough system of atoms that approaches that seen in true bulk solids, i.e. of the order of 10^{23}–atoms. However it is possible to simulate a larger number of atoms than would otherwise be possible using a super–cell approach, as is described in diagram 3.1. In this diagram the central cell contains the atoms that are represented in the calculation, and those atoms are then effectively tessellated across all space using periodic boundary conditions to simulate a bulk crystal, fluid or gas. This is the *super–cell* approach used in atomistic simulations.

In this diagram the central cell is the cell which is considered and stored. The image cells surrounding it are there to simulate the infinite bulk surrounding

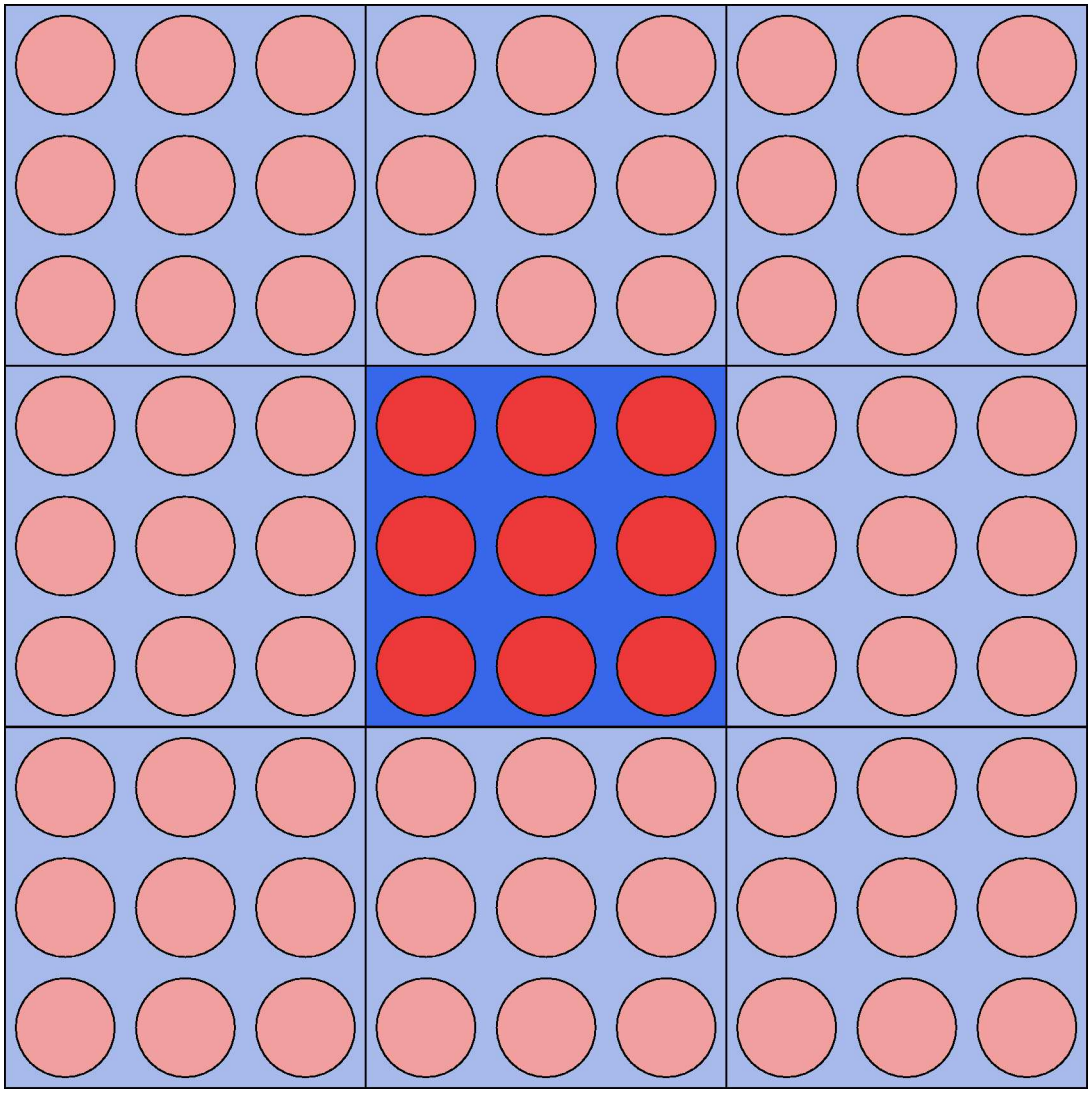

Figure 3.1: Diagram showing the super–cell approach in computational simulation. The central cell is the simulation cell, and the surrounding cells are the periodic images of this cell.

the cell. Interactions occur through the walls of the cell, and if an atom moves out of the central cell its periodic image will replace it as it moves in on the other side of the cell.

This approach make no assumptions as to how the energy of and the forces on the atoms are calculated. In an *ab initio* framework it is possible to simulate a large bulk using only a small number of atoms, but when using an empirical potential it is necessary to prevent self–interactions by using a larger number of atoms.

Some empirical potentials have a long tail, which asymptotically approaches the zero as $r \to \infty$, and so effectively leads to interactions over large distances. In a zero–dimensional cluster GA, the fact that there is a tail is unimportant,

since there is no periodicity in the system, but once calculations are performed using a super–cell it is necessary to truncate this tail. Also, if the simulation super–cell is smaller than the maximum interaction (or *cut–off*) radius, R_{cut}, then it is possible for an atom to interact with itself. To prevent this from happening it is necessary to make the super–cell large enough to prevent self–interaction. Normally atom i will interact with atom j and all its images, but the minimum image convention defines that atom i will only interact with the closest image of atom j.

3.2.2 The Lennard–Jones Potential

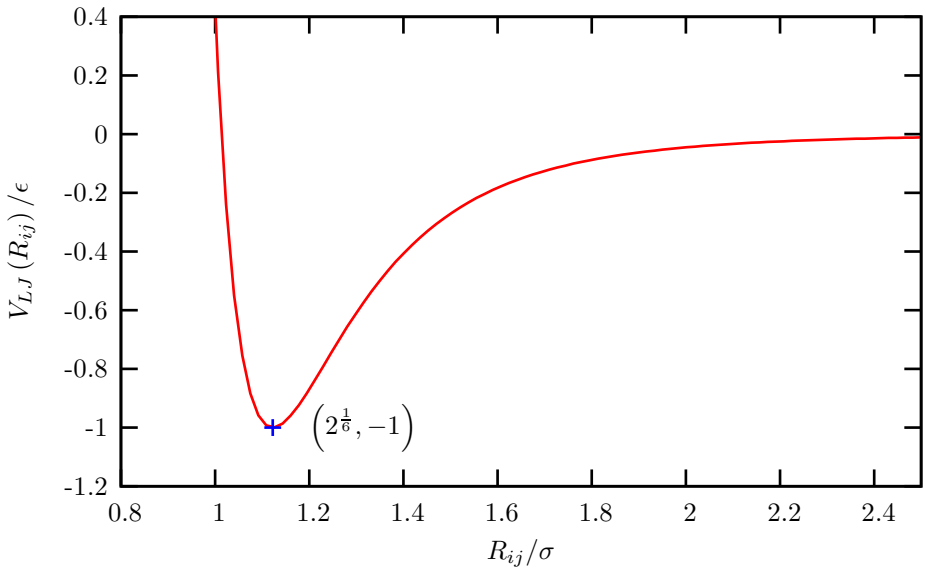

Figure 3.2: The Lennard–Jones Potential

The Lennard–Jones potential [Lennard-Jones and Ingham, 1925] shown in figure 3.2 has the form

$$V_{LJ}\left(R_{ij}\right) = 4\varepsilon\left[\left(\frac{\sigma}{R_{ij}}\right)^{12} - \left(\frac{\sigma}{R_{ij}}\right)^{6}\right] \tag{3.1}$$

and is a potential that was originally used as an approximation to the energy surface experienced by the Noble Gases [Pollack, 1964b]. Due to its ease of calculation it is now primarily used as a useful test case potential for computational methods.

The force felt by atoms moving in this potential can be easily determined by

$$\mathbf{F}_i = -\frac{\mathrm{d}V}{\mathrm{d}\mathbf{R}_i} \tag{3.2}$$

where $\mathbf{R}_{ij}$ is the vector connecting atom i to atom j. In this way the force is

$$\mathbf{F}_{LJ}(R_i) = \sum_j 4\varepsilon \left[\left(\frac{12\sigma^{12}}{R_{ij}^{13}} \right) - \left(\frac{6\sigma^6}{R_{ij}^7} \right) \right] \hat{\mathbf{R}}_{ij} \tag{3.3}$$

The potential has an infinitely long tail, so to prevent any calculation falling foul of the minimum image convention (section 3.2.1) it is often truncated in some way. The method that I shall use is the shifted–force formulation of Stoddard and Ford [1973]

$$V^{SF}(R_{ij}) = \begin{cases} V_{LJ}(R_{ij}) - V_{LJ}(R_{cut}) - \left(\frac{\mathrm{d}V_{LJ}(R_{ij})}{\mathrm{d}R_{ij}} \right)_{(R_{ij}=r_{cut})} (R_{ij} - R_{cut}) & R_{ij} \leq R_{cut} \\ 0 & R_{ij} > R_{cut} \end{cases} \tag{3.4}$$

which ensures that *both* the energy and force vanish at R_{cut}.

The Lennard–Jones potential has a HCP ground state structure, with the FCC–phase being nearly degenerate in energy with it. This is not the case with Noble Gases, where FCC is the ground state structure [Pollack, 1964b].

3.2.3 The Dzugutov Potential

The Dzugutov potential was developed by Mikhail Dzugutov [1992] as a simple pair potential that has polytetrahedral and icosahedral order [Doye *et al.*, 2003]. It is similar to the Lennard–Jones potential (section 3.2.2) but it has an extra repulsive component as can be seen in figure 3.3. The potential is defined as

$$\Phi_1\left(R_{ij}\right) = \begin{cases} A\left(R_{ij}^{-m} - B\right)\exp\left(\frac{c}{R_{ij}-a}\right) & R_{ij} < a \\ 0 & R_{ij} \geq a \end{cases} \tag{3.5}$$

$$\Phi_2\left(R_{ij}\right) = \begin{cases} B\exp\left(\frac{d}{R_{ij}-b}\right) & R_{ij} < b \\ 0 & R_{ij} \geq b \end{cases} \tag{3.6}$$

$$\Phi\left(R_{ij}\right) = \Phi_1\left(R_{ij}\right) + \Phi_2\left(R_{ij}\right) \tag{3.7}$$

with the constants defined in table 3.1. It was originally formulated to simulate liquid systems, however it has also been shown to have some interesting solid phases [Roth and Denton, 2000], and can also be used to form quasi–crystals [Dzugutov, 1993].

The force felt by atoms moving in the Dzugutov potential can be determined by using equation 3.2. The phases of the Dzugutov potential will be discussed in detail in chapter 7. A graph comparing the Lennard–Jones potential and the Dzugutov potential is shown in figure 3.4

m	A	c	a	B	d	b
16	5.82	1.1	1.87	1.28	0.27	1.94

Table 3.1: Table of parameters used in the Dzugutov potential (equation 3.7).

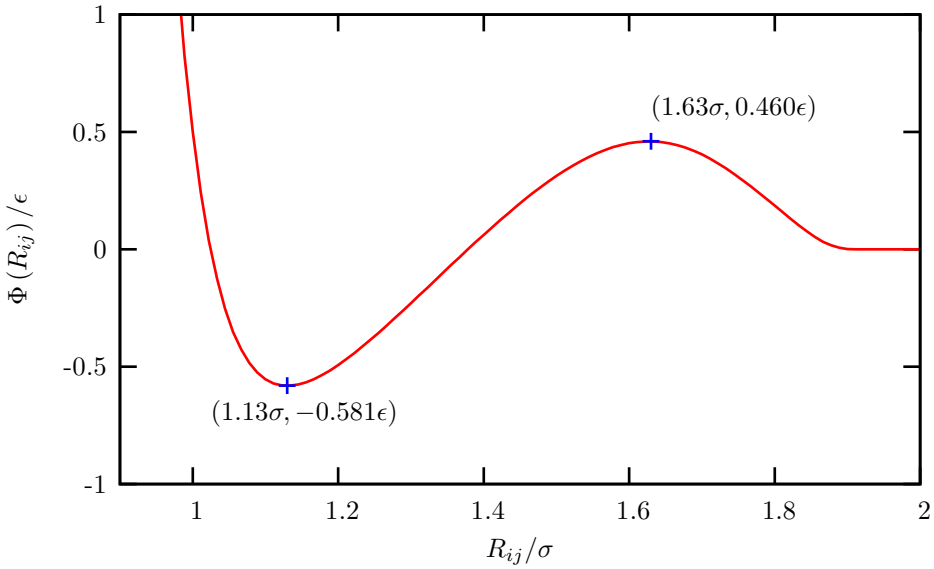

Figure 3.3: The Dzugutov potential (equation 3.7 using the values in table 3.1) scaled with respect to the Lennard–Jones potential (equation 3.1). The values of σ and ε are the same between both figures.

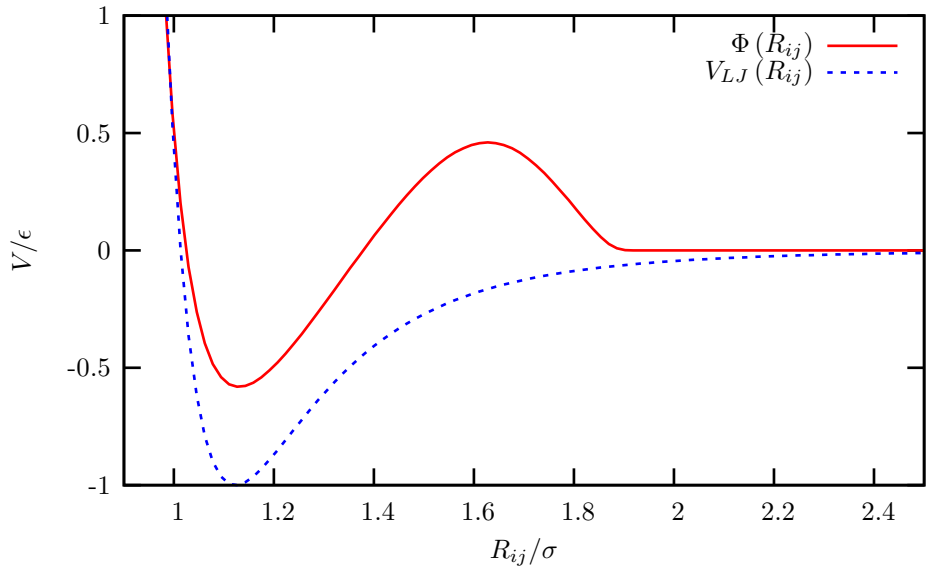

Figure 3.4: Graph showing both the Dzugutov potential (red) and the Lennard–Jones potential (blue).

3.3 Density Functional Theory

The GA method that has been developed, and will be described in detail in chapter 4, has been incorporated into the *ab initio* plane–wave density functional theory (DFT) code CASTEP [Segall *et al.*, 2002]. I will give a brief description of what density functional theory is, and I will also touch on some points that become important when the GA method is applied to DFT in a variable cell optimisation.

3.3.1 Mathematical Framework

3.3.1.1 The Born–Oppenheimer Approximation

A time–independent system of N fully interacting electrons and $\mathfrak{N}$ ions can be described by

$$\hat{H}\left|\Psi\left(\mathbf{R}_1,\ldots,\mathbf{R}_{\mathfrak{N}};\mathbf{r}_1,\ldots,\mathbf{r}_N\right)\right\rangle = U\left|\Psi\left(\mathbf{R}_1,\ldots,\mathbf{R}_{\mathfrak{N}};\mathbf{r}_1,\ldots,\mathbf{r}_N\right)\right\rangle \tag{3.8}$$

where

$$\begin{aligned}\hat{H} = &-\frac{\hbar^2}{2M_i}\sum_{\mathfrak{i}=1}^{\mathfrak{N}}\nabla_{\mathfrak{i}}^2 - \frac{\hbar^2}{2m_e}\sum_{i=1}^{N}\nabla_i^2 - \frac{e^2}{4\pi\varepsilon_0}\sum_{\mathfrak{i}=1}^{\mathfrak{N}}\sum_{j=1}^{N}\frac{Z_{\mathfrak{i}}}{\left|\mathbf{R}_{\mathfrak{i}}-\mathbf{r}_j\right|}\\ &+\frac{1}{2}\frac{e^2}{4\pi\varepsilon_0}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{1}{\left|\mathbf{r}_i-\mathbf{r}_j\right|}+\frac{1}{2}\frac{e^2}{4\pi\varepsilon_0}\sum_{\mathfrak{i}=1}^{\mathfrak{N}}\sum_{\mathfrak{j}\neq\mathfrak{i}}^{\mathfrak{N}}\frac{Z_{\mathfrak{i}}Z_{\mathfrak{j}}}{\left|\mathbf{R}_{\mathfrak{i}}-\mathbf{R}_{\mathfrak{j}}\right|}\end{aligned} \tag{3.9}$$

and U is the total energy of the system. The Born–Oppenheimer approximation [Born and Oppenheimer, 1927] allows the electronic problem to be separated from the ionic problem. It states that due to the large mass of the nuclei compared to the electrons the nuclear and electron motions are decoupled, and the nuclear wave–functions can be treated as δ–functions. In this case the purely electronic Hamiltonian can be written as

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \sum_{\mathfrak{i}=1}^{\mathfrak{N}}\sum_{j=1}^{N}\frac{Z_{\mathfrak{i}}}{\left|\mathbf{R}_{\mathfrak{i}}-\mathbf{r}_j\right|}+\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N}\frac{1}{\left|\mathbf{r}_i-\mathbf{r}_j\right|} \tag{3.10}$$

where, for ease of computation the standard units to use are atomic units where

$$\hbar = e = m_e = 4\pi\varepsilon_0 = 1.$$

The first term on the right–hand side of equation 3.10 is the kinetic energy of the electron, the second term is the electron–ion interaction, or the *external* potential $V_{ext}(\mathbf{r})$, and the third term is the electron–electron interaction. This means that the purely electron Hamiltonian can be written as

$$\hat{H}\left|\Phi\left(\mathbf{r}_i,\ldots,\mathbf{r}_N\right)\right\rangle = E\left|\Phi\left(\mathbf{r}_i,\ldots,\mathbf{r}_N\right)\right\rangle \tag{3.11}$$

where Φ is the wave–function of the electrons only and E is the total energy of the electrons.

The approach that DFT takes is to say that ground–state properties are dependent only on the ground–state density. The ground state energy is a unique *functional* [1] of the density of the system n.

The Hohenberg–Kohn theorem [Hohenberg and Kohn, 1964] states that the ground–state density $n(\mathbf{r})$ minimises the functional $E[n]$.

Proof. Suppose that we have two different external potentials, $V_{ext}(\mathbf{r})$ and $V'_{ext}(\mathbf{r})$ which give the same ground state density, $n_0(\mathbf{r})$. These potentials will give two different Hamiltonians, $\hat{H}$ and $\hat{H}'$, with different ground–state wave–functions, Φ and Φ' (which both give the same ground state density of a non–degenerate ground–state) with total energies E and E'. It therefore follows that

$$E = \left\langle\Phi\left|\hat{H}\right|\Phi\right\rangle < \left\langle\Phi'\left|\hat{H}\right|\Phi'\right\rangle \tag{3.12}$$

which will give the inequality

$$\begin{aligned} E &< \left\langle\Phi'\left|\hat{H}'\right|\Phi'\right\rangle + \left\langle\Phi'\left|\hat{H}-\hat{H}'\right|\Phi'\right\rangle \\ &< E' + \iiint \mathrm{d}^3\mathbf{r}\,\left[V_{ext}(\mathbf{r}) - V'_{ext}(\mathbf{r})\right] n_0(\mathbf{r}) \end{aligned} \tag{3.13}$$

By using the exactly the same method for E'

$$E' < E + \iiint \mathrm{d}^3\mathbf{r}\,\left[V'_{ext}(\mathbf{r}) - V_{ext}(\mathbf{r})\right] n_0(\mathbf{r}) \tag{3.14}$$

Then adding equations 3.13 and 3.14 we get

$$E + E' < E + E' \tag{3.15}$$

This inconsistency means that there can be no two external potentials (which

[1] a *functional* maps a function from a certain domain and returns a number

are different by more than a constant) which give rise to the same non–degenerate ground state density [Hohenberg and Kohn, 1964; Martin, 2004].

If we consider the exact ground state Hamiltonian, $\hat{H}_{gs}$, constructed from the ground–state density $n_0(\mathbf{r})$, then now say we have two wave–functions, Φ which will give the ground–state density, and Φ' which will give a different density $n'(\mathbf{r})$. Following on from equation 3.12 this gives

$$E_0 = \langle \Phi | \hat{H}_{gs} | \Phi \rangle < \langle \Phi' | \hat{H}_{gs} | \Phi' \rangle = E' \tag{3.16}$$

which means that the energy of the functional $E[n]$ constructed using the ground–state density is lower for the ground–state density n_0 than for any other density n' [Martin, 2004]. If this functional were known, then minimising the total energy of the system with respect to $n(\mathbf{r})$ would give the exact ground state density and energy. This theory only applies for ground state properties, and cannot be applied to excited states. □

So the ground state density of a system will give the correct ground state energy. Kohn and Sham [1965] proposed this Hamiltonian to be written in terms of non–interacting electrons, which still gives the same ground–state density as a system of fully–interacting electrons

$$\Phi(\mathbf{r}_i, \ldots, \mathbf{r}_N) = \sum_{i=1}^{N} \phi_i(\mathbf{r}) \tag{3.17}$$

where the density of the system is

$$n(\mathbf{r}) = \sum_{i=1}^{N} \phi_i^*(\mathbf{r}) \phi_i(\mathbf{r}) \tag{3.18}$$

The kinetic energy functional of the system of non–interacting "single–particles" is

$$T_s[n(\mathbf{r})] = -\frac{1}{2} \sum_{i=1}^{N} \langle \phi_i(\mathbf{r}) \left| \nabla_i^2 \right| \phi_i(\mathbf{r}) \rangle \tag{3.19}$$

while the functional representing the classical Coulomb (or Hartree) potential is

$$V_H[n(\mathbf{r})] = \frac{1}{2} \iint \frac{n(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}' \tag{3.20}$$

the functional of the external potential is known to be

$$V_{ext}\left[n\left(\mathbf{r}\right)\right] = \int \mathrm{d}\mathbf{r}\, V_{ext}\left(\mathbf{r}\right) n\left(\mathbf{r}\right) \tag{3.21}$$

The remaining unknown terms in $\hat{H}$ can then be placed into a functional known as the *exchange–correlation* functional

$$E_{xc}\left[n\left(\mathbf{r}\right)\right] = \left(T\left[n\left(\mathbf{r}\right)\right] - T_s\left[n\left(\mathbf{r}\right)\right]\right) + \left(V_{e\text{–}e}\left[n\left(\mathbf{r}\right)\right] - V_H\left[n\left(\mathbf{r}\right)\right]\right) \tag{3.22}$$

where $T\left[n\left(\mathbf{r}\right)\right]$ is the kinetic energy functional and $V_{e\text{–}e}\left[n\left(\mathbf{r}\right)\right]$ is the electron–electron potential of the fully interacting system. These quantities are unknown and so the form of $E_{xc}\left[n\left(\mathbf{r}\right)\right]$ must be approximated.

Now by use of the variational principle we can write equation 3.11 in terms of a single–particle wave–function (or *orbital*) and the Kohn-Sham Hamiltonian, $\hat{H}_{KS}$

$$\hat{H}_{KS}|\phi_i\left(\mathbf{r}\right)\rangle = \varepsilon_i|\phi_i\left(\mathbf{r}\right)\rangle \tag{3.23}$$

where ε_i is the energy of the single–particle orbital and $\hat{H}_{KS}$ is

$$\hat{H}_{KS} = -\frac{1}{2}\nabla^2 + V_{ext}\left(\mathbf{r}\right) + \int \mathrm{d}\mathbf{r}'\, \frac{n\left(\mathbf{r}'\right)}{\left|\mathbf{r}-\mathbf{r}'\right|} + V_{xc}\left(\mathbf{r}\right) \tag{3.24}$$

where

$$V_{xc}\left(\mathbf{r}\right) = \frac{\delta E_{xc}\left[n\left(\mathbf{r}\right)\right]}{\delta n\left(\mathbf{r}\right)} \tag{3.25}$$

which is the functional derivative of $E_{xc}\left[n\left(\mathbf{r}\right)\right]$. Due to the corrections made by the use of V_{xc}, the density $n\left(\mathbf{r}\right)$ found using equation 3.23 will be the same as that found using equation 3.11.

3.3.2 The Local Density Approximation

While this DFT is powerful it will be impossible to calculate unless the form of $E_{xc}\left[n\right]$ can be found. It may be thought that the form of this functional must be as complex as the system it describes. However it is possible to gain meaningful results from a simple approximation to this functional. Kohn and Sham [1965] made an approximate form of E_{xc} using the following approximation

$$E_{xc}^{LDA}\left[n\left(\mathbf{r}\right)\right] = \iiint \mathrm{d}^3\mathbf{r}\, n\left(\mathbf{r}\right) \varepsilon_{xc}\left(n\left(\mathbf{r}\right)\right) \tag{3.26}$$

where $\varepsilon_{xc}(n)$ is the exchange–correlation energy per electron in a homogeneous electron gas of density n. Despite being a very simple equation it gives surprisingly good results and by using the methods described above a large number of systems can be studied accurately [Segall *et al.*, 2002]. The form of ε_{xc} is made by fitting to data from a Quantum Monte Carlo calculation performed on a homogeneous electron gas, such as that of Ceperley and Alder [1980], which has been parameterised in some way, as in the work of Perdew and Zunger [1981].

There are other methods of calculating an approximation to E_{xc}, by using information about the gradient of the density approximation. These methods are known as Generalised Gradient Approximations (GGAs) [Perdew *et al.*, 1992]. I will only be using the LDA functional for reasons that will be explained in chapter 6.

3.3.3 Calculation of Forces: The Hellmann–Feynman Theorem

Although DFT gives a method for calculating the ground state energy of the system, it does not describe the method in which the "forces" are calculated. Since the local minimisation stages require forces as well as energies then the following will describe how this can be done.

One brute force method is to calculate the energy of a configuration of atoms, make a slight perturbation to this configuration and recalculate the energy, and then repeat the process. This method allows a graph to be made of position verses energy, and so the force can be calculated from this graph. The so–called Hellmann–Feynman theorem [Feynman, 1939] gives a framework for calculating the force without the need for repeated energy calculations. This theorem is only valid for the steady–state case, i.e. for

$$\hat{H}|\phi\rangle = E|\phi\rangle \tag{3.27}$$

where the wave–function $\langle\phi\rangle$ is an eigenfunction of $\hat{H}$. The energy of the system can therefore be defined as

$$E = \left\langle \phi \middle| \hat{H} \middle| \phi \right\rangle \tag{3.28}$$

If we differentiate with respect to a co–ordinate of the system, λ, which could be position, then

$$\frac{\partial E}{\partial \lambda} = \left\langle \phi \middle| \frac{\partial \hat{H}}{\partial \lambda} \middle| \phi \right\rangle + \left\langle \frac{\partial \phi}{\partial \lambda} \middle| \hat{H} \middle| \phi \right\rangle + \left\langle \phi \middle| \hat{H} \middle| \frac{\partial \phi}{\partial \lambda} \right\rangle \tag{3.29}$$

then due to the relation in equation 3.27 and that

$$\langle\phi|\hat{H} = \langle\phi|E \tag{3.30}$$

(because $\hat{H}$ is Hermitian) equation 3.29 can be written as

$$\frac{\partial E}{\partial\lambda} = \left\langle\phi\left|\frac{\partial\hat{H}}{\partial\lambda}\right|\phi\right\rangle + E\left\langle\frac{\partial\phi}{\partial\lambda}\middle|\phi\right\rangle + E\left\langle\phi\middle|\frac{\partial\phi}{\partial\lambda}\right\rangle \tag{3.31}$$

These last two terms can be written as

$$E\left\langle\frac{\partial\phi}{\partial\lambda}\middle|\phi\right\rangle + E\left\langle\phi\middle|\frac{\partial\phi}{\partial\lambda}\right\rangle = E\frac{\partial}{\partial\lambda}\langle\phi|\phi\rangle \tag{3.32}$$

and since $\langle\phi|\phi\rangle = 1$ this term is zero. Therefore

$$\frac{\partial E}{\partial\lambda} = \left\langle\phi\left|\frac{\partial\hat{H}}{\partial\lambda}\right|\phi\right\rangle \tag{3.33}$$

and so the force can be written as

$$F = -\frac{\partial E}{\partial\lambda} = -\left\langle\phi\left|\frac{\partial\hat{H}}{\partial\lambda}\right|\phi\right\rangle \tag{3.34}$$

The reason why this approach must be taken is that, unlike for energy, there is no variational principle for forces.

3.3.4 k–points

Using DFT it is necessary to calculate certain reciprocal–space properties. In a computational framework it also simplifies matters when certain quantities are calculated in reciprocal space. While this thesis will not discuss the finer points of electronic structure calculations I will briefly sketch over some points that become relevant when combining DFT with genetic algorithms.

3.3.4.1 Reciprocal Lattice

My intention is to use the ideas of Genetic Algorithms described in chapter 2 to study periodic bulk crystals (and we are also using a super–cell approach which is inherently periodic). In this case our cell can be defined by three lattice vectors, **a**, **b** and **c** which define the co-ordinate directions in our cell. We can also define the primitive vectors of our reciprocal lattice from the primitive

vectors of the crystal lattice

$$\begin{aligned} \mathbf{a}^\star &= 2\pi \frac{\mathbf{b}\times\mathbf{c}}{\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})} \\ \mathbf{b}^\star &= 2\pi \frac{\mathbf{c}\times\mathbf{a}}{\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})} \\ \mathbf{c}^\star &= 2\pi \frac{\mathbf{a}\times\mathbf{b}}{\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c})} \end{aligned} \tag{3.35}$$

where $\mathbf{a}\cdot(\mathbf{b}\times\mathbf{c}) = \Omega$, the volume of the unit cell. These vectors have the property

$$\mathbf{a}\cdot\mathbf{a}^\star = 2\pi \quad \text{etc.} \tag{3.36}$$

$$\mathbf{a}\cdot\mathbf{b}^\star = 0 \quad \text{etc.} \tag{3.37}$$

and points on the reciprocal lattice are defined by these vectors

$$\mathbf{G} = h\mathbf{a}^\star + k\mathbf{b}^\star + l\mathbf{c}^\star \;\; \forall h,k,l \in \mathbb{Z} \tag{3.38}$$

3.3.4.2 First Brillouin Zone

An important concept in solid–state physics is that of the first Brillouin zone. It is formed by the Wigner–Seitz cell of the reciprocal–lattice (which is the smallest volume repeatable unit in the reciprocal lattice). The reciprocal–space lattice vectors $\{\mathbf{G}\}$ contained within the first Brillouin zone are important as described by Bloch's theorem below.

3.3.4.3 Bloch's Theorem

In a crystal lattice, the potential that the electrons are moving in is periodic with the lattice.

$$V(\mathbf{r}) = V(\mathbf{r}+\mathbf{L}) \tag{3.39}$$

When this is the case the wave–function of the single–particle states existing in this potential can be written as a product of a function which is periodic with the potential and a wave–like part

$$\phi_{\mathbf{k}}(\mathbf{r}) = u_{\mathbf{k}}(\mathbf{r}) \exp(i\mathbf{k}\cdot\mathbf{r}) \tag{3.40}$$

where

$$u_{\mathbf{k}}(\mathbf{r}+\mathbf{L}) = u_{\mathbf{k}}(\mathbf{r}) \tag{3.41}$$

This function can be represented as a sum of plane–waves

$$u_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{k},\mathbf{G}} \exp(i\mathbf{G}\cdot\mathbf{r}) \tag{3.42}$$

where $\{\mathbf{G}\}$ are reciprocal lattice vectors and $c_{\mathbf{k},\mathbf{G}}$ are the co–efficients of this plane–wave set. Re–writing equation 3.40

$$\phi_{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{k},\mathbf{G}} \exp(i(\mathbf{k}+\mathbf{G})\cdot\mathbf{r}) \tag{3.43}$$

so now the single–particle wave–functions, $\phi_{\mathbf{k}}(\mathbf{r})$, can be written as a sum of plane–waves, where $\mathbf{k}$ is a reciprocal vector from within the first Brillouin zone of the system, since any $\mathbf{k}$–vector from outside the first Brillouin zone can be translated back inside the first Brillouin zone.

Proof. If we consider a reciprocal lattice vector $\mathbf{q}$, which is from the set of $\{\mathbf{G}\}$ vectors. Then we can modify equation 3.40 to become

$$\phi_{\mathbf{k}}(\mathbf{r}) = u_{\mathbf{k}}(\mathbf{r}) \exp(i\mathbf{q}\cdot\mathbf{r}) \exp(i(\mathbf{k}-\mathbf{q})\cdot\mathbf{r}) \tag{3.44}$$

The function $u_{\mathbf{k}}(\mathbf{r}) \exp(i\mathbf{q}\cdot\mathbf{r})$ is also a function which is periodic with the lattice, $\tilde{u}_{\mathbf{k}}(\mathbf{r})$, and so now we have a new vector, $\mathbf{k}' = \mathbf{k}-\mathbf{q}$, which is also a valid $\mathbf{k}$–vector. In this way it is possible to translate any $\mathbf{k}$–vector until it is within the first Brillouin zone □

If we consider the electron density, $n(\mathbf{r})$,

$$n(\mathbf{r}) = \sum_{\mathbf{G}} n_{\mathbf{G}} \exp(i\mathbf{G}\cdot\mathbf{r}) \tag{3.45}$$

such that

$$n(\mathbf{r}+\mathbf{L}) = \sum_{\mathbf{G}} n_{\mathbf{G}} \exp(i\mathbf{G}\cdot\mathbf{r}) \exp(i\mathbf{G}\cdot\mathbf{L}) \tag{3.46}$$

where

$$\mathbf{L} = u\mathbf{a} + v\mathbf{b} + w\mathbf{c} \quad \forall u, v, w \in \mathbb{Z} \tag{3.47}$$

but

$$\exp(i\mathbf{G} \cdot \mathbf{L}) = \exp(2\pi i (hu + kv + lw)) = 1 \tag{3.48}$$

since $h, k, l, u, v, w \in \mathbb{Z}$, therefore

$$n(\mathbf{r} + \mathbf{L}) = n(\mathbf{r}) \tag{3.49}$$

so the only non–zero Fourier components are those at the reciprocal lattice points **G**. There will still be a large number of these points!

3.3.4.4 Number of k–points

To accurately sample the first Brillouin zone it may be that a large number of **k**–points are required. While there are some schemes that use cell symmetry to reduce the number of **k**–points required [Monkhorst and Pack, 1976; Pack and Monkhorst, 1977], in a general case there will not be any symmetry to take advantage of.

The density of k–points needs to be chosen through a number of calculations. When converging the energy of the system, the energy may oscillate as the density of **k**–points is increased and so convergence must be ascertained before the accuracy of a calculation can be accepted.

3.3.5 Plane–Waves

There are a number of different basis–sets that can be chosen for $\phi_{\mathbf{k}}(\mathbf{r})$. Two of the most popular are Gaussian Basis–Sets and Plane–Wave Basis–Sets. I will not go into a detailed comparison of Gaussian Basis–Sets to Plane–Wave Basis–Sets, but needless to say there are arguments for both sides. Plane–waves are a good choice for periodic systems for the following reasons.

As can be seen from equation 3.43, the wave–functions of electrons moving in a periodic potential can be represented as sums of plane–waves. To accurately represent the system we would need a large number of plane–waves and some way to determine how many plane–waves are enough. While the wave–function in the space between ions can be represented by a small num-

ber of plane–waves, the wave–functions close to the ionic cores will contain a large number of short–range features and require a large basis–set of plane waves to represent them.

The number of plane waves included in a calculation can be easily characterised by a single parameter, the *cut–off energy*.

$$\frac{|\mathbf{k}+\mathbf{G}|^2}{2} \leq E_{cut} \tag{3.50}$$

By increasing the cut–off energy more and more plane–waves will be included in the basis set. Eventually adding more plane waves will not alter the energy of the system significantly. At this point the energy has been converged with respect to the number of plane–waves. The energy will converge monotonically with increasing numbers of plane–waves.

3.3.5.1 Variable–cell calculations

As was shown in section 2.2.4 the cell of a structure can also be allowed to vary during a calculation. At the start of a calculation the value of E_{cut} will define the number of plane–waves required for the initial cell size. By changing the cell size the reciprocal–lattice also changes, and so will change the size of the Brillouin zone of this calculation and this in turn will change the effective E_{cut}. If the number of plane–waves is kept fixed during a calculation then the accuracy of the answer cannot be guaranteed. Allowing the number of plane–waves to vary during a calculation means that the effective value of E_{cut} can be kept constant.

To be able to perform variable–cell calculations, an optimisation technique such as the BFGS method described in section 2.2.4 needs to be used, since this method is able to cope with an applied strain, $\underline{\underline{\varepsilon}}$. This strain will alter the unit cell by

$$\underline{\underline{h}} = \left(\mathbb{1} + \underline{\underline{\varepsilon}}\right) \underline{\underline{h}}_0 \tag{3.51}$$

where $\underline{\underline{h}} = [\mathbf{a}, \mathbf{b}, \mathbf{c}]$. The BFGS algorithm is able to evolve the unit cell along with the atomic co–ordinates.

3.3.6 Convergence

As has been described in sections 3.3.4.4 and 3.3.5 to be sure of the accuracy of a calculation, the number of k–points and plane–waves the system requires for

a converged solution needs to be determined. When performing DFT calculations the first set of calculations should be single energy calculations on the initial system configuration to determine the correct number of k–points and energy cut–off to use.

The graphs shown in figure 3.5 show an example of convergence plots for a 4–atom diamond unit cell. From these graphs a suitable number of **k**–points would be found by using a 0.05Å^{-1} sampling of reciprocal space and a suitable number of plane–waves for our basis–set would be characterised by a 400 eV plane–wave cut–off energy.

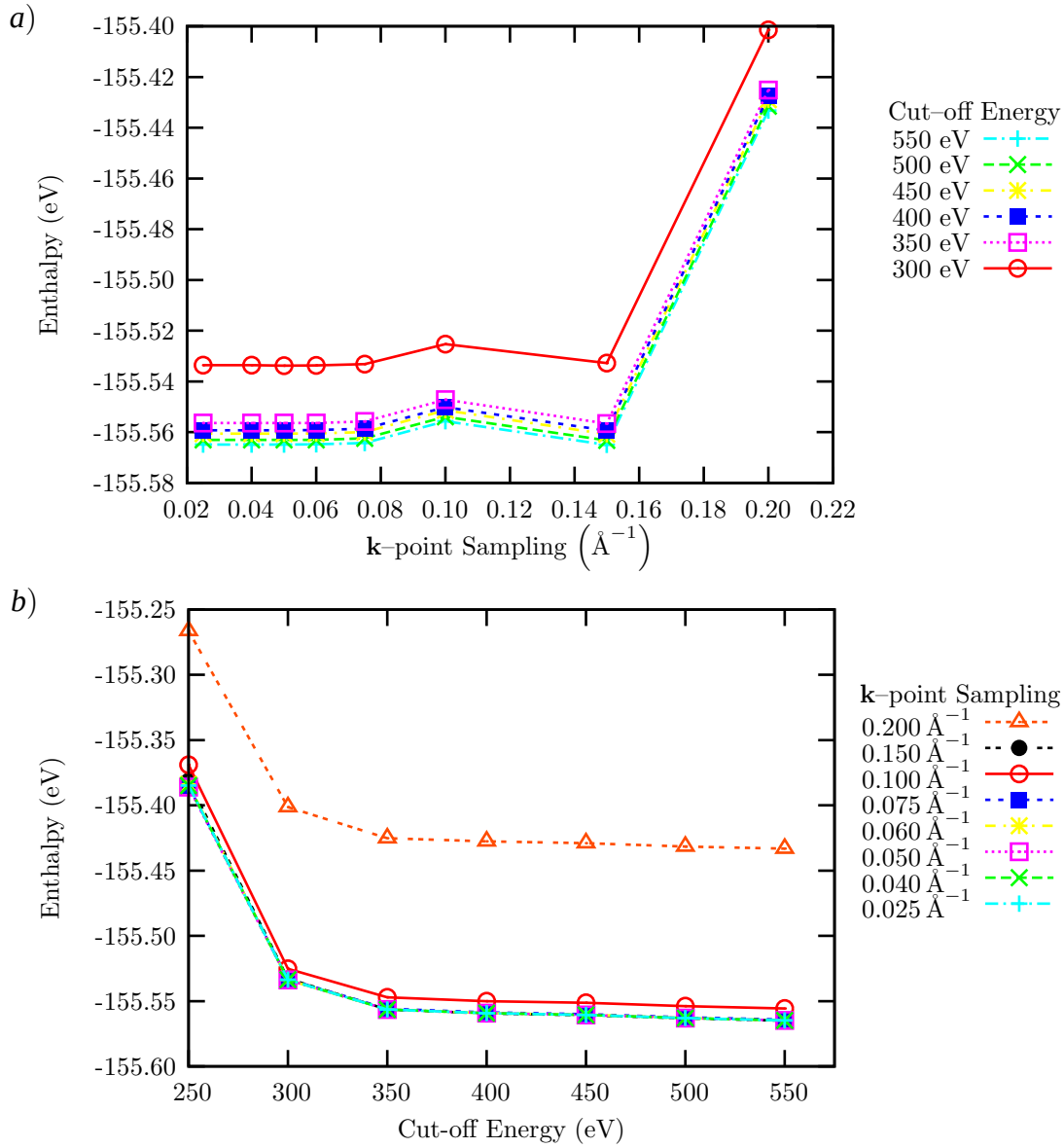

Figure 3.5: Graphs showing *a*) the convergence of the enthalpy of 4–atom diamond with increasing **k**–point density and *b*) the monotonic convergence of the enthalpy of 4–atom diamond with increasing the cut-off energy used in the calculation. The final values chosen for these parameters were a cut-off energy of 400 eV and a **k**–point sampling of reciprocal space of 0.05Å^{-1}.

3.4 Enthalpy

While these methods above describe how to calculate the total energy of the system, I will actually be using *enthalpy*, $\mathcal{H}$, in this study. Since all calculations are being performed at zero temperature this is an appropriate free energy. When at zero pressure it is the energy, $\mathcal{U}$, that defines the potential energy surface that the system explores. Adding the effects of an external pressure will not affect the operation of the BFGS algorithm described in section 2.2.4, since this has been taken into account in the stress term. Now the enthalpy surface of the system,

$$\mathcal{H} = \mathcal{U} + \mathcal{P}\mathcal{V}, \tag{3.52}$$

can be explored, where $\mathcal{P}$ is the pressure on the system and $\mathcal{V}$ is the volume of the simulation super–cell. This has most relevance to chapter 7 when I will be studying systems at different pressures.

3.5 Conclusions

There are many ways of defining empirical potentials so as to make simple approximations to systems of interest that can be simulated in a reasonable timescale, whereas Density Functional Theory is a much more rigorous approach to the calculation of the energy of system. However, in a Genetic Algorithm approach the method only requires that an energy is returned, and the method is independent of the energy's calculation.

Chapter 4

A Genetic Algorithm with Periodic Boundary Conditions

4.1 Methodology

In this chapter, I will demonstrate that for Genetic Algorithm studies with periodic boundary conditions using a periodic cut in the crossover operation is superior to a planar one. The periodic cut is chosen to have the same periodicity as the super–cell of the population member, and this reduces the discontinuities produced in the crossover operation, which would cause extra work for the local minimiser that is required in this GA scheme. Results show that a periodic cut has a faster convergence than a planar one for large systems.

4.1.1 The Problem

As was discussed in the previous chapter, there have been a large number of attempts at solving the minimum structure problem. While the method proposed by Deaven and Ho [1995] is excellent at solving zero–dimensional systems such as clusters, as periodic boundary conditions are imposed in one–, two– and three–dimensions the method could be improved.

By performing cuts in real–space, a version of Holland's schema theorem (section 2.3.4.1) was being utilised. It was the local structure of the members that was being maintained in the crossover procedure. Any disruption caused in the process is then relaxed by the local minimiser, as described in section 2.2. In the cluster case the only disruption is caused along the cut itself. Relaxation is required for the surface of the cluster, but this would already have been done before crossover.

When periodic boundary conditions are introduced, and the cut procedure does not take them into account, then in effect at least *two* cuts are being made, one along the cut itself, and the others along the periodic boundaries. These extra cuts cause extra disruption to the local structure and so will hamper convergence to the global minimum structure. This is not to say that the method will not find the global minimum structure, but that this method could be improved. Figure 4.1 shows the crossover operation using a planar cut in a super–cell approach.

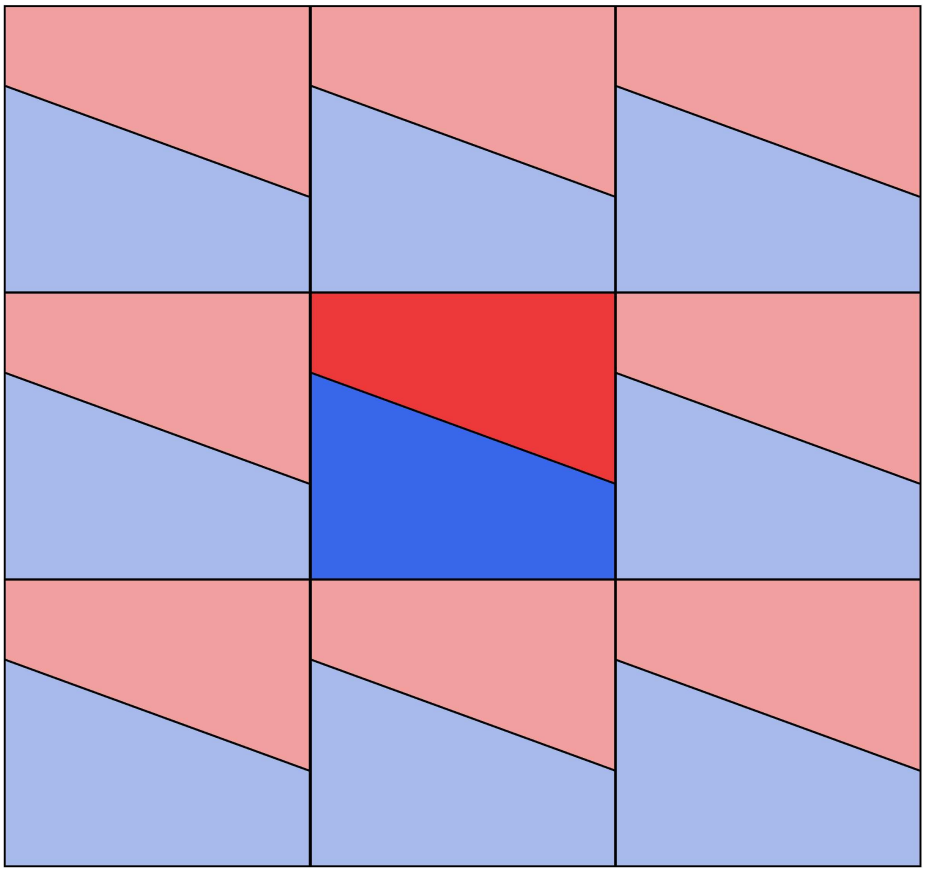

Figure 4.1: Diagram showing the crossover operation using a planar cut with periodic images also shown. The red section is from one parent and the blue from the other. Apart from the disruption caused along the cut there is also disruption along the periodic boundaries.

4.1.2 The Solution

The solution to these extra discontinuities is to use periodic cuts in the crossover operation. By forcing the cuts to be periodic with the super–cell then unwanted disruption can be reduced.

There are also some extra improvements that can be added so that the method can be made more flexible. Instead of the cuts being made in Cartesian co–

ordinates as previous studies have done, the cuts are made in the fractional co–ordinates of the ions, made with reference to the simulation super–cell, so it does not *"know"* about the Cartesian shape of the cell. While this initially sounds like an unnecessary complication, it allows for population members with different cells to be bred during crossover, rather than being constrained to both parents having the same super–cell. The technique also allows the cell size and shape to be evolved along with the crystal structure (if the local minimiser can also optimise the cell along with the ionic positions). This part of the technique will be discussed in more detail in chapter 6.

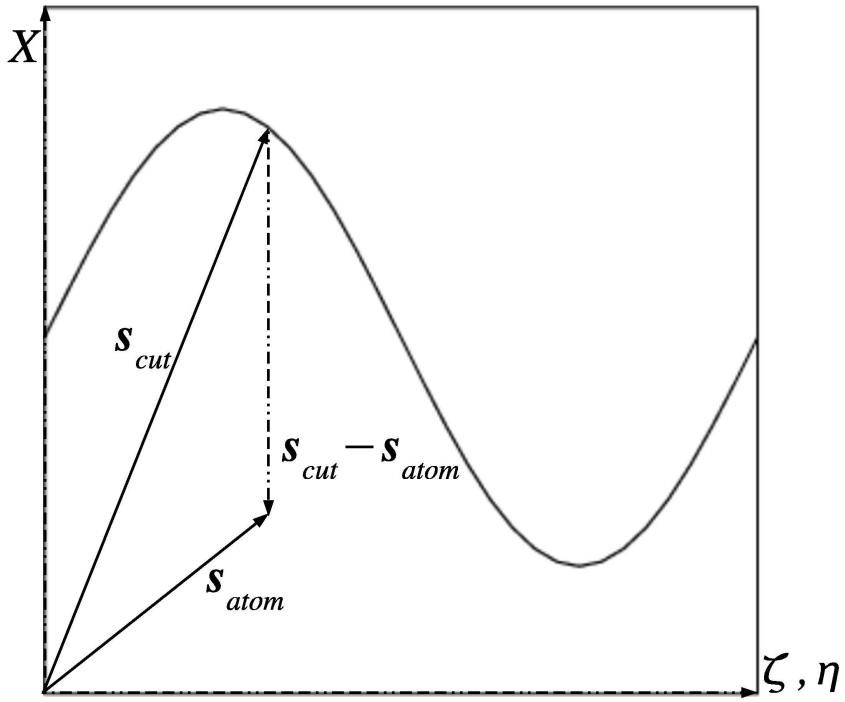

Figure 4.2: Diagram showing crossover in a fractional representation. For each (ζ, η) = either $(\mathbf{a},\mathbf{b})$, $(\mathbf{b},\mathbf{c})$ or $(\mathbf{c},\mathbf{a})$ then $\mathbf{X}$ = either $\mathbf{c}$, $\mathbf{a}$ or $\mathbf{b}$.

The crossover is performed in fractional coordinates, as described in figure 4.2. The cut is defined by any periodic function with the same periodicity as the cell, $\mathbf{f}\left(\mathbf{s}_{atom}^{(\zeta,\eta)}\right)$, where $\mathbf{s}_{atom}^{(\zeta,\eta)}$ is the fractional position vector for each atom along the $(\zeta,\eta) = (\mathbf{a},\mathbf{b})$, $(\mathbf{b},\mathbf{c})$ or $(\mathbf{c},\mathbf{a})$ directions. This function gives a vector (in fractional coordinates) $\mathbf{s}_{cut}$ for each $\mathbf{s}_{atom}$ in the population member. The metric tensor $\underline{\underline{g}} = \underline{\underline{h}}^{\mathrm{T}}\underline{\underline{h}}$, where $\underline{\underline{h}} = [\mathbf{a},\mathbf{b},\mathbf{c}]$ (in Cartesian coordinates), is used to calculate the product

$$\alpha_{cut} = \left(\mathbf{s}_{cut} - \mathbf{s}_{atom}\right)^{\mathrm{T}} \underline{\underline{g}}\,\mathbf{X} \tag{4.1}$$

where $\mathbf{X} = \begin{cases} \mathbf{a} & = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ \mathbf{b} & = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \mathbf{c} & = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{cases}$ in fractional coordinates ($\mathbf{X} = \mathbf{c}$ when $(\zeta, \eta) = (\mathbf{a}, \mathbf{b})$ etc.)
and with the criterion

$$\alpha_{cut} \begin{cases} > 0 & \text{the atom is } \textit{"above"} \text{ (outside) the cut} \\ \leq 0 & \text{the atom is } \textit{"below"} \text{ (inside) the cut.} \end{cases} \tag{4.2}$$

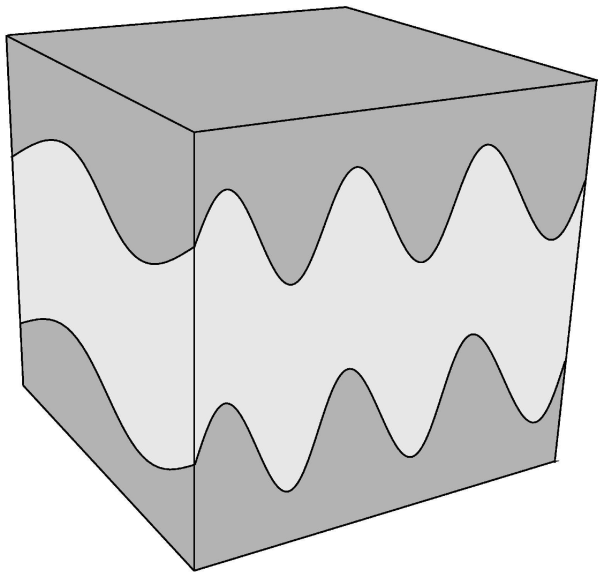

Figure 4.3: Real–space representation of the periodic cuts in the crossover operation. Different wavelengths and amplitudes can be used for the cuts along the different cell directions. The cuts are calculated in fractional coordinates which allows crossover between parents with different cells. The dark grey sections represent one part of the cell, the light grey the other, and it is these parts that are swapped in crossover.

Figure 4.3 shows the crossover operation where two cuts are required in the cell. The reason for using two cuts can be seen in figure 4.4. A single cut would still have discontinuities at one of the periodic boundaries.

Every crossover operation performed has an equal probability of being calculated with the cuts made in reference to either the **a**, **b** or **c** directions. This ensures that none of the three co–ordinate directions is preferred over the other two, but also allows large areas of each of the population members to be undisturbed by the cut. To ensure that no one parent is preferred over the other, the centre of each of the cuts should be made one–quarter and three–quarters up the chosen cell axis which gives approximately even mixing.

In the Deaven and Ho [1995] formulation, the plane of the cut was defined

by the creation of a random unit vector on the surface of a sphere which was centred on the centre of mass of the cluster. In our method cuts made in different cell directions can have different random wavelengths and amplitudes, although a maximum amplitude should be defined so that the cut is contained within the simulation super–cell, and obviously the wavelength of the cut cannot be longer than twice the cell vector in that direction, whilst any cut with a wavelength of less than half the atomic separation will appear as a flat plane.

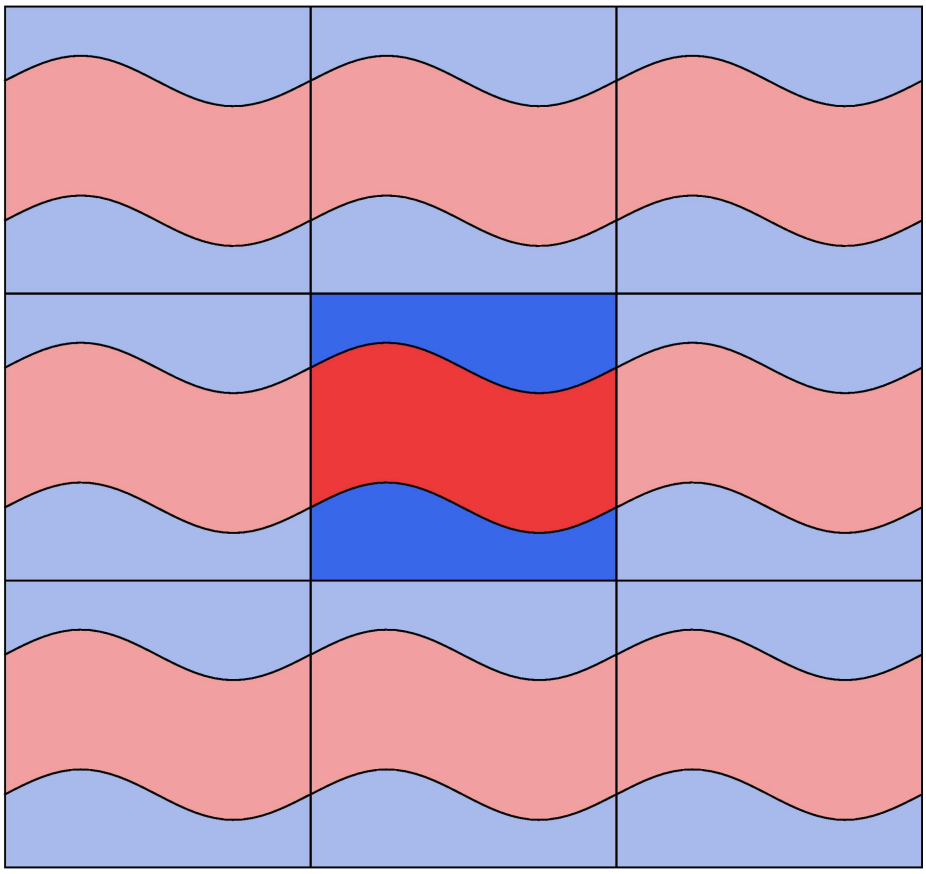

Figure 4.4: Diagram showing the crossover operation using periodic cuts with periodic images also shown. The red section is from one parent and the blue from the other. The only disruption that occurs is along the cuts themselves.

Fitness is determined by the relative enthalpy per atom of the population members, and each population member was chosen for crossover based on its fitness using roulette wheel selection [Johnston, 2003]. Similar to Chuang *et al.* [2004] the number of atoms in each individual population member can be varied by accepting all solutions after crossover, or if the number of atoms needs to be constrained then solutions are rejected until offspring are generated that have the correct number.

Either elitist or hybrid selection (as defined in section 2.3.3) was used in the update procedure, which determines which members of the original population

should progress through to the next generation.

4.2 Validation on Empirical Systems

All calculations were performed by adding the above GA formulation to the *ab initio* plane–wave DFT code CASTEP [Segall *et al.*, 2002], which has also been modified for ease of algorithm testing to allow the use of the empirical Lennard–Jones potential (section 3.2.2). This potential gives rise to a HCP ground state structure [Pollack, 1964a] which is almost degenerate with the FCC structure (energy difference from HCP $+0.1\%$ [Kane and Goeppert-Mayer, 1940]. The energy difference between the FCC and HCP super–cells used in this study was $+0.072\%$, due to the above formulation of the Lennard–Jones potential). The value of σ was set to 3.405 Å, ε was set to 120 K, and R_{cut} was set to 2.5σ. While the ground states are very close in energy, to switch from FCC to HCP four out of every six layers require a stacking fault.

4.2.1 Results

For the Lennard–Jones results a fixed super–cell was used, but the number of atoms could either be fixed for each of the population members, or be allowed to vary. GA minimisation calculations were performed using either the planar or periodic cuts. If a planar cut was taken then a unit vector on the surface of a sphere was generated, as in the Deaven and Ho [1995] case, which defined the cut.

4.2.1.1 Choice of initial conditions

The number of population members was fixed at $M = 16$, and the initial number of atoms in each population member was set to $N = 150$ using a hexagonal super–cell, which is commensurate with both perfect FCC and HCP structures without stacking faults. The initial configuration of the population members is totally randomised, then minimised with the local minimiser before proceeding. A total of 200 generations was run for each simulation.

To determine what the initial values of the mutation rate, mutation amplitude were I set off a series runs with differing values of these parameters. Periodic cuts and hybrid selection were chosen for these runs, and the initial starting values of the mutation rate were $m_R = 0.05, 0.10$ and 0.15. For each of these value I set the value of $m_A = 1.0$Å$, 2.5$Å$, 5.0$Å and 10.0Å. Of these runs, the

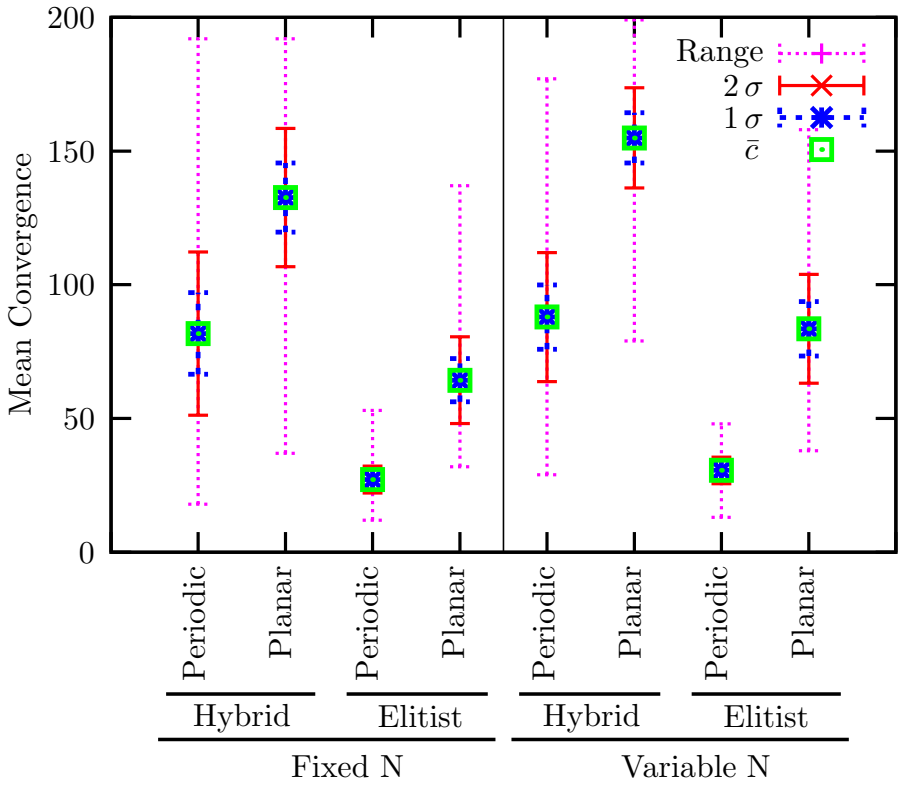

Figure 4.5: Summary of the convergence times (in number of generations) for each variation of the method presented. For each run, either periodic or planar cuts were used, using either the hybrid or the elitist update scheme, and the number of atoms could either be kept fixed, or be allowed to vary.

only set of values that returned an ordered minimised structure was with m_R set to 0.10 (10%) and m_A set to 2.5Å.

In total 15 simulations were performed from a random start for each of the eight combinations of either fixed or variable number of atoms, using either the roulette wheel or simple update scheme, and with crossover performed using either periodic cuts or a planar cut. A summary of the convergence times is shown in figure 4.5.

4.2.1.2 Empirical Lennard–Jones bulk studies with a fixed number of atoms.

For these studies the number of atoms was kept fixed at 150 during the whole of the simulation. However, none of the 60 calculations resulted in minimisation down to FCC or HCP stacking. The elitist update scheme was much faster at reaching convergence, and using periodic cuts was much faster than using a planar cut when using either update scheme.

As noted in the studies of Chuang *et al.* [2004] allowing the number of atoms to vary helped convergence. In that study ordered structures were found even

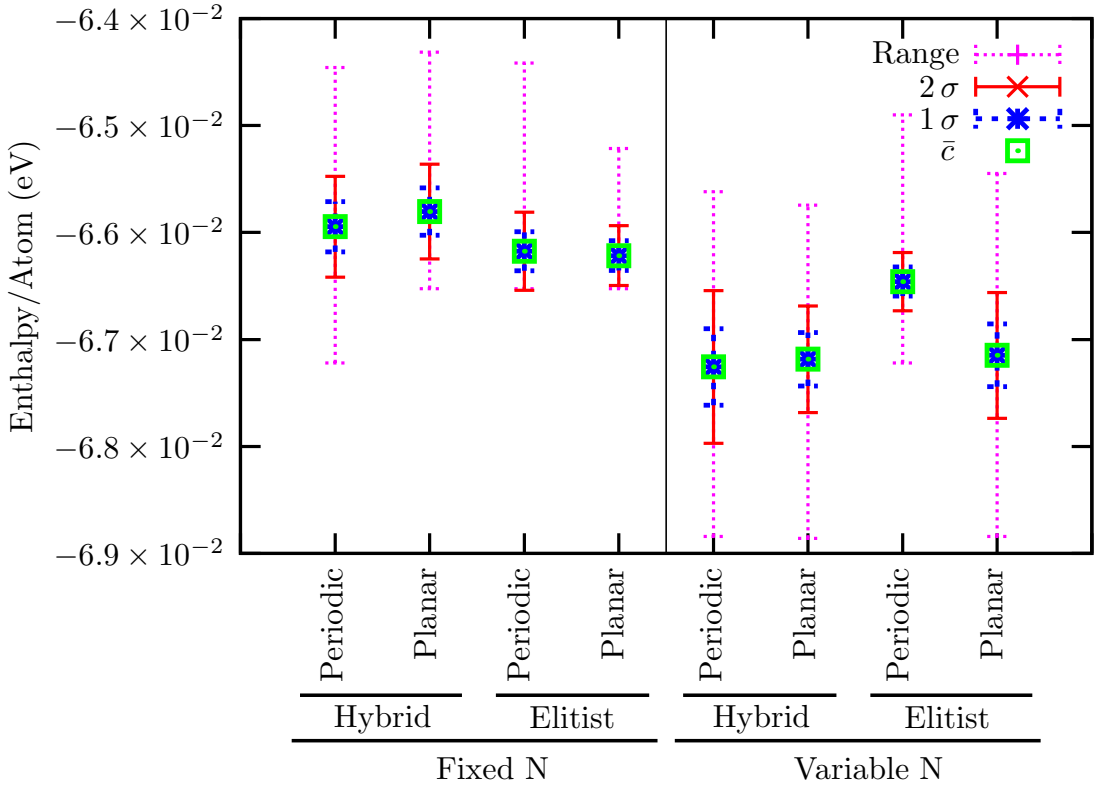

Figure 4.6: Summary of the enthalpies of the structures found for the results shown in figure 4.5

in a fixed–N case. This may be due to the extra constraints imposed by surface effects as opposed to performing a fully bulk calculation.

4.2.1.3 Empirical Lennard–Jones bulk studies with a variable number of atoms.

The trend in the results obtained from varying the number of atoms agree with the trend seen in the study of Chuang *et al*. [2004], where allowing the number of atoms to vary during a calculation improved the convergence of the calculation. The use of periodic cuts using the hybrid update scheme, or the planar cut using the roulette wheel or simple update scheme, allowed the system to be minimised to a defect–free ground state structure [1]. Periodic cuts were faster to convergence than a planar cut, as shown above.

We found that the system did not converge into a perfect lattice structure without allowing for variable atom number. Figure 4.7 shows a typical set of results, using periodic cuts with roulette wheel selection for update. In this case the system converged in 29 generations to the structure shown in figure 4.8. This configuration is an FCC–HCP hybrid with an energy difference

[1] a structure with no vacancies, although there were stacking faults

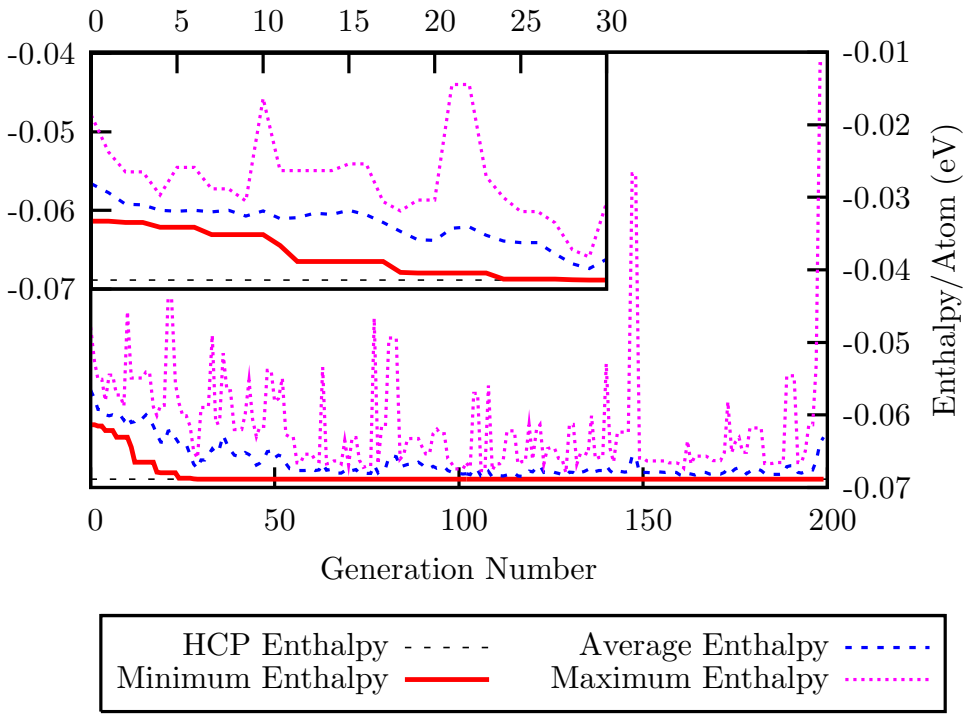

Figure 4.7: Typical results from a 150 atom (variable), 16 population member calculation starting from an initial random configuration with a mutation rate of 10% and mutation amplitude of 2.5Å, using roulette wheel selection in the update procedure. Periodic cuts were used and the system converged in 29 generations to a structure with 150 atoms and a local minimum enthalpy +0.024% above the HCP minimum (see insert).

of +0.024% from the HCP ground state, which is due to a single FCC plane stacking fault. Similar structures were also found using a planar cut, but with longer convergence times. A pure HCP structure was found only once, in a calculation performed with a planar cut in the the crossover procedure.

4.2.1.4 Cell–Atom Coupling

It is also interesting to note that the planes of atoms have aligned themselves with the cell axis, which was unexpected. The super–cell with periodic images is a computational construct for ease of simulation, and the structure should take any orientation that fulfils the periodicity. Indeed, the first occurrence of an ordered structure such as that in figure 4.8 is usually at a random orientation to the cell axis, but the periodic cuts, which are made in reference to one of the three axes, provide a weak coupling to the atomic positions. This causes the structure to rotate until it is aligned correctly. With the planes aligned this way with the cell axes the crossover procedure will cause the minimum amount of disruption to the structure.

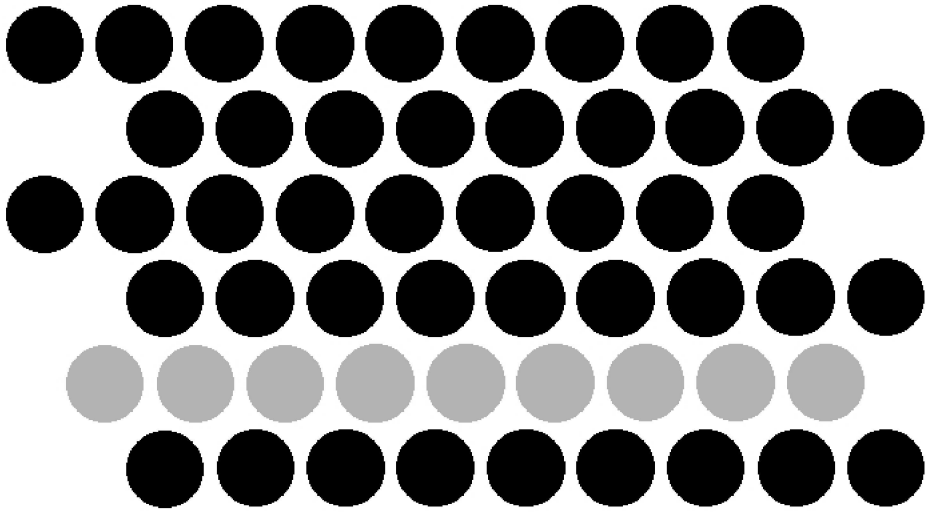

Figure 4.8: Side on view of the minimised structure from figure 4.7, looking down the $[0\bar{1}1]$ direction. The colours show the mixture of FCC (grey) and HCP (black) stacking.

4.3 Conclusions

Even though the structure found in this case was a defect–free structure (figure 4.8), there was one lower enthalpy configuration, that of pure HCP, which was only found once, by using a planar cut with hybrid selection for update. Only three defect–free structures were found using these options however, one HCP, one structure similar to figure 4.7 and one pure FCC. By comparison six defect–free structures were found using a periodic cut.

Although results from calculations performed with a fixed number of atoms are not as encouraging as those presented by Chuang *et al*. [2004], results from calculations performed with a variable number of atoms show that the use of a periodic cut decreases the number of generations required to reach a well–ordered structure, and the quality of the results are comparable between the two crossover techniques.

The one exception to this last statement is the case of elitist selection with a variable number of atoms using periodic cuts. These results show a higher average energy than the planar cut case (figure 4.6). I believe that this can be interpreted by referring back to the schema theorem (section 2.3.4.1) for binary strings. The use of periodic cuts is very good at maintaining local structure during crossover so that when only the fittest members are selected for update (as in elitist selection) this system becomes too easily frozen in local minima. By contrast, planar cuts are so bad at maintaining local structure in periodic

Atom Number	**Update Type**	**Cut Type**	**Pure HCP**	**Intermediate HCP–FCC**	**Pure FCC**
Variable	Hybrid	Periodic	0	6	0
		Planar	1	1	1
	Elitist	Periodic	0	0	0
		Planar	0	3	1

Table 4.1: Table giving summarising the number of each ordered structure type found for the different methods discussed. The results for fixed atom number have not been included. Numbers given are out of a total of 15 calculations.

systems that even elitist selection cannot cause rapid stagnation because the disruption is large enough to force the system out of local minima.

While this effect could be seen as beneficial, the size of this effect cannot be controlled as easily as with the mutation rate and mutation amplitude. It is better therefore to use periodic cuts and have more control through these parameters.

Table 4.1 summarises the results found with the variable atom number calculations. Although HCP was found by using a planar cut, the most successful method was to use periodic cuts, especially when taking the time taken to achieve convergence into account. I shall address the failure of the method to find HCP in the next chapter.

Chapter 5

The Structure Factor and its uses

5.1 Introduction: The Stagnation Problem

As could be seen figure 4.7, even though the system converged in 29 generations, no lower enthalpy structures were found. This is a common occurrence in GA calculations, that of *stagnation*. The system will find itself in a local minimum, and is unable to find itself out of it, since mutation is unable to create any new features that are beneficial. Over a long enough time frame the system may work itself out of this local minimum, but this may take a large number of generations.

What is needed is some way of determining, during the course of a calculation, if two structures are similar. If this were possible then the selection procedure could be modified so that dissimilar structures would be chosen over similar ones. This should prevent stagnation and aid convergence.

One such method is that proposed by Kabsch [1976, 1978] which determines the rotation matrix between two structures, along with a measure of how close those two structures are. However, this method requires knowledge of the ordering of the atoms within the arrays that define them. If the matching of atoms is not known, the process to order the atoms and test all the possible ordering is factorial. While the information given by this method would be ideal it is unfeasible for any structures with more than a few atoms. Instead the members structure factor can be used as a comparison, once a few difficulties have been overcome.

5.2 Definition

The structure factor is a mathematical description of how a crystal scatters radiation which is incident upon it. It is defined as

$$F(\mathbf{k}) = \int \rho(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}} \mathrm{d}^3 r \tag{5.1}$$

where $\mathbf{k} = \mathbf{s} - \mathbf{s}_0$ is the scattering vector, where $\mathbf{s}_0$ is the wave–vector of the incident radiation, and $\mathbf{s}$ is the wave–vector of the scattered radiation. These vectors are at an angle of 2θ to each other, and $|\mathbf{s}| = |\mathbf{s}_0| = \frac{1}{\lambda}$ where λ is the wavelength of the incident radation.

If $\rho(\mathbf{r}) \rightarrow \rho'(n)\,\delta(\mathbf{r} - \mathbf{r}_n)$ where $\mathbf{r}_n$ is the position of ion n and $\rho'(n)$ is the scattering factor of that ion, the structure factor becomes

$$F(\mathbf{k}) = V \sum_n \rho'(n) \exp(2\pi i \mathbf{k}\cdot\mathbf{r}_n). \tag{5.2}$$

The scattered intensity is

$$\begin{aligned} I(\mathbf{k}) &= |F(\mathbf{k})|^2 \\ &= F^*(\mathbf{k}) F(\mathbf{k}) \\ &= V^2 \sum_n \sum_m \rho'(n)\rho'(m) \exp(2\pi i \mathbf{k}\cdot\mathbf{r}_n) \exp(-2\pi i \mathbf{k}\cdot\mathbf{r}_m) \\ &= V^2 \sum_n \sum_m \rho'(n)\rho'(m) \exp(2\pi i \mathbf{k}\cdot(\mathbf{r}_n - \mathbf{r}_m)) \end{aligned} \tag{5.3}$$

Setting $\mathbf{r}_n - \mathbf{r}_m = \mathbf{R}_{nm}$ now gives

$$I(\mathbf{k}) = V^2 \sum_n \sum_m \rho'(n)\rho'(m) \exp(2\pi i \mathbf{k}\cdot\mathbf{R}_{nm}) \tag{5.4}$$

The dot–product term becomes

$$\mathbf{k}\cdot\mathbf{R}_{nm} = \frac{2\sin\theta}{\lambda} R_{nm}\cos\phi \tag{5.5}$$

and setting

$$k' = \frac{4\pi\sin\theta}{\lambda} \tag{5.6}$$

gives

$$I\left(k'\right) = V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \exp\left(ik' R_{nm} \cos\phi\right) \tag{5.7}$$

The probability of $\mathbf{R}_{nm}$ being at an angle ϕ with $\mathbf{k}$ is

$$\left(2\pi R_{nm} \sin\phi\right) R_{nm} \mathrm{d}\phi$$

and the total probability of $\mathbf{R}_{nm}$ being in any direction is $4\pi R_{nm}^2$.
Taking the average of $I\left(k'\right)$ now gives

$$\left\langle I\left(k'\right)\right\rangle = V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \frac{1}{4\pi R_{nm}^2} \int_0^\pi \mathrm{d}\phi \, \exp\left(ik' R_{nm} \cos\phi\right) 2\pi R_{nm}^2 \sin\phi \tag{5.8}$$

To solve this integral, let

$$\begin{aligned} y &= \exp\left(ik' R_{nm} \cos\phi\right) \\ \mathrm{d}y &= -ik' R_{nm} \sin\phi \exp\left(ik' R_{nm} \cos\phi\right) \mathrm{d}\phi \end{aligned}$$

so that

$$\begin{aligned} \left\langle I\left(k'\right)\right\rangle &= V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \int_{\phi=0}^{\phi=\pi} \frac{\mathrm{d}y}{2ik' R_{nm}} \\ &= V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \left[\frac{-y}{2ik' R_{nm}}\right]_{\phi=0}^{\phi=\pi} \\ &= V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \frac{1}{k' R_{nm}} \left(\frac{\exp\left(ik' R_{nm}\right) - \exp\left(-ik' R_{nm}\right)}{2i}\right) \end{aligned} \tag{5.9}$$

and using the identity

$$\sin\psi = \frac{\exp\left(i\psi\right) - \exp\left(-i\psi\right)}{2i}$$

this gives

$$\left\langle I\left(k'\right)\right\rangle = V^2 \sum_n \sum_m \rho'\left(n\right) \rho'\left(m\right) \frac{\sin\left(k' R_{nm}\right)}{k' R_{nm}} \tag{5.10}$$

Where $R_{nm} = |\mathbf{r}_n - \mathbf{r}_m|$ and $k' = \frac{4\pi \sin\theta}{\lambda}$. Equation 5.10 can also be written as

$$\langle I\left(k'\right)\rangle = V^2\left[N\sum_{n=1}^{N}\rho'^2\left(n\right) + \ 2\sum_{n=1}^{N}\sum_{m>n}^{N}\rho'\left(n\right)\rho'\left(m\right)\frac{\sin\left(k'|\mathbf{r}_n-\mathbf{r}_m|\right)}{k'|\mathbf{r}_n-\mathbf{r}_m|}\right] \tag{5.11}$$

This function,

$$\frac{\sin\psi}{\psi} = j_0\left(\psi\right)$$

is the zeroth order spherical Bessel function of the first kind. The important property of equation 5.11 is that it is a function of inter–atomic distances within the crystal, $|\mathbf{r}_n-\mathbf{r}_m|$, and is therefore independent of origin. This is the Debye Scattering Formula [Debye, 1915].

A function with similar properties is

$$\boxed{\Lambda\left(k_r\right) = V^2\left[N\sum_{n=1}^{N}\rho'^2\left(n\right) + 2\sum_{n=1}^{N}\sum_{m>n}^{N}\rho'\left(n\right)\rho'\left(m\right)J_0\left(\sqrt{3}\pi k_r|\mathbf{r}_n-\mathbf{r}_m|\right)\right]} \tag{5.12}$$

which instead uses the zeroth order Bessel function of the first kind. It is this "structure factor" function which I shall use for this chapter. The function $\Lambda\left(k_r\right)$ is positive definite.

5.2.0.5 Comparing Structure Factors

A function suitable for comparing two structure factors is [see Srinivasan and Parthasarathy, 1976, chapter 6]

$$R\left(F\left(\mathbf{k}\right)\right) = \frac{\sum_{\mathbf{k}}\left||F'\left(\mathbf{k}\right)| - |F\left(\mathbf{k}\right)|\right|}{\sum_{\mathbf{k}}|F'\left(\mathbf{k}\right)|} \tag{5.13}$$

which would give a value that would correctly identify translations, but not rotations. However, I am using $\Lambda\left(k_r\right)$, not $F\left(\mathbf{k}\right)$. Replacing this in equation 5.13 gives

$$R\left(\Lambda\left(k_r\right)\right) = \frac{\sum_{k_r}|\Lambda'\left(k_r\right) - \Lambda\left(k_r\right)|}{\sum_{k_r}\Lambda'\left(k_r\right)} \tag{5.14}$$

which should still be suitable for use. Here $\Lambda'\left(k_r\right)$ is the structure factor of the "known" structure, and $\Lambda\left(k_r\right)$ is the structure factor of the structure being compared against this structure.

The behaviour of this function is different whether or not you are dealing with a fixed or a variable cell case. In a fixed cell calculation the trend of $R(\Lambda(k_r))$ can be seen in figure 5.1

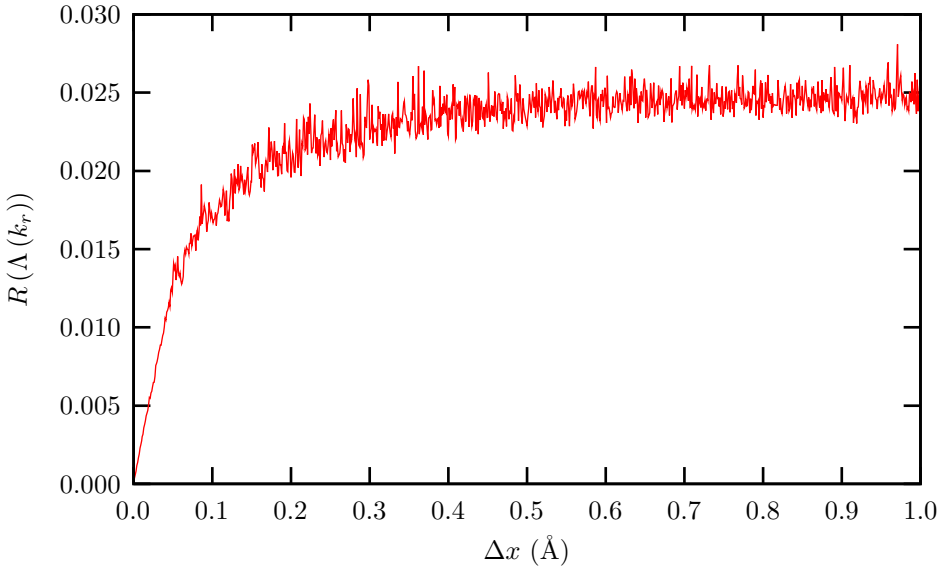

Figure 5.1: Graph showing the trend of $R(\Lambda(k_r))$ for a fixed cell calculation. Behaviour calculated for a 4–atom carbon cell.

In contrast, the behaviour for variable cell calculations can be seen in figure 5.2.

In both these plots Δx is the random displacement from the original structure, whose comparison function is given by $\Lambda(k_r)$.

5.3 A Two–Part Fitness Function

The use of this structure factor comparison, given in equation 5.14, is incorporated into the fitness function such that fitness is now made up of two parts, one part from the enthalpy as previously used, e.g., one of the definitions of f_i from equation 2.11, and the new value from equation 5.14, so now

$$f_i' = (1-w)\, f_i + wR(\Lambda(k_r)) \tag{5.15}$$

where f_i' is the value of the fitness that will be used in the update and crossover procedures, and w is a weighting factor which has a value between zero and one.

Since we are using this routine to differentiate between like and unlike structures, rather than any form of comprehensive structural analysis, we can simplify this comparison somewhat. If we are performing a variable number of

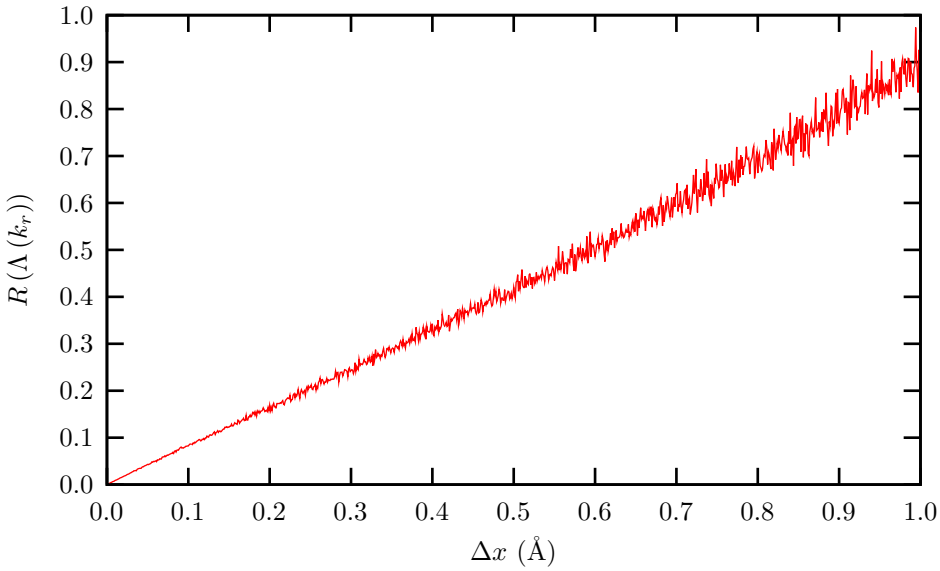

Figure 5.2: Graph showing the trend of $R(\Lambda(k_r))$ for a variable cell calculation. Behaviour calculated for a 4–atom carbon cell where the cell also had an associated Δx modification.

atom calculation, we can guess that two structures with different numbers of atoms are different, so we will have no need to compare these structures. We also do not need to compare all structures with all other structures, since we are merely trying to prevent stagnation, and so we can simply compare all structures with the minimum enthalpy structure which has the same number of atoms. We also want to keep the fact that lower enthalpy structures are "better" than higher enthalpy ones, so we further weight the value of $R(\Lambda(k_r))$ that a structure has by the value of f_i of the fittest member in that "set" which is made up of members with the same number of atoms. So now the new fitness function is

$$ {}^{j}f_i' = (1-w)\,{}^{j}f_i + w\,{}^{j}f_{fit} * \begin{cases} 1 & {}^{j}f_i \equiv {}^{j}f_{fit} \\ R(\Lambda(k_r)) & {}^{j}f_i \not\equiv {}^{j}f_{fit} \end{cases} \tag{5.16} $$

Where the left–superscript j above denotes comparing between groups with similar number of atoms only. This means that the fitness of the fittest member of each group ($^{j}f_{fit}$) will be unchanged from its enthalpy value, and all other values in the group will be scaled accordingly. If the value of the fitness weight, w, is set to 1 then the maximum value of $^{j}f_i'$ that any member could have is the same value of the fittest member of the group, $^{j}f_{fit}$.

Since in a fixed cell case the value of $R(\Lambda(k_r))$ is between zero and one f_i' will remain between zero and one, as in the enthalpy case. However, as could be seen in the trend from the variable cell case (figure 5.2) in this case $R(\Lambda(k_r))$

could become greater than one, in which case it would become more "fit" than the fittest member with the same number of atoms. The solution to this is to set any member with a value of $R(\Lambda(k_r)) \geq 1$ equal to one. Now the condition that $0 \leq f_i' \leq 1$ is satisfied.

5.4 Validation on Empirical Systems

5.4.1 Results

5.4.1.1 Results from the use of the Structure Factor Comparison

The use of the structure factor in the selection procedure has had a marked effect on the quality of the results produced as seen in figure 5.3. The use of the structure factor drives the system so that for a fitness weight of 0.75 finding a HCP structure is much more likely.

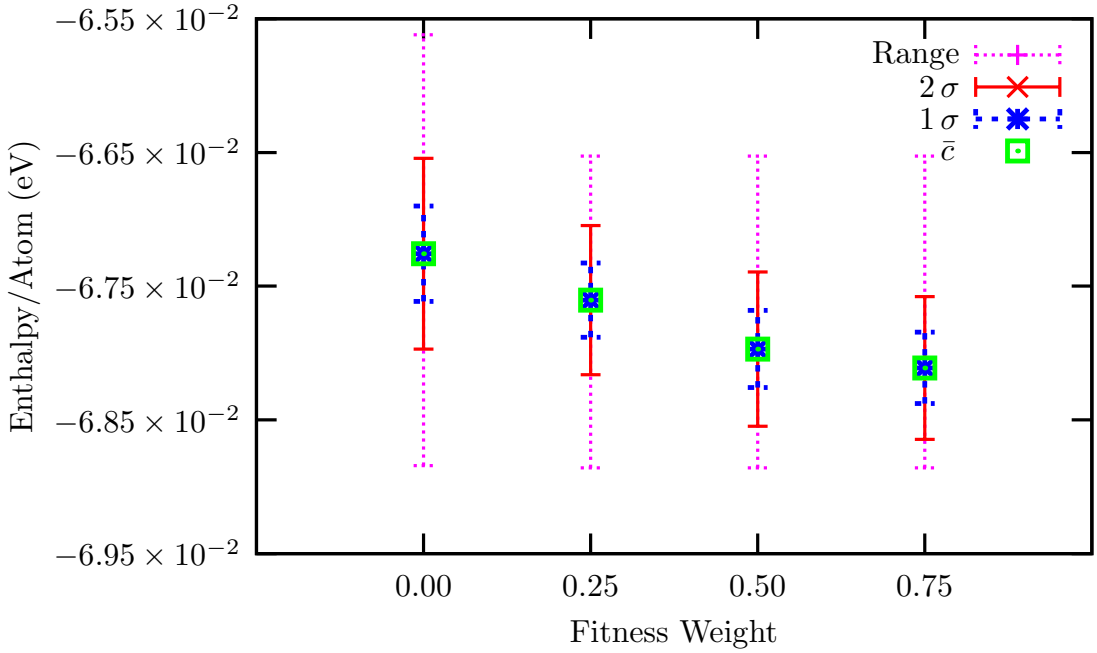

Figure 5.3: Summary of the enthalpies of the structures found for different fitness weights, which controls how much the structure factor is considered during selection for update and crossover. The values for $w = 0.0$ are those from figure 4.6.

The effect on convergence is interesting however, as seen in figure 5.4. There is little increase in the time to convergence, although there is a greater spread in the values.

Figure 5.5 shows the results from a calculation performed with $w = 0.75$. I have included these results in particular because it shows the system going

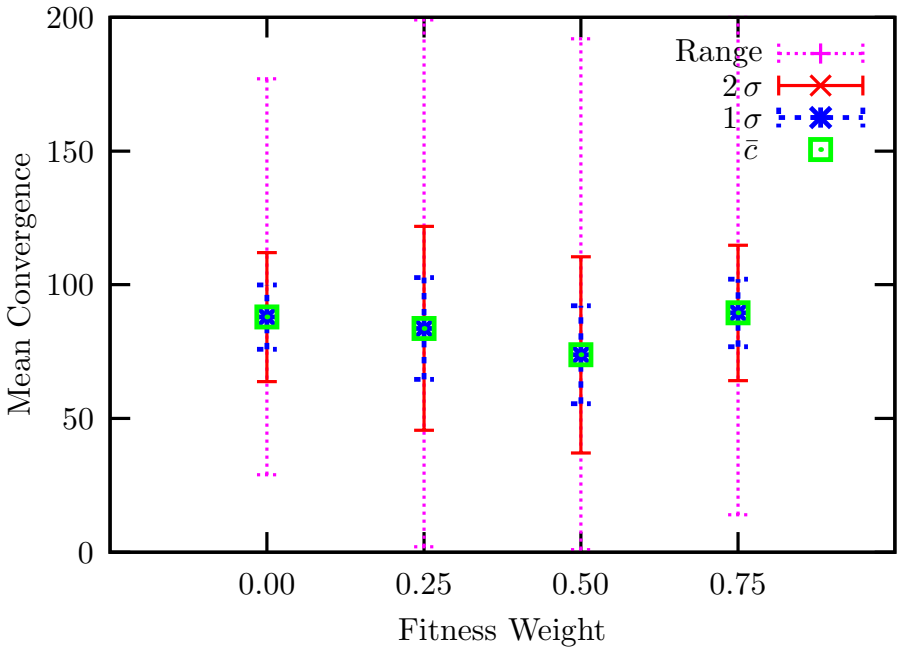

Figure 5.4: Summary of the convergence times for the results shown in figure 5.3. The values for $w = 0.0$ are those from figure 4.5.

from a FCC structure to a HCP structure through two intermediate stacking fault structures. All these structures can be seen in figure 5.6.

5.5 Conclusions

The effect that the use of the structure has on a calculation is not exactly as I originally envisaged. I had hoped that the modified fitness function (equation 5.16) would prevent structures dominating and as such prevent stagnation. I also thought that this may mean that time to convergence would have increased, and although the maximum value is greater and the overall spread is larger, as seen in figure 5.4, the average has not changed greatly, so the time required for a calculation has not increased. The actual calculation of the structure factor (equation 5.12) does take time, but this time is much less than the time take for an energy calculation in an *ab initio* DFT framework.

As can be seen from figure 5.5, "stagnation" of a type occurs. In this case the structure that was the minimum enthalpy structure was in fact the global minimum configuration, but structural diversity was not maintained throughout the rest of the population. In this, and in other calculations, a very well ordered structure such as structure "*c*" from figure 5.6 could dominate and this structure would propagate through the population. When this occurs, and the whole population is the same structure, the program will exit the calculation

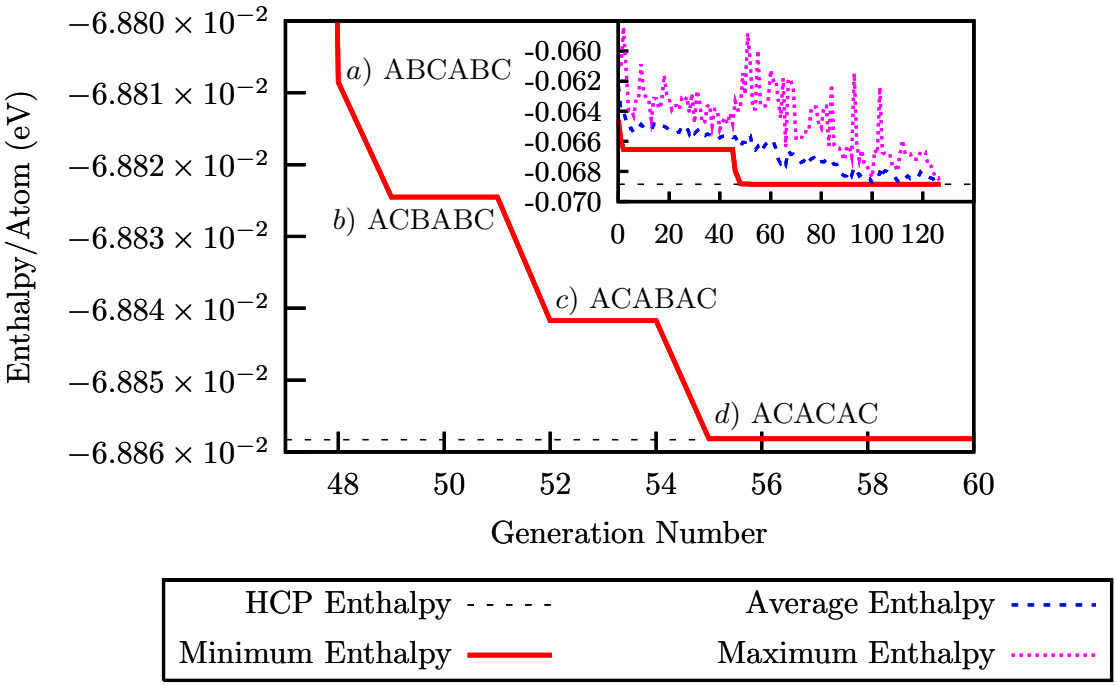

Figure 5.5: Plot showing convergence to HCP minimum structure for a calculation with $w = 0.75$. The stacking patterns of the minimum enthalpy solutions are shown next to their appearance during the course of the simulation. The system converged to a HCP structure in 55 generations, and by the 127^{th} generation all members were the same. Structures a), b), c) and d) are shown in figure 5.6

early.

Fitness Weight	**Pure HCP**	**Intermediate HCP–FCC**	**Pure FCC**
0.00	0	6	0
0.25	3	3	0
0.50	3	6	0
0.75	6	3	0

Table 5.1: Table comparing the number of each ordered structure type found for different values of the fitness weighting factor w. Numbers given are out of a total of 15 calculations.

Table 5.1 summarises the success of the structure factor in the fitness function, when compared with table 4.1 from chapter 4. The number of ordered structures increases, and the probability of finding the global minimum structure also increases. Incorporation of the structure factor into the fitness function can be seen as a useful addition to the method.

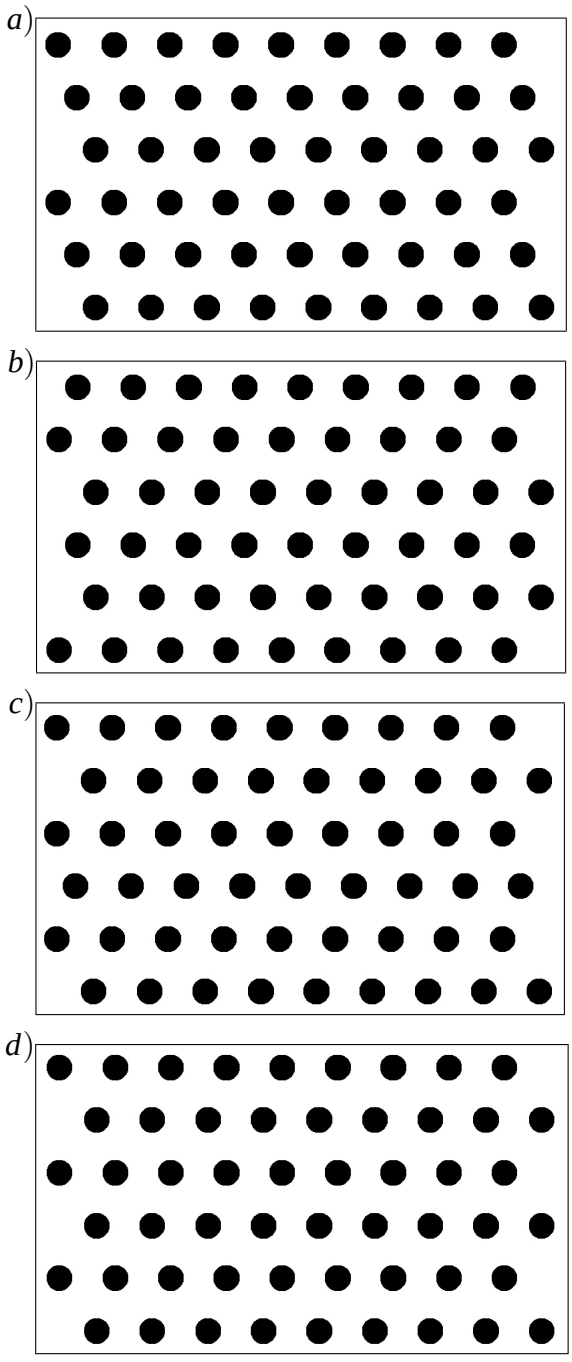

Figure 5.6: The structures outputted from the calculation shown in figure 5.5 looking down the $[0\bar{1}1]$ direction. They are a) pure FCC from generation 48, b) a stacking fault structure with lower enthalpy than FCC from generation 51, c) another stacking fault structure with a lower enthalpy than b) from generation 54, d) pure HCP from generation 127 (the frames are guides to the eye and do not represent the super–cell used in the calculation).

Chapter 6

Ab Initio Carbon Polymorphs

6.1 Introduction

In the previous two chapters I described the basic framework of the GA, but these studies were on systems with a fixed cell. To be a truly *ab initio* method this method should also be able to work with systems where this is not a restriction. In this chapter I will show how to perform crossover in a variable–cell case, and then apply this method to the search for carbon polymorphs using Density Functional Theory. I will compare this technique with the scatter approach taken by Pickard and Needs [2006].

6.2 Variable Cell calculations

As was shown in section 2.2.4 local minimisation is capable of relaxing the cell along with the atomic co–ordinates. To take advantage of this it is necessary that the GA can also cope with having the cell size and shape change during the course of the calculation. In fact this is an almost trivial change due to the way that the GA was formulated in chapter 4. As was shown in figure 4.2 the cuts are made in fractional co-ordinates. This means that the cut itself has no knowledge of the shape of the cell, the cut itself working in an orthogonal space of fractional co–ordinates. After crossover, to interpret what these co–ordinates mean, a set of lattice vectors must be imposed onto the cell.

Figure 6.1 shows how having different cells can still allow parents to have offspring. When converted into fractional co-ordinates the red parent and the blue parent exist in the same space, and therefore it is trivial to breed these two structures.

After crossover a new cell must therefore be determined for the offspring. In

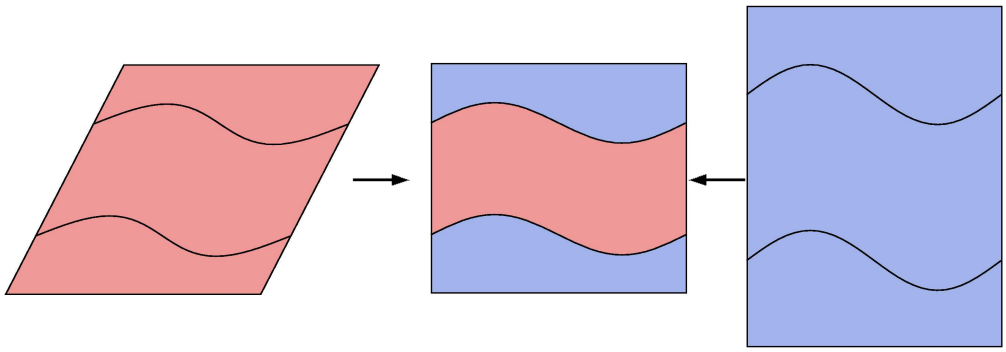

Figure 6.1: Diagram showing how crossover in fractional co–ordinates allows two parents with different cells to be compatible parents.

this study I have elected to randomly choose one cell from each parent to be assigned to one of the two offspring generated in the crossover step. The cell from the other parent is then assigned to the other offspring. Another scheme for defining the cell of the offspring could be to average the two cells of the parents and then give this to both offspring.

In the mutation step that follows the cell is also mutated with the same probability as an atomic mutation (although the amplitude of this mutation can be different to that of an atomic mutation). After this step the whole structure is locally minimised, along with the cell.

In the initialisation phase of the GA calculation, when the random population members are created, it is also necessary to highly randomise the cell of each of these members as well. After this the structures are minimised as usual before being selected for crossover.

6.2.1 Extension with regards to Density Functional Theory

There are a number of considerations that need to be made when performing genetic algorithm studies in an *ab initio* framework, and specifically in the plane–wave DFT framework of CASTEP [Segall *et al.*, 2002]. By incorporating the GA method within CASTEP, rather than using a scripting wrapper around CASTEP, or another *ab initio* code as other methods do [Oganov *et al.*, 2006; Oganov and Glass, 2006], this allows these issues to be addressed more efficiently as the calculation is progressing.

6.2.1.1 *How Many* Plane Waves!

As was discussed in section 3.3.5 the accuracy of a calculation is in part defined by the number of planes waves used in the basis set of the calculation. The number of plane waves is defined by a single parameter, E_{cut}, and as was described in section 3.3.5.1 in a variable cell calculation the number of plane–waves must be allowed to vary to maintain an effective E_{cut} and so maintain the accuracy of the calculation.

Within a GA framework this is doubly important since the enthalpies of different population members will be used the calculate the relative fitnesses of each of the population members, and these fitnesses will be directly compared. If a fixed number of plane–waves is used in this case there is no guarantee that the fitness value of each member, or the relative order of the members determined after the fitnesses have been calculated, is the correct one. It is vital that these are correct for the selection for crossover or update procedures to function correctly.

Prior to a GA calculation the value of E_{cut} will need to have been determined in a fashion similar to that described in section 3.3.6 with some cell with the same atomic species to that being investigated with the GA. After this value has been found it is simply a matter of ensuring that the number of plane–waves is allowed to vary during the calculation.

6.2.1.2 Consistent Number of k–points

In a similar fashion of plane–waves, the number of **k**–points used a calculation must be consistent so that when the fitnesses are determined they are consistent. It is necessary therefore when performing a variable–cell GA calculation that instead of using a symmetry reduced grid of **k**–points [Monkhorst and Pack, 1976; Pack and Monkhorst, 1977] a full, dense, mesh of points is required. The necessary density of **k**–points is determined in the same way to that shown in section 3.3.6. It is this spacing that is used during a calculation.

However, the cell may change in shape and volume drastically during a local minimisation step. If this happens it is necessary to re–perform the local minimisation *after* re–calculating the required number of **k**–points to maintain this sampling. This re–minimising will need to be done to convergence in number of **k**–points, and at this stage the energies of the different members in the population can now be compared.

Since this GA method has been incorporated directly into the CASTEP code this allows for some time–saving measures. Usually when calculating the

initial wave–functions in the first step of the minimisation a random set are chosen and then minimised using a conjugate–gradient minimiser (see section 2.2.3). In this case since the structure is already well minimised (just at the incorrect **k**–point density) using the final wave–functions from the previous minimisation is a better first guess than a random one, and so may save some time in calculation.

With the number of plane–waves allowed to vary, and the **k**–point density kept constant, Genetic Algorithms and Density Functional Theory can be used in conjunction to study systems truly "from first principles". In this chapter I will apply these methods to the study of carbon polymorphs. GAs are a good technique for the stochastic study of polymorphs since these will appear as a by–product of a GA calculation [Woodley, 2004a].

6.3 Allotropes of Carbon

There are number of different carbon allotropes that are seen regularly, and these have many different properties, with many different applications. While this system has been well studied, it is a good test case for the GA method to see how well it performs in searching for different polymorphs.

6.3.1 Diamond

6.3.1.1 Cubic Diamond

The structure of cubic diamond was first solved by Bragg and Bragg [1913a,b] and was the first element whose structure was solved by the use of X–ray diffraction techniques [Donohue, 1974]. Its space group is $Fd\bar{3}m$ (227) and it has sp^3–bonding. It is this strong sp^3–bonding that makes it the hardest known substance. The structure of diamond can be seen in figure 6.2

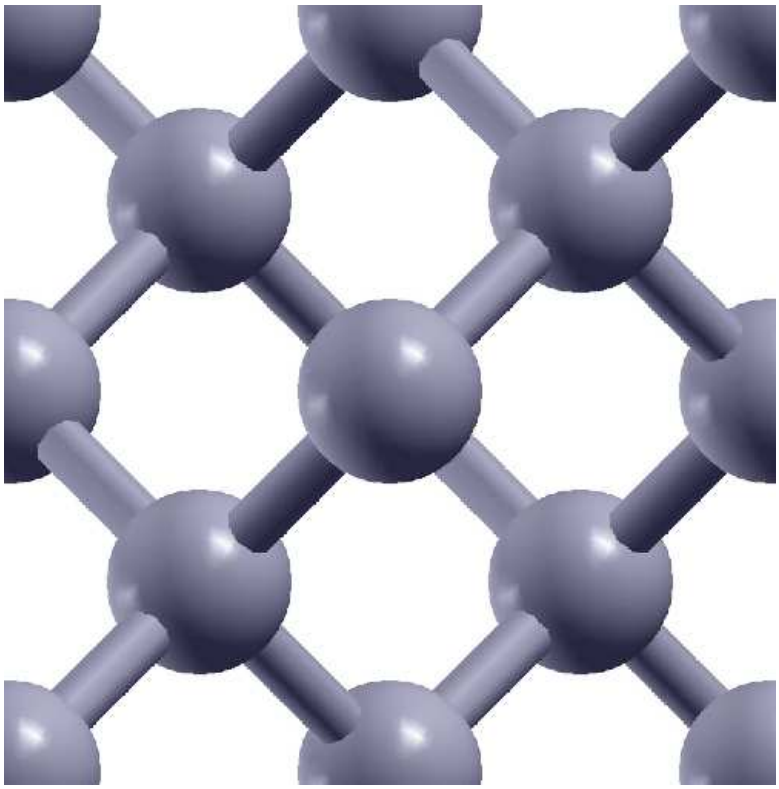

Figure 6.2: Diamond.

6.3.1.2 Hexagonal–Diamond: Lonsdaleite

This structure was first suggested by Ergun and Alexander [1962] as a comparison between the cubic zinc–blende structure and the hexagonal wurtzite structure [Donohue, 1974]. Its space group is $P6_3/mmc$ (194) and it has sp^3–bonding. It is named in honour of Kathleen Lonsdale. Its structure can be seen in figure 6.3

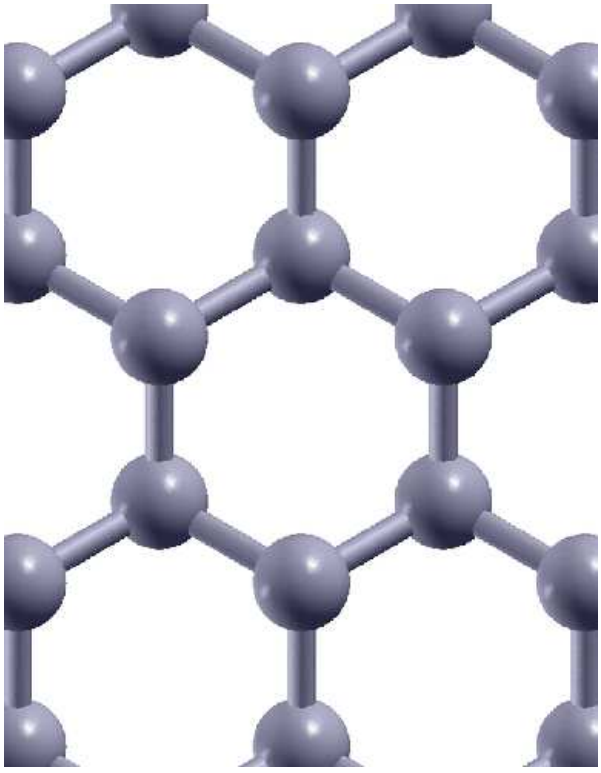

Figure 6.3: Lonsdaleite.

6.3.2 Graphite

Van der Waals forces are responsible for the inter–sheet bonding in graphite, and when performing DFT calculations on graphite–like structures, any local or semi–local method of approximating the exchange–correlation potential V_{xc} will not include these effects. It is possible to use the Local Density Approximation, as opposed to any Generalised Gradient Approximations as these are known to under–bind weakly interacting systems such as graphite sheets. While the LDA does not accurately represent graphite, it does allow for the sheets to be bound. In a GGA calculation the graphene sheets will not be bound and they will separate during a geometry optimisation.

6.3.2.1 Hexagonal Graphite

Graphite has been used for a number of different purposes since the 1780's. It is, unlike diamond, a good electrical conductor and is also used a lubricant. The hexagonal form has a space group is $P6_3/mmc$ (194) . Its bonding is sp^2–hybridised. This is the most common form of graphite, and its structure can be seen in figure 6.4.

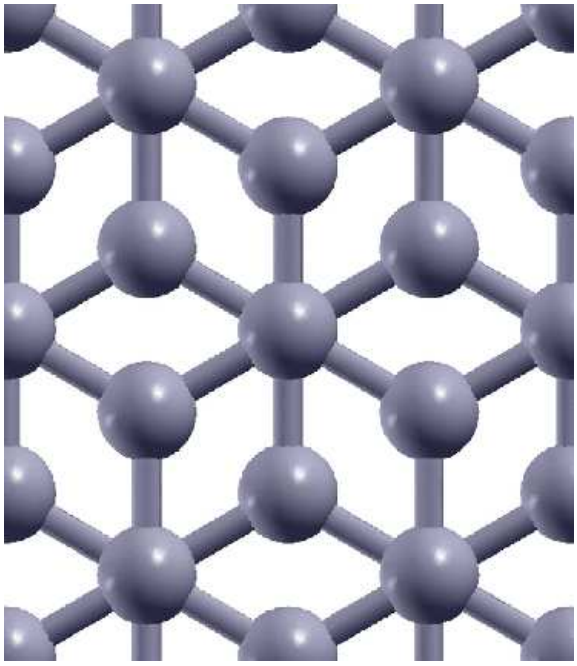

Figure 6.4: Graphite.

6.3.2.2 Buckled Graphite

It was originally thought that graphite was buckled, since some measurements seemed to indicate this [Donohue, 1974]. The buckled form of graphite has a space group of $P6_3mc$ (186) and it also has sp^2–hybridised bonding. A view showing the buckling of the layers can be seen in figure 6.5.

It should also be noted that in the calculations presented below, the buckled graphite phase is not stable and will minimise to hexagonal graphite.

6.3.2.3 Rhombohedral Graphite

This is another sp^2–hybridised structure, similar to hexagonal graphite, but in this case the stacking is as shown in figure 6.6. The space group of this structure is $R\bar{3}m$ (166) .

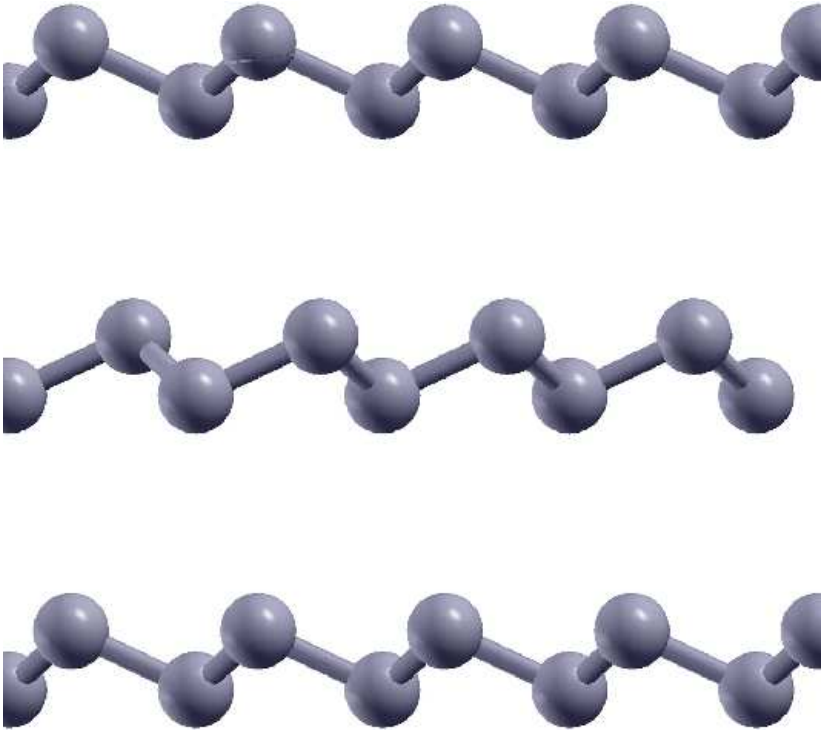

Figure 6.5: Buckled Graphite.

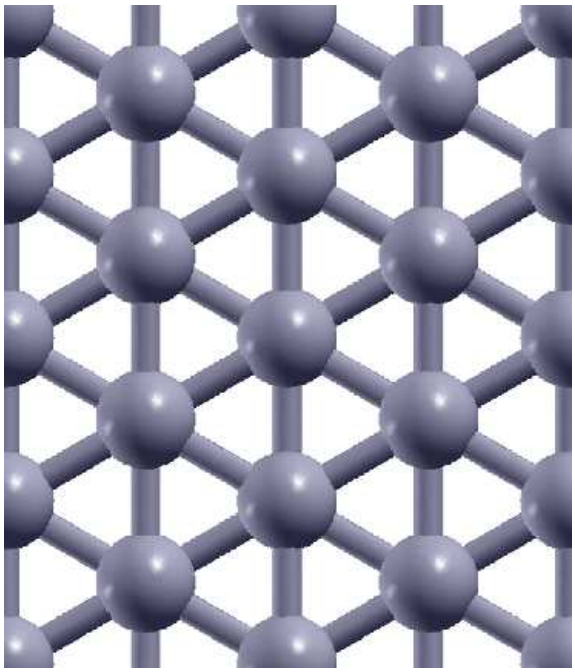

Figure 6.6: Rhombohedral Graphite.

6.3.3 Other Forms of Carbon

6.3.3.1 sp^2– and sp^3–hybridised structures from graph theory

There have been some recent systematic studies on both sp^2– and sp^3–hybridised structures using graph theory. A study of 4 and 6 atom carbon polymorphs has been performed using this method [Winkler *et al.*, 1999, 2001], and showed a range of structures and symmetry groups. Due to the nature of the method only lattice structures could be predicted, so graphite was unable to be found using this method. The sp^3–hybridised study of Strong *et al.* [2004] focused only on 4 atoms per unit cell, and found 8 other sp^3structures as well as diamond and Lonsdaleite. A summary of these structures can be found in table 6.1.

Structure	Enthalpy (eV/Atom)	Symmetry (Space Group)
Diamond	-155.559	$Fd\bar{3}m$ (227)
Graphite	-155.556	$P6_3/mmc$ (194)
Rhombohedral Graphite	-155.555	R3m (166)
Lonsdaleite	-155.534	$P6_3/mmc$ (194)
D *	+0.198	I4/mmm (139)
E *	+0.364	Imma (74)
F *	+0.803	C2 (5)
G *	+1.101	$P4_122$ (91)
H *	+1.138	Fddd (70)
I *	+1.328	C2 (5)
J *	+1.579	$I2_12_12_1$ (24)
K *	+1.677	$I2_12_12_1$ (24)
$4(3)1$ **	+1.166	$I4_132$ (214)
$4(3)2^d$ **	+0.448	$I4_1/amd$ (141)

* These structures are taken from Strong *et al.* [2004]. These calculations used a GGA, with a plane–wave basis cut–off of 550 eV and a Monkhorst–Pack sampling of reciprocal space corresponding to $0.05\,\text{Å}^{-1}$.

** These structures are taken from Winkler *et al.* [2001]. These calculations used a GGA and a sampling of reciprocal space corresponding to $0.04\,\text{Å}^{-1}$.

Table 6.1: Summary of 4–atom carbon polymorphs; diamond, Lonsdaleite and graphite, and structures from Strong *et al.* [2004] and Winkler *et al.* [2001]. The data taken from the source material gives the enthalpies relative to diamond, however, due to the computational differences between this study and these published results, the equivalent values may be slightly different.

6.3.3.2 Carbon Structures with 3– and 4–Member Rings

Schultz *et al*. [1999] showed that it is possible to form sp^3–hybridised carbon structures which contain 3– and 4–member rings, as opposed to the 6–member rings found in diamond, Lonsdaleite and graphite. The structure with 4–member rings is in fact structure D from Strong *et al*., and is shown in figure 6.7. Its space group is I4/mmm (139) .

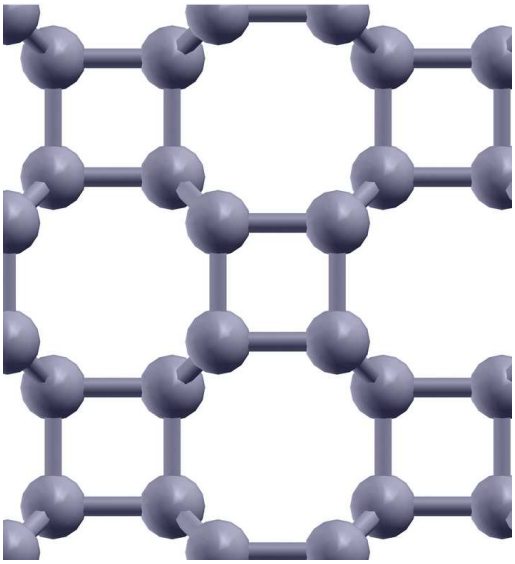

Figure 6.7: 4–member ring structure.

The 3–member ring structure is shown in figure 6.8, and has space group $P6_3/mmc$ (194) . While the 4–member ring structure has 4 atoms in its unit cell, this structure has 6 atoms, and so will not be accessible to this study.

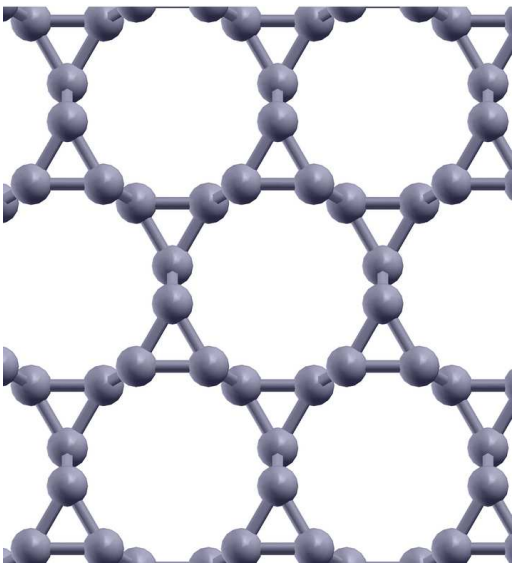

Figure 6.8: 3–member ring structure.

6.3.3.3 Restrictions

There is a large amount of literature on different possible forms of carbon, including Bucky–balls and carbon nanotubes. While these are interesting structures that warrant investigation, they do not display the three–dimensional periodicity that this GA method was designed for. Their large unit cells would also mean that any DFT calculations performed on these structures would be expensive. I will focus this study on bulk carbon, with 4 atoms per unit cell. The data shown in table 6.1 means that it will be possible to test the effectiveness of the GA at finding structures other than diamond or graphite, although the GA method has advantages over the graph–theory approaches since it does not assume the type of bonding in a structure and so will allow both sp^2– and sp^3–bonded structures to occur in the population concurrently. It is also able to have sheeted structures such as graphite.

Restricting the search to 4–atom polymorphs means that the time taken for calculations should be reasonable, even in an *ab initio* level of theory. A by–product of this constraint is that some structures, like that of 8–atom glitter [Bucknum *et al.*, 2005] are inaccessible. There is also a proposed sp^2–hybridised metallic polymorph with 8 atoms per unit cell and a space group of $I4_1/amd$ (141) , which should also be inaccessible [Hoffmann *et al.*, 1983]. Bucknum *et al.* [2005] also references large number of other proposed carbon polymorphs, which I shall not discuss here.

6.4 Results

In these results I will present results from GA calculations with 8 population members, as opposed to 16 in previous studies, but rather than just being interested in the breeding population, it is necessary to consider all structures found during the course of a calculation. Rather than starting using coarser parameters it was also realised that running with fine tolerances from the start of a calculation was necessary. While this may slow down the initial gathering of results, in the end it gives better data. It is also important that all population members have fully converged electron density and minimised atomic structure in an *ab initio* case.

If an un–converged structure is allowed in the breeding population it increases the likelihood that offspring from this member will also fail to converge. If this happens it may end up with all population members being un–converged, and in this case the results from a calculation may be meaningless. As was shown in the convergence graphs in figure 3.5, a plane–wave cut–off energy of 400 eV

and a **k**–point sampling of 0.05Å^{-1} were chosen for these calculations.

When using the GA as a polymorph search technique, it is not necessary to keep the fittest structure at the end of each generation, so for these calculations a pure roulette–wheel update procedure was adopted. It is also only necessary to consider the offspring produced, rather than the population members themselves (i.e. the members determined after the update procedure which are allowed to perform crossover). Since all structures produced are fully converged and relaxed, they are all viable polymorphs, as has been discussed in other GA studies [Woodley, 2004a].

All symmetries stated were found using Materials Studio, a Windows based package by Accelrys [2001–] that includes a commercial version of CASTEP. It has a very useful symmetry finder, and the "other possible symmetries" column in the tables presented are symmetries suggested using different levels of tolerance in position. The symmetry given in the "Symmetry (Space Group)" column is that found using a fine (0.01 Å) tolerance, the symmetries listed in the "Other Possible Symmetries" column are those found by reducing the tolerance. While this has not managed to replicate all the structures found in the study of Strong *et al*. [2004] it was able to find a number of different sp^2– and sp^3–bonded structures concurrently, and also found sheeted as well as lattice structures. While the structures shown in section 6.3.3.2 were not found using the GA, there were a large number of structures that displayed similar character by having 3– and 4–member rings.

For the GA calculations, the program ran for a total of 6 generations with eight members in the initial generation, giving 56 possible structures (8 from the initial population, and 8 offspring produced from each of the 6 subsequent generations). These calculations are compared with a random scatter search [Pickard and Needs, 2006], in which 56 independent calculations are performed. This is a relatively short calculation, when compared with the 200 generations used in the Lennard–Jones studies on 130+ atom cells of chapters 4 and 5. However, the number of atoms in this study is very much smaller, and the number of minima in the potential energy surface will grow exponentially with the number of atoms, meaning that this relatively short run should still give meaningful results. The number of structures produced is also comparable with the number of structures produced by Pickard and Needs in their study on silane in which they generated 40 structures at each pressure, with two SiH_4 units per simulation cell.

6.4.1 Polymorph Search

The results presented in this first GA calculation did not make use of the updated fitness function described in chapter 5. The fitness function only used the hyperbolic tangential weighting of equation 2.11c. One offspring, created in the first generation did not converge, but was not selected in the update procedure and so did not sire any offspring.

A histogram showing the enthalpy ranges of the structures found is shown in figure 6.9. The structures from each generation are also colour–coded to allow the progression of the distribution to be seen. A more detailed breakdown of enthalpies and symmetries is shown in tables 6.2 and 6.3.

By the end of the first generation, a large spread of enthalpies had been found, although the GA was still finding a large range of enthalpies throughout all generations. There were also more higher–symmetry structures found in later generations, and the accuracy of the symmetries found also improved as the generations progressed.

The GA found diamond twice (although the structure found in the zeroth generation has a high enthalpy), as well as rhombohedral graphite and other graphite–like structures. Structures E and F from Strong *et al*. [2004] were also found, in fact, structure E was found twelve times, showing that it may be from a large basin in the potential energy surface. This is one reason why the improved fitness function was developed, to prevent a large amount of similar structures being produced. Neither Lonsdaleite or any other structures from Strong *et al*. [2004] or Winkler *et al*. [2001] were found in this short run.

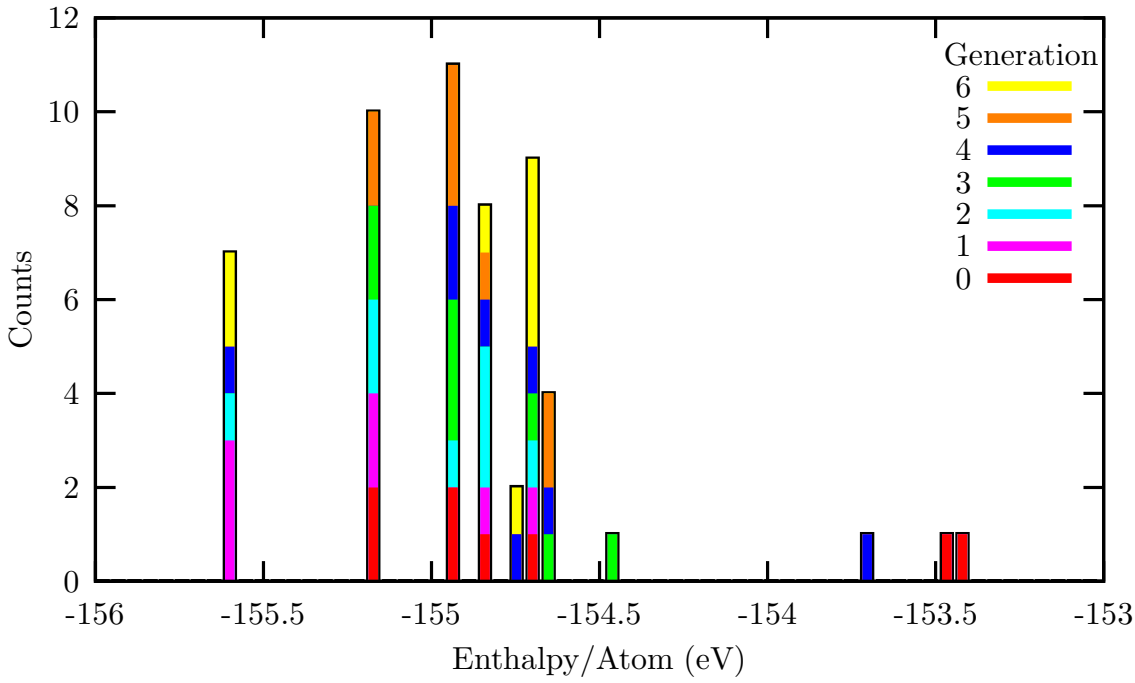

Figure 6.9: Histogram showing the enthalpy distribution found when using the GA with a fitness weight of 0.00 broken down by generation. Only 7 members of generation 1 are shown; member 4 is an un–converged structure with a high enthalpy/atom.

Gen	Member	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
0	1 [S]	-154.896	P$\bar{1}$ (2)	C2/m (12) ; Cmmm (65)
	2 [e]	-155.145	Imma (74)	
	3 [S,3]	-154.810	P$\bar{1}$ (1)	Fmm2 (42)
	4 [S]	-154.697	P$\bar{1}$ (2)	C2/m (12)
	5 [L,3]	-154.901	Cmmm (65)	
	6 [e]	-155.146	Imma (74)	
	7 [N]	-153.403	P$\bar{1}$ (2)	Pmma (51)
	8 [a]	-153.427	Fd$\bar{3}$m (227)	
1	1 [R]	-155.556	P$\bar{1}$ (2)	R$\bar{3}$m (166)
	2 [e]	-155.145	Imma (74)	
	3 [G]	-155.555	P$\bar{1}$ (2)	$P2_1/m$ (11) ; Cmcm (63) ; $P6_3/mmc$ (194)
	4 [†]	-145.753	P$\bar{1}$ (1)	
	5 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
	6 [R]	-155.556	P$\bar{1}$ (2)	R$\bar{3}$m (166)
	7 [S,3]	-154.810	P1 (1)	Cm (8) ; Fmm2 (42)
	8 [e]	-155.145	Cm (8)	Fmm2 (42)
2	1 [R/G]	-155.556	C2/m (12)	R$\bar{3}$m (166)
	2 [e]	-155.146	Imma (74)	
	3 [L,4]	-154.901	Cmmm (65)	
	4 [S,3]	-154.810	P1 (1)	Fmm2 (42)
	5 [S,3]	-154.810	P$\bar{1}$ (1)	C2 (5) ; Fmm2 (42)
	6 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
	7 [e]	-155.145	Imma (74)	
	8 [L,4]	-154.810	Cmmm (65)	
3	1 [S,4]	-154.896	C2/m (12)	Cmmm (65)
	2 [L,3]	-154.637	C2 (5)	
	3 [e]	-155.145	Imma (74)	
	4 [S,4]	-154.896	P$\bar{1}$ (2)	Cmmm (65)

[†] This structure is un–converged. It was not selected to be included in the next generation and so had no offspring.
[a] Diamond.
[e] Structure E from Strong *et al*. [2004].
[f] Structure F from Strong *et al*. [2004].
[G] Graphite–like structure.
[R] Rhombohedral graphite–like structure.
[L] Lattice structure.
[S] Sheeted structure.
[N] Neither a lattice or sheeted structure.
[3] Structure with 3–member rings.
[4] Structure with 4–member rings.

Table 6.2: Summary of 6–generation polymorph search showing generations 0–2 and the first half of generation 3.

Gen	Member	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
3	5 [s]	-154.696	$P\bar{1}$ (2)	C2/m (12)
	6 [S,4]	-154.896	$P\bar{1}$ (2)	
	7 [e]	-155.145	Imma (74)	Cmmm (65)
	8 [e]	-154.433	Imma (74)	
4	1 [S,4]	-154.897	P2/m (10)	Cmmm (65)
	2 [L]	-154.833	Fddd (70)	$I4_1/amd$ (141)
	3 [L,3]	-154.657	Imm2 (44)	
	4 [f]	-154.705	C2 (5)	
	5 [L,3]	-154.637	C2 (5)	
	6 [G]	-155.555	Cmcm (63)	
	7 [S,4]	-154.896	$P\bar{1}$ (2)	Cmmm (65) ; $P6_3/mmc$ (194)
	8 [e]	-153.674	Imma (74)	
5	1 [e]	-155.145	Imma (74)	
	2 [L,3]	-154.637	C2 (5)	
	3 [e]	-155.145	Imma (74)	
	4 [S,4]	-154.897	$C2/\bar{m}$ (12)	Cmmm (65)
	5 [S,4]	-154.896	$P\bar{1}$ (2)	Cmmm (65)
	6 [L,3]	-154.637	Cmcm (63)	R32 (155)
	7 [S,4]	-154.896	C2/m (12)	Cmmm (65)
	8 [S,4]	-154.806	$C2/\bar{m}$ (12)	Cmmm (65)
6	1 [a]	-155.559	$Fd\bar{3}m$ (227)	
	2 [L]	-154.654	$I\bar{4}m2$ (119)	
	3 [L,3]	-154.657	Imm2 (44)	
	4 [S,3]	-154.806	P1 (1)	Cm (8) ; R3m (160)
	5 [f]	-154.705	C2 (5)	
	6 [r]	-155.556	C2/m (12)	$R\bar{3}m$ (166)
	7 [L,3]	-154.657	Imm2 (44)	
	8 [L,3]	-154.696	Imm2 (44)	

[a] Diamond.
[r] Rhombohedral graphite.
[e] Structure E from Strong *et al*. [2004].
[f] Structure F from Strong *et al*. [2004].
[L] Lattice structure.
[S] Sheeted structure.
[3] Structure with 3–member rings.
[4] Structure with 4–member rings.

Table 6.3: Summary of 6–generation polymorph search showing the second half of generation 3 and generations 4–6.

6.4.2 Improvements: The use of the Structure Factor

In the previous chapter I showed that using the structure factor improved convergence on to the global minimum structure. This method should not just do this however, it should also enable the differentiation of different structures during the course of a calculation and therefore improve structural diversity in the population. It should also try to prevent a large amount of any one structure being produced as offspring by reducing the number of each type of structure in the population. Again the GA was run for 6 generations giving 56 possible structures, but in this case the fitness weight parameter, w, was set to 0.75.

A histogram showing the enthalpy distribution is shown in figure 6.10, and a comparison of this graph and that produced from the previous section is shown in figure 6.11. As can be clearly seen, the use of the structure factor has led to the distribution of enthalpies being skewed towards lower enthalpy structures to a greater extent than in the previous calculation.

The results presented in tables 6.4 and 6.5 give more detailed symmetry information. As well as finding diamond, rhombohedral and hexagonal graphite, and structures E and F from Strong *et al*. [2004], Lonsdaleite was also found. The Lonsdaleite structures are actually the smaller peak to the right of the lowest enthalpy peak in figure 6.10, as well as one higher enthalpy graphite–like structure. Fewer instances of structure E were also found, only 7 in this case.

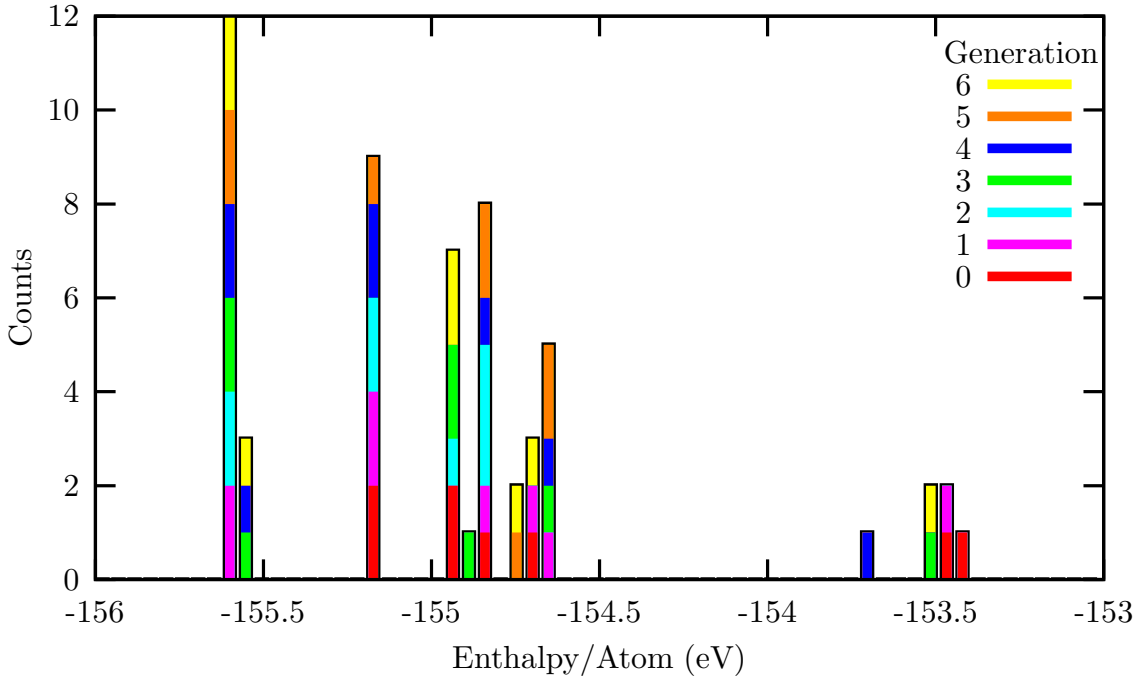

Figure 6.10: Diagram showing the enthalpy distribution found when using the GA with a fitness weight of 0.75 broken down by generation.

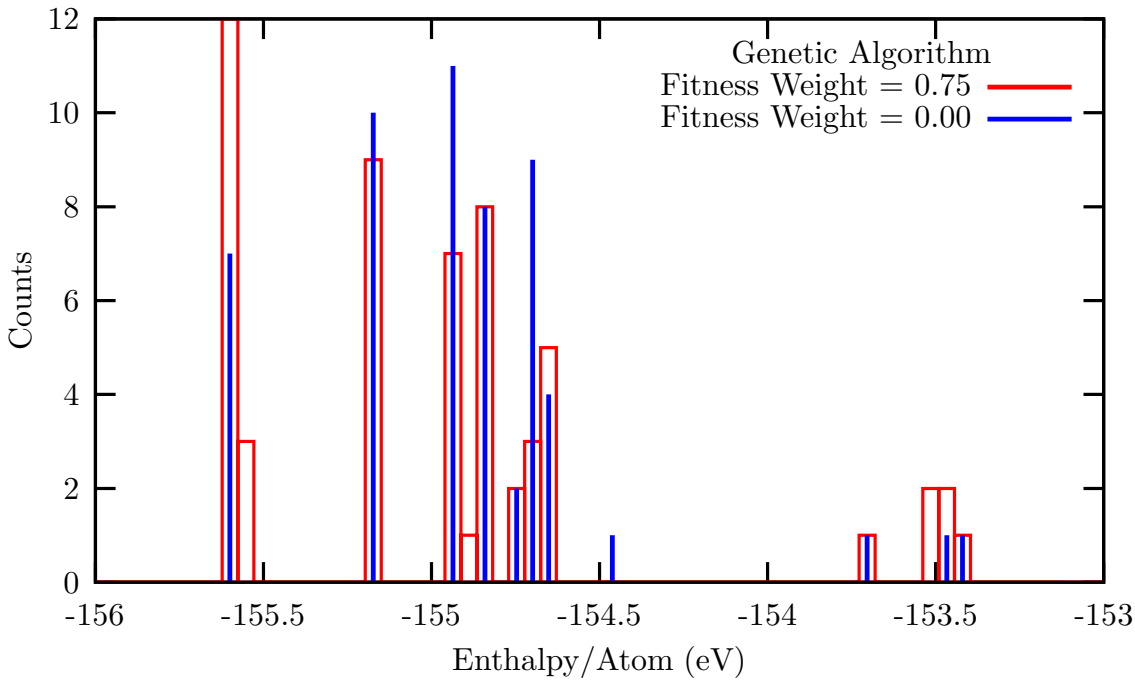

Figure 6.11: Comparison of the enthalpy distribution found when using the GA with a fitness weight of 0.00 or 0.75. The extra low–enthalpy peak at $-155.53\,\mathrm{eV}$ is due to two instances of Lonsdaleite and one of a graphite–like structure.

Gen	Member	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
0	1 [S,4]	-154.896	$P\bar{1}$ (2)	C2/m (12) ; Cmmm (65)
	2 [e]	-155.145	Imma (74)	
	3 [S,3]	-154.810	Cm (8)	
	4 [S]	-154.697	$P\bar{1}$ (2)	C2/m (12)
	5 [L]	-154.901	Cmmm (65)	
	6 [e]	-155.146	Imma (74)	
	7 [N]	-153.403	$P\bar{1}$ (2)	Pmma (51)
	8 [L,3]	-153.427	C2 (5)	
1	1 [N]	-153.451	P1 (1)	Cm (8)
	2 [e]	-155.145	Imma (74)	
	3 [G]	-155.555	$P\bar{1}$ (2)	C2/m (12)
	4 [S]	-155.555	$P\bar{1}$ (2)	C2/c (15) ; $R\bar{3}$m (166)
	5 [L]	-154.657	Imm2 (44)	
	6 [L,3]	-154.637	C2 (5)	
	7 [S,3]	-154.811	P1 (1)	
	8 [L]	-155.145	Imm2 (44)	
2	1 [G]	-155.554	$P\bar{1}$ (2)	C2/c (15)
	2 [L]	-154.901	Cmmm (65)	C2/m (12)
	3 [S,3]	-154.806	P1 (1)	Cm (8) ; $R\bar{3}$m (166)
	4 [S,3]	-154.811	P1 (1)	Fmm2 (42)
	5 [e]	-155.145	Imma (74)	
	6 [S,3]	-154.807	Cm (8)	
	7 [G]	-155.552	$P\bar{1}$ (2)	$P2_1$/m (11) ; Cmcm (63)
	8 [e]	-155.145	Imma (74)	
3	1 [L]	-154.901	C2/m (12)	Cmmm (65)
	2 [N]	-153.471	Cm (8)	Amm2 (38)
	3 [G]	-155.544	C2/m (12)	
	4 [‡,L]	-154.852	C2/c (15)	Fddd (70)

[‡] Possibly the sp^2–bonded structure from Hoffmann *et al*. [1983]
[e] Structure E from Strong *et al*. [2004].
[L] Lattice structure.
[S] Sheeted structure.
[N] Neither a lattice or sheeted structure.
[G] Graphite–like structure.
[3] Structure with 3–member rings.
[4] Structure with 4–member rings.

Table 6.4: Summary of 6–generation polymorph search with $w = 0.75$ showing generations 0–2 and the first half of generation 3.

Gen	Member	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
3	5 [S,4]	-154.896	Cmmm (65)	
	6 [G]	-155.556	$P\bar{1}$ (2)	$R\bar{3}m$ (166)
	7 [L,3]	-154.637	C2 (5)	
	8 [S,4]	-155.556	Cmmm (65)	
4	1 [e]	-155.146	Imma (74)	
	2 [G]	-155.556	$P\bar{1}$ (2)	Cmcm (63) ; $P6_3/mmc$ (194)
	3 [N]	-153.667	P1 (1)	C2/m (12)
	4 [R]	-155.556	$P\bar{1}$ (2)	C2/c (15) ; $R\bar{3}m$ (166)
	5 [c]	-155.534	$P6_3/mmc$ (194)	
	6 [e]	-155.145	Imma (74)	
	7 [L,3]	-154.637	C2 (5)	
	8 [L]	-154.807	Imm2 (44)	
5	1 [L,3]	-154.637	C2 (5)	
	2 [S,3]	-154.806	P1 (1)	Cm (8)
	3 [L,3]	-154.637	C2 (5)	
	4 [S,3]	-154.806	P1 (1)	Cm (8) ; R3m (130)
	5 [f]	-154.705	C2 (5)	
	6 [R]	-155.555	$P\bar{1}$ (2)	$R\bar{3}m$ (166)
	7 [r]	-155.555	$R\bar{3}m$ (166)	
	8 [L]	-155.145	Imm2 (44)	
6	1 [f]	-154.704	C2 (5)	
	2 [c]	-155.534	$P6_3/mmc$ (194)	
	3 [S,4]	-154.896	C2/m (12)	Cmmm (65)
	4 [N]	-153.470	Cm (8)	Amm2 (38)
	5 [S,4]	-154.896	$P\bar{1}$ (2)	C2/m (12) ; Cmmm (65)
	6 [S]	-154.696	C2/m (12)	
	7 [a]	-155.559	$Fd\bar{3}m$ (227)	
	8 [L,3]	-155.554	C2 (5)	

[a] Diamond.
[r] Rhombohedral graphite.
[c] Lonsdaleite.
[e] Structure E from Strong *et al*. [2004].
[f] Structure F from Strong *et al*. [2004].
[G] Graphite–like structure.
[R] Rhombohedral graphite–like structure.
[L] Lattice structure.
[S] Sheeted structure.
[N] Neither a lattice or sheeted structure.
[3] Structure with 3–member rings.
[4] Structure with 4–member rings.

Table 6.5: Summary of 6–generation polymorph search with $w = 0.75$ showing the second half of generation 3 and generations 4–6.

Structure 4 from generation 3 is also rather interesting. It appears to be very similar to the sp^2–bonded structure proposed in Hoffmann *et al*. [1983], and is shown in figure 6.12. The bond lengths and angles of this structure are given in table 6.6, and are close to the suggested values of 120° and 1.44 Å. The structure proposed by Hoffmann *et al*. has space group $I4_1/amd$ (141) , as well as having 8 atoms in its unit cell. This structure has a space group of C2/c (15) (or Fddd (70) with lower tolerances) however. When the symmetry of space group Fddd (70) is enforced the bond angles become 125.893° and 117.054° with bond lengths of 1.433 Å and 1.465 Å.

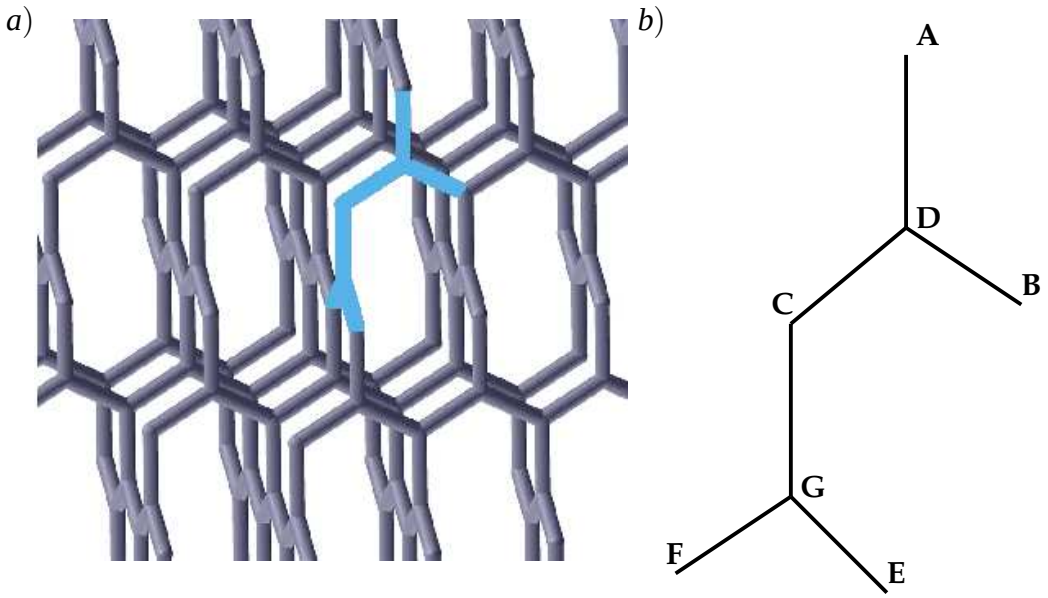

Figure 6.12: *a*) Structure 4 from generation 3. The highlighted section is described in *b*) The labelling convention for table 6.6.

Angles		Bond Lengths	
∠ADB	125.90°	AD	1.433 Å
∠ADC	117.17°	BD	1.465 Å
∠BDC	116.93°	CD	1.433 Å
∠CGE	116.84°	CG	1.465 Å
∠CGF	125.88°	EG	1.433 Å
∠EGF	117.28°	FG	1.432 Å
∠DCG	116.94°		

Table 6.6: Table giving bond lengths and angles for the structure shown in figure 6.12 *a*). The naming convention is as shown in figure 6.12 *b*).

6.4.3 Comparison: Random Scatter

In order to compare the methods above with another of similar capabilities, I will test against the method of Pickard and Needs [2006] (who tested their method on 40 structures for silane and 100 structures for 2–atom silicon) by performing a random search of 56 structures, which is a comparable number to that chosen for the study on silane. This is equivalent to performing a generation 0 calculation on 56 population members, and then exiting the GA calculation before any breeding has occurred. This will also be a test of the structure factor, since the use of this in the fitness function should force the population into other areas of the potential energy surface not explored in the initial generation.

Histograms comparing the enthalpy distribution of the scatter method and that of the two previous GA runs are shown in figures 6.14 and 6.15, showing the $w = 0.0$ and $w = 0.75$ comparisons respectively. The most obvious fact about these plots is the fact that there are a much larger number of higher enthalpy structures produced by the scatter method than in the GA calculations, and that the spread of enthalpies is much more evenly distributed across the range in the scatter calculations. There are two low enthalpy peaks that do not appear in either of the GA calculations, one is structure D from Strong *et al*. [2004], the other structure is shown in figure 6.13.

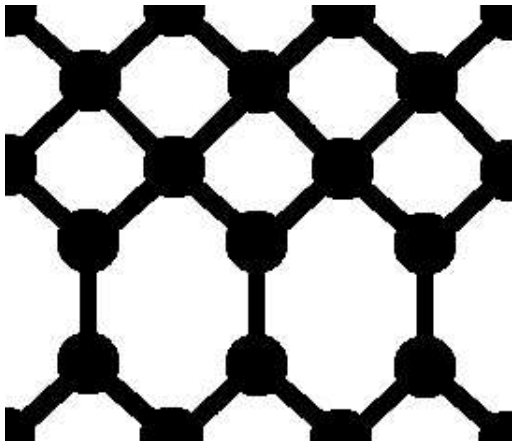

Figure 6.13: Structure 49 found using the random scatter method.

While Lonsdaleite was found, diamond and structure F from Strong *et al*. [2004] were not. Structure E from Strong *et al*. [2004] was found 8 times and there were also a large number of neither sheeted or lattice structures found. Some of these were an array of carbon atoms arranged in linear chains and structures similar to these have been reported as a polymorph of carbon known as carbyne [Lagow *et al*., 1995; Scemama *et al*., 2002].

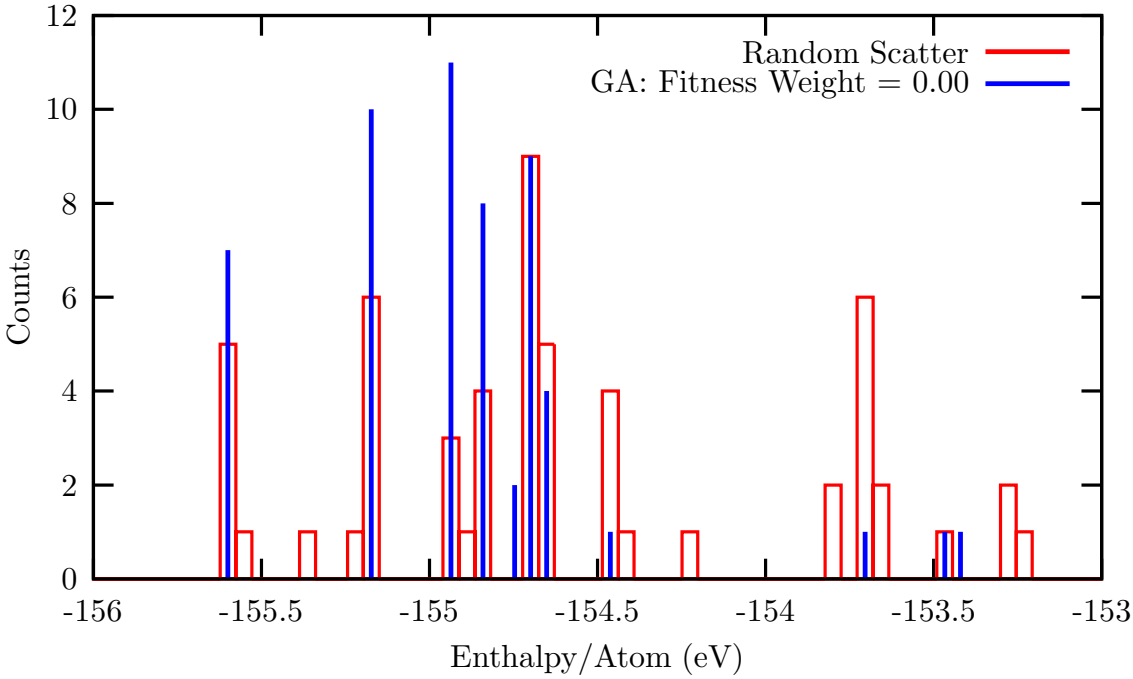

Figure 6.14: Comparison of the enthalpy distribution found when using a scatter method or the GA with a fitness weight of 0.00.

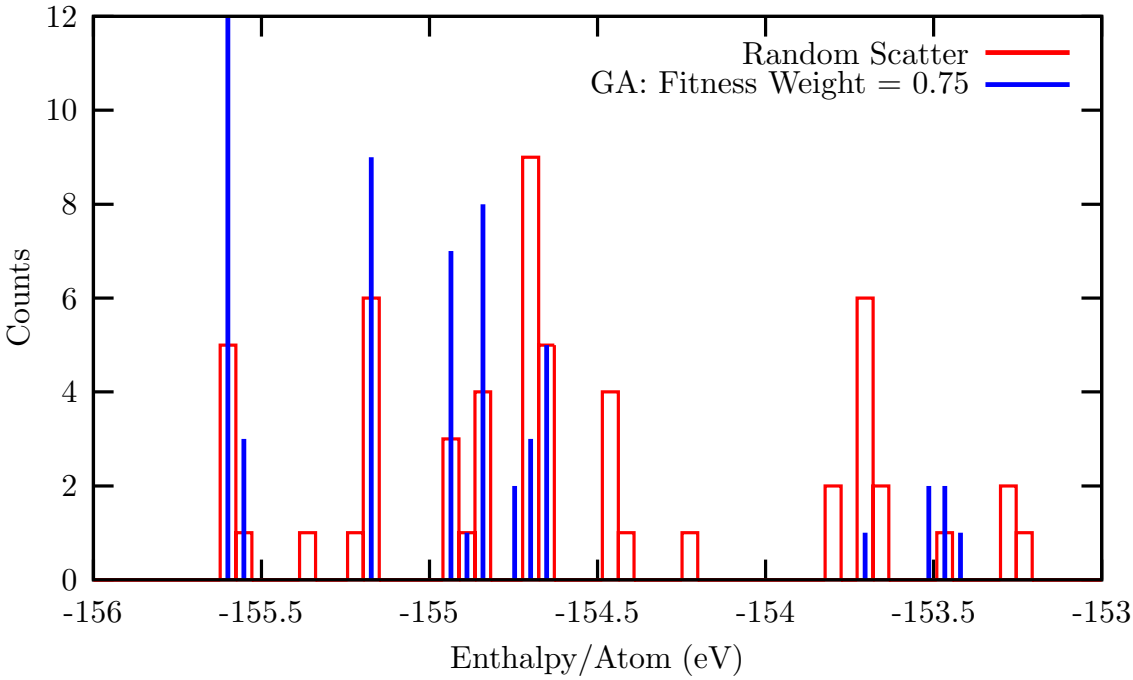

Figure 6.15: Comparison of the enthalpy distribution found when using a scatter method or the GA with a fitness weight of 0.75.

Structure	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
1 [R]	-153.221	P$\bar{1}$ (2)	R$\bar{3}$m (166)
2 [N]	-153.243	Pm2 (6)	
3 [N]	-153.277	P1 (1)	Pm2 (6)
4 [N]	-153.448	P1 (1)	Pm2 (6)
5 [N]	-153.611	P1 (1)	
6 [N]	-153.616	P1 (1)	C2 (5)
7 [N]	-153.665	P$\bar{1}$ (1)	Cm (8)
8 [N]	-153.673	P$\bar{1}$ (2)	P2/m (10)
9 [e]	-153.675	Imma (74)	
10 [N]	-153.675	P$\bar{1}$ (2)	P2/m (10)
11 [N]	-153.675	P$\bar{1}$ (2)	P2/m (10)
12 [N]	-153.677	P$\bar{1}$ (1)	F222 (22)
13 [N]	-153.793	P$\bar{1}$ (2)	Pmma (51)
14 [N]	-153.794	P$\bar{1}$ (2)	C2/m (12) ; Cmcm (63)
15 [l]	-154.205	P$\bar{1}$ (1)	Ama2 (40) ; $P4_2mmc$ (131)
16 [l]	-154.411	P$\bar{1}$ (2)	F$\bar{2}$22 (22)
17 [l]	-154.418	P$\bar{1}$ (2)	R$\bar{3}$m (166)
18 [e]	-154.421	Imma (74)	
19 [l]	-154.426	P$\bar{1}$ (2)	R$\bar{3}$m (166)
20 [S]	-154.428	P$\bar{1}$ (2)	
21 [L,3]	-154.637	C2 (5)	
22 [L]	-154.637	C2 (5)	
23 [L,3]	-154.637	C2 (5)	
24 [R]	-154.637	C2/m (12)	R$\bar{3}$m (166)
25 [L,3]	-154.637	C$\bar{2}$ (5)	
26 [L]	-154.654	I$\bar{4}$m2 (119)	
27 [L]	-154.657	Imm2 (44)	
28 [L,3]	-154.657	Imm2 (44)	

[e] Structure E from Strong *et al*. [2004].
[l] Linear structure.
[R] Rhombohedral graphite–like structure.
[L] Lattice structure.
[S] Sheeted structure.
[N] Neither a lattice or sheeted structure.
[3] Structure with 3–member rings.

Table 6.7: Summary of Random Scatter minimisations showing structures 1–28.

Structure	Enthalpy (eV/Atom)	Symmetry (Space Group)	Other Possible Symmetries
29 [S]	-154.696	P$\bar{1}$ (2)	
30 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
31 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
32 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
33 [S]	-154.696	P$\bar{1}$ (2)	
34 [S]	-154.696	P$\bar{1}$ (2)	C2/m (12)
35 [S,3]	-154.810	P1 (1)	Fmm2 (42)
36 [S,3]	-154.810	P1 (1)	Cm (8) ; Fmm2 (42)
37 [S,3]	-154.810	P1 (1)	Cm (8) ; Fmm2 (42)
38 [S,3]	-154.810	P1 (1)	Cm (8) ; Fmm2 (42)
39 [R]	-154.854	C2/m (12)	R$\bar{3}$m (166)
40 [S,4]	-154.896	Cmmm (65)	
41 [S]	-154.897	P1 (1)	Cmmm (65)
42 [L]	-154.901	Cmmm (65)	
43 [e]	-155.145	Imma (74)	
44 [e]	-155.145	Imma (74)	
45 [e]	-155.145	Imma (74)	
46 [e]	-155.145	Imma (74)	
47 [e]	-155.145	Imma (74)	
48 [e]	-155.146	Imma (74)	
49 [L]	-155.203	Cmmm (65)	
50 [d]	-155.338	I4/mmm (139)	
51 [c]	-155.534	P6$_3$/mmc (194)	
52 [R]	-155.555	C2/m (12)	R$\bar{3}$m (166)
53 [R]	-155.556	C2/c (15)	R$\bar{3}$m (166)
54 [R]	-155.556	P$\bar{1}$ (2)	R$\bar{3}$m (166)
55 [R]	-155.556	P$\bar{1}$ (2)	R$\bar{3}$m (166)
56 [R]	-155.556	P$\bar{1}$ (2)	R$\bar{3}$m (166)

[r] Rhombohedral graphite.
[c] Lonsdaleite.
[d] Structure D from Strong *et al*. [2004].
[e] Structure E from Strong *et al*. [2004].
[R] Rhombohedral graphite–like structure.
[L] Lattice structure.
[S] Sheeted structure.
[N] Neither a lattice or sheeted structure.
[3] Structure with 3–member rings.
[4] Structure with 4–member rings.

Table 6.8: Summary of Random Scatter minimisations showing structures 29–56.

While there were are only a relatively small number of structures produced in this study, it is similar number to that created by Pickard and Needs [2006], and even though only two GA calculations were performed, as compared to the 15 runs produced for the Lennard–Jones calculations, the trend of the enthalpy distribution is still clear and is as would be expected. The structures produced by the scatter method are independent and so would have a much more even enthalpy distribution compared to the GA, which is a method that is designed to search for minimum enthalpy structures.

Which of these methods is the best for this type of study? Looking at the results the scatter method has found some structures that the GA did not find, and the GA found other structures that were not found in the scatter approach.

A much more rigorous comparison of these two methods would be to perform the scatter search until all the structures listed in table 6.1 have been found, and to then perform a number of GA calculations to produce the same number of new structures, varying the number of population members in each generation. When performing these GA calculations it is necessary to consider not only how many structures are produced, but how many are produced in each generation. These calculations presented here were for 8 population members over 6 generations, but the same number of structures would have been produced with 4 population members over 13 generations.

By setting off a range of GA calculations all producing the same number of structures as those required by the scatter approach would allow a full analysis of the role that population size plays in the GA. The results presented here show that as more structures are generated by the scatter approach they will also proportionally generate a larger number of high–enthalpy structures, whereas the GA will focus more on (but not be limited to) the lower–enthalpy structures. As the system size increases this will increase the number of structures required by the scatter approach to unfeasible levels as it samples the potential energy surface

6.5 Conclusions

By allowing the cuts made in the crossover operation to be made in fractional co–ordinates each population member may be allowed to have unit cells which have different sizes and shapes. This means that it is possible to use the GA as a polymorph search technique without having to test all possible symmetries in a systematic search. Since the GA will always start from an initial random structure so there is no initial bias to any preconceived solution.

The initial search only used roulette wheel update, and did not use the improved fitness function described in chapter 5. These results showed a distribution skewed towards lower enthalpy structures (as would be expected from the weighting factor associated with the roulette wheel method). By using the structure factor the number of different structures increased slightly, and the enthalpy distribution was more skewed towards lower enthalpies. The use of the structure factor resulted in fewer instances of the higher enthalpy Imma (74) structure, and more of the lower enthalpy graphite–like structures.

The structure factor augmented fitness function is behaving in the same way as in chapter 5. It will move the system away from high enthalpy structures, until it reaches lower enthalpy structures. At this point, since it cannot drive the system down further these structures will start to become more numerous in the population, but the total number of different structures found during a calculation will increase, since the system is being forced into new symmetry structures. An improvement to this method would be to store the structure factor of all previous structures that have been found, and to compare current structures against these, as well as those of the current population.

When the GA is compared with the scatter approach of Pickard and Needs [2006] it is clear that although the scatter approach samples more of the potential energy surface, it samples it more evenly, and although it found two structures not found in the GA calculations, it also missed some structures found in these studies. In effect, the GA will sample the lower–enthalpy regions of the PES more regularly. However, these calculations were only over 56 possible structures, due to the cost involved in performing *ab initio* GA calculations.

As the system size is increased by adding more atoms then the number of total minima in the PES of the system will increase exponentially. In this case the even distribution of structures shown by the scatter method will mean that it will require more and more structures to be generated. The skew introduced by the GA will mean the physically interesting areas of the PES will be those searched more thoroughly. As was shown in the previous two chapters, the GA performs very well at systems with 130+ atoms, and the scatter approach will not behave as well when applied in these cases.

Chapter 7

Pressure Induced Phase Transitions

7.1 Introduction

In all the studies that have been shown so far in this work, the principle method of comparison involved the enthalpy of the structure, $\mathcal{H} = \mathcal{U} + \mathcal{PV}$. In the Lennard–Jones potential HCP is always the most stable phase, and FCC is always meta–stable but the Dzugutov potential has a number of meta–stable phases which undergo phase transitions at different pressures. It is possible to simulate the effect of an applied pressure and so allow the GA method to study systems at any pressure. With an applied pressure a meta–stable phase at zero–pressure may become the most stable phase at increased pressure. Minimising within this enthalpy surface will allow for the investigation of pressure–induced phase transitions.

While the GA may not be able to produce from a single calculation the minimum enthalpy structure at a given pressure it should be able, as in the polymorph–search application described in the previous chapter, to suggest structures of interest. The GA may not be able to give the exact pressure at which phase transitions between different structures occur, but it will be able to suggest stable phases. Any transitions between these phases can then be examined by other methods.

GA calculations will be started at a number of different pressures and the population will be allowed to vary both the cell and the number of atoms within that cell. This method therefore make no assumptions about the cell shape *or* the number of atoms within that cell, unlike in the previous chapter where the number of atoms was kept fixed. Any phases suggested can then be compared for stability at different pressures.

As was shown in chapters 4 and 5 the GA is capable of dealing with large

systems of up to 150–atoms. In this chapter I will study systems of over 240–atoms by using the Dzugutov potential [Dzugutov, 1992] (see section 3.2.3).

7.2 The use of the Dzugutov Potential

The extra repulsive component that this potential exhibits means that this potential has some interesting phases at varying temperatures and pressures, which have been catalogued in detail in Roth and Denton [2000]. However, due to the constraints of the local BFGS minimiser (section 2.2.4) finite temperature Genetic Algorithm calculations cannot be performed at present, so accessing high temperature regions of the phase diagram is not possible. It is possible to increase the pressure applied to the system, so attempts can be made to probe the system at increasing pressure at zero temperature.

This potential has been used extensively to study quasi–crystals [Roth and Denton, 2000] and in molecular dynamics simulations of liquids with poly-tetrahedral and icosahedral order [Doye *et al.*, 2003], and similar potentials with a second minimia have been developed, although they will not be discussed here. The study by Roth and Denton included a large number of possible structures which were tested for stability. However, it was found that only three phases were stable at zero temperature, body–centred cubic (BCC), the σ–phase and face–centred cubic (FCC).

In Roth and Denton [2000] the system was studied by both molecular dynamics cooling and thermodynamic perturbation theory [Weeks *et al.*, 1971]. The full pressure–temperature phase diagram was determined by MD cooling.

The pressure ranges are given in table 7.1, and the enthalpy per atom of each of the phases suggested by Roth and Denton [2000] at the pressures studied here are given in table 7.2. I have also plotted the energy–volume curve for these phases in figure 7.1. From this curve it is possible to determine the phase transition pressure between phases. The curve plots internal energy, $\mathcal{U}$, against volume, $\mathcal{V}$. The gradient of the line of common tangent between curves will give the pressure, $\mathcal{P}$, of phase coexistence, since

$$\begin{aligned} d\mathcal{U} &= \bar{d}\mathcal{Q} - \bar{d}\mathcal{W} \\ &= \mathcal{T}d\mathcal{S} - \mathcal{P}d\mathcal{V} \end{aligned}$$

where $\bar{d}\mathcal{Q} = \mathcal{T}d\mathcal{S}$ is the infinitesimal heat flow into the system, $\mathcal{T}$ the temperature and $\mathcal{S}$ the entropy of the system, and $\bar{d}\mathcal{W} = \mathcal{P}d\mathcal{V}$ is the infinitesimal work

done by the system. For these calculations $\mathcal{T} = 0$ so

$$\mathrm{d}\mathcal{U} = -\mathcal{P}\mathrm{d}\mathcal{V}. \tag{7.1}$$

This potential also gives rise to tetrahedrally–close–packed (TCP) structures, where space is attempted to be filled by regular stacking of tetrahedron [Szeto and Villain, 1987; Shoemaker and Shoemaker, 1988]. This leads to strain within the cell, and will also create some rather large cells. According to Roth and Denton no TCP phases are stable at zero temperature.

Phase	Pressure Range ‡ (reduced units)	Pressure Range (MPa)
BCC	$P\sigma^3/\varepsilon \leq 1.70$	$P \leq 86.893$
σ–phase †	$1.70 < P\sigma^3/\varepsilon \leq 2.85$	$86.893 < P \leq 138.739$
FCC	$P\sigma^3/\varepsilon > 2.85$	$P > 138.739$

‡ Data taken from Roth and Denton [2000].
† This structure must be relaxed at higher pressures. The fractional co–ordinates of the σ–phase at higher pressure are not the same as those at zero pressure.

Table 7.1: Table giving the pressure ranges for the three stable phases of the Dzugutov potential at $T = 0$K in reduced units.

Pressure (MPa)	Enthalpy/Atom (eV) BCC	σ	FCC
0	-2.7256×10^{-2}	-2.6781×10^{-2}	-2.2241×10^{-2}
50	-1.3023×10^{-2}	-1.2863×10^{-2}	-1.0210×10^{-2}
90	-1.7610×10^{-3}	-1.8942×10^{-3}	7.8191×10^{-4}
100	1.0361×10^{-3}	8.2633×10^{-4}	1.5582×10^{-3}
150	1.4901×10^{-2}	1.4304×10^{-2}	1.3193×10^{-2}

Table 7.2: Table showing the enthalpies of each of the 3 phases suggested by Roth and Denton for the pressures studied.

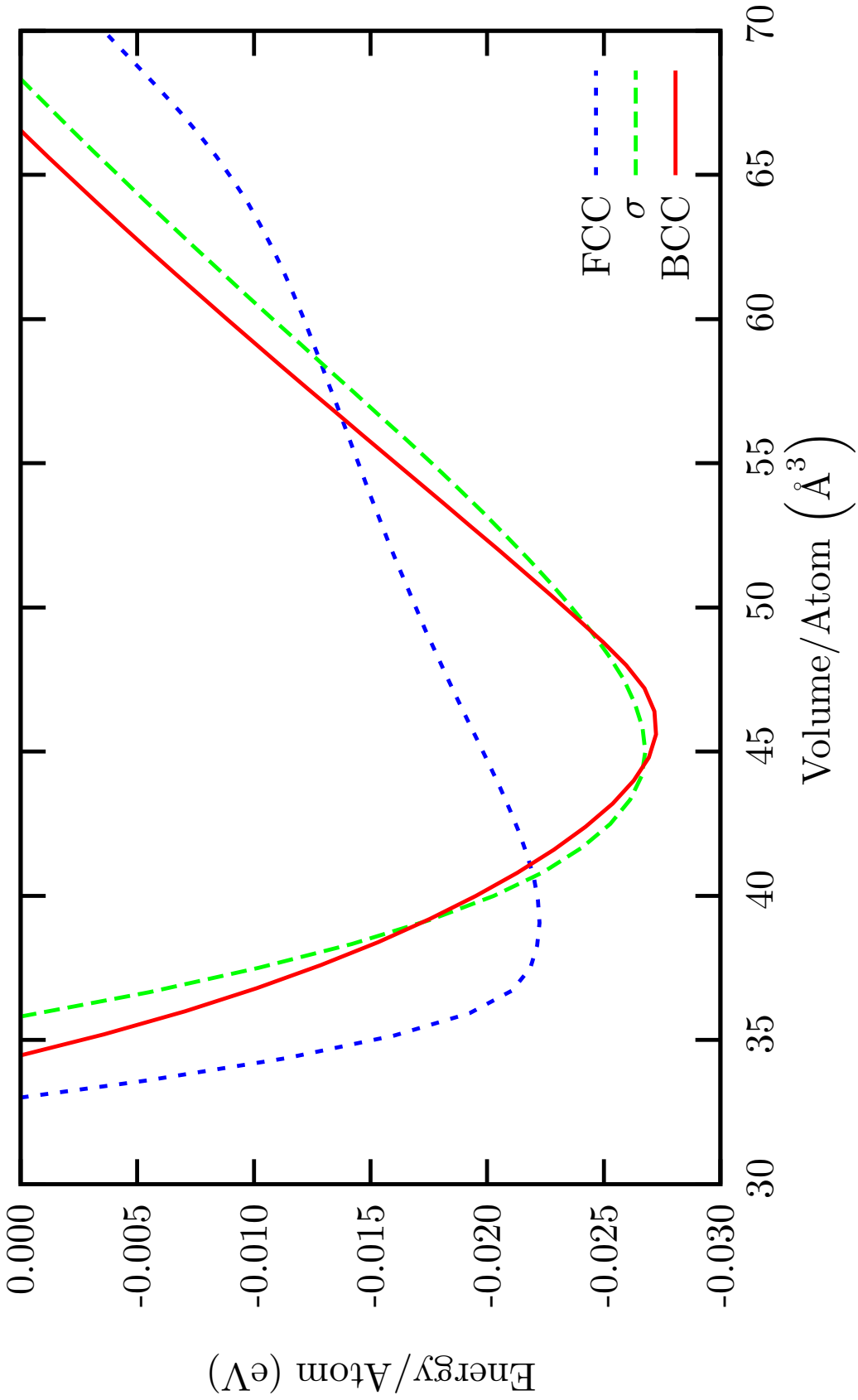

Figure 7.1: Energy–Volume curves for the Dzugutov potential, showing BCC, FCC and the σ–phase. The curve for the σ–phase was made assuming an isotropic expansion due to computational difficulties.

7.2.1 The σ–Phase

There are three phases which should appear at 0K [Roth and Denton, 2000]: a BCC phase, followed by a phase labelled as the σ–phase, which is the structure of β–U [Crystal Lattice Structures, 1995–], and finally FCC at higher pressures. While the BCC and FCC phases can be represented using cubic cells with 2– or 4–atoms respectively, the σ–phase has a complicated 30–atom unit cell which is tetragonal and has lattice parameters of $a = b = 13.680837$, $c = 7.214048$ and $\alpha = \beta = \gamma = 90^\circ$.

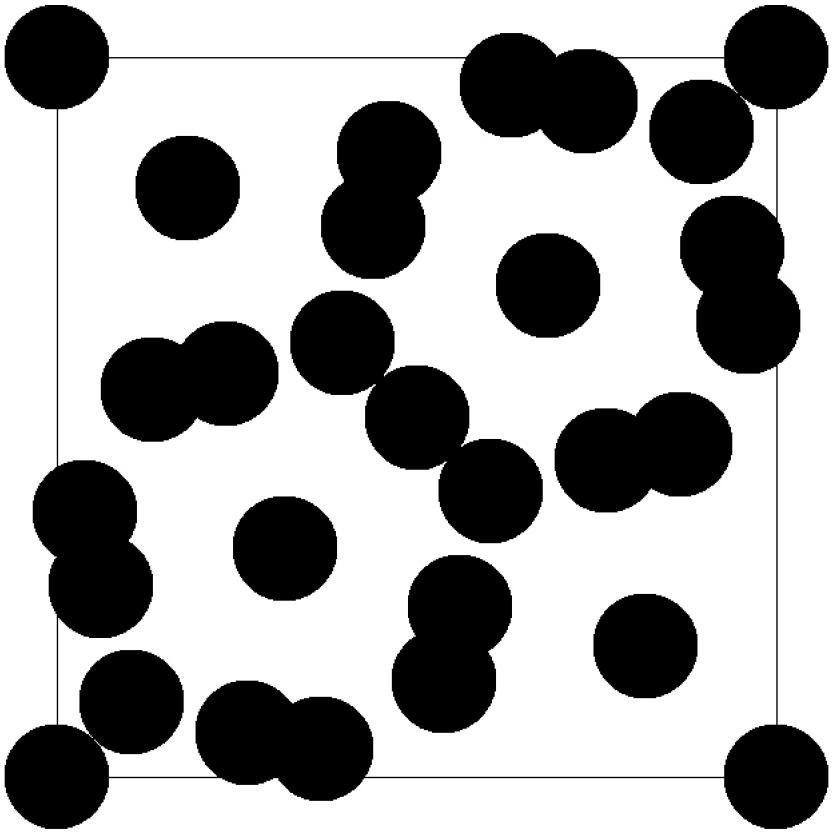

Figure 7.2: The unit cell of the Dzugutov potential σ–phase looking down the $[00\bar{1}]$ direction.

7.3 Results

The aim of these calculations is to determine the stable and meta–stable phases of the Dzugutov potential at different pressures, without assumptions. This potential is a useful test case due to its ease of calculation and because of the comparison study of Roth and Denton.

7.3.1 Exploding and Shrinking Cells

For results obtained using this potential an addition was made to the GA in the crossover step. Previously the atom–number could either be kept fixed or be allowed to vary. For these calculations I added a third option which is to allow the atom number to vary within an allowed amount of the original number of atoms within the input cell.

While this is not necessary in a fixed–cell calculation, for a variable–cell calculation it is essential. Without this constraint it would be possible for the number of atoms to keep decreasing, the cell getting smaller and smaller until the minimum image convention is violated, at which point the calculation will stop. It might also allow a calculation to keep adding atoms at the crossover stage and then allow the cell to grow to accommodate them. In this way the calculation would increase in size and take a longer and longer time for each minimisation step. This percentage cut–off keeps the advantages of a variable atom–number calculation without these problems.

7.3.2 62–Atom Cells

Both the FCC– and BCC–phases require only one atom in the primitive cell (or 4 and 2 atoms respectively in the conventional cubic cell), whereas the σ–phase cell is 30–atoms. Here, the parameterisation of the Dzugutov potential is beneficial, and so in this case it is possible to use a relatively small simulation super–cell, which is preferred since with less atoms there are less possible glassy states. As was discussed earlier in chapter 3, the simulation super–cell needs to be large enough to prevent self interaction. In the Lennard–Jones parameterisation the cut–off radius is 2.5σ, but in the Dzugutov potential it is only 1.94σ. While constructing a cubic 108–atom FCC cell is not possible in the Lennard–Jones case, it is possible to construct a 60–atom σ–phase cell. Running a GA calculation at 100MPa means that the σ–phase should be the preferred structure found.

Initial GA runs on this 60–atom cell proved a little too successful. Out of 16

population members, it was often possible for the σ–phase to found in the zeroth generation, before any breeding or mutation steps. While this shows that the σ–phase exists in a relatively large area of the potential energy surface, in a situation where the "answer" is unknown it may not be the case that it is possible to input the correct number of atoms into the GA. I therefore decided to start with 62–atoms in the initial cell, thus making sure that zeroth generation would not contain any σ–phase structures. Starting with few atoms may mean that as the number of atoms varied during the calculation the cell may shrink and violate the minimum image convention. Starting with 62 atoms, with a maximum allowed variation in the number of atoms of 15%, means that the number of atoms accessible is between 53 and 71.

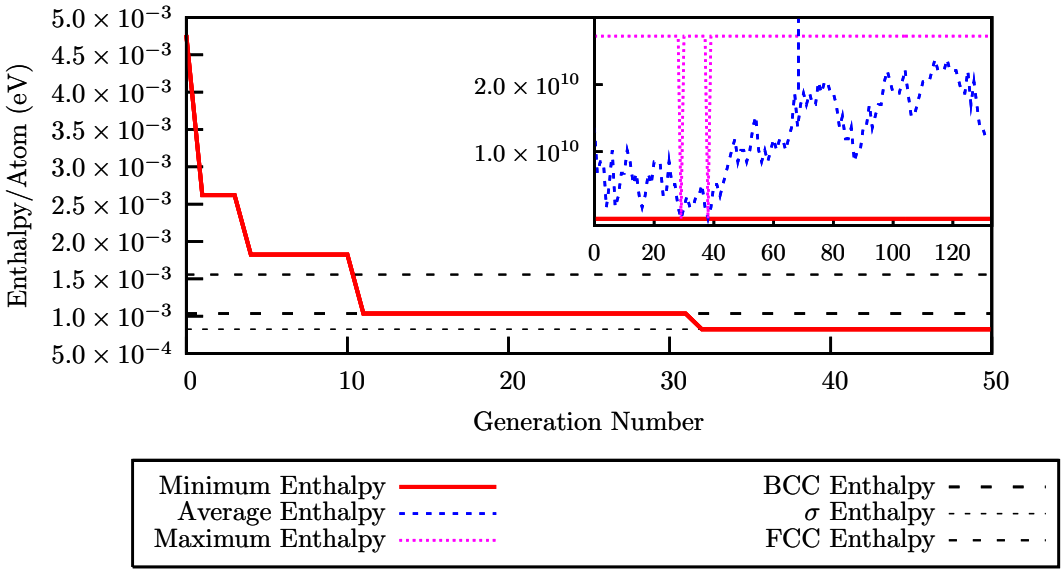

Figure 7.3: Convergence plot of a variable–atom–variable–cell 62–atom σ–phase calculation at 100MPa. The inset shows the complete calculation. The BCC-phase structure is shown in figure 7.4 and the minimum–enthalpy structure found is shown in figure 7.5.

Results using a 62–atom initial cell at varying pressures showed that the GA can find BCC, σ–phase and FCC structures in a small number of generations. The convergence plot shown in figure 7.3 for a 100MPa calculation shows convergence to the BCC–phase (figure 7.4) before the structure then found the σ–phase (figure 7.5). The reason for the high enthalpies in the convergence plot is due to the fact that some structures had not converged properly due to some atoms being in the same place. When this happens the the GA will assign this structure with a very high enthalpy, but will not stop the calculation. The high enthalpy will reduce the probability that this population member will produce offspring.

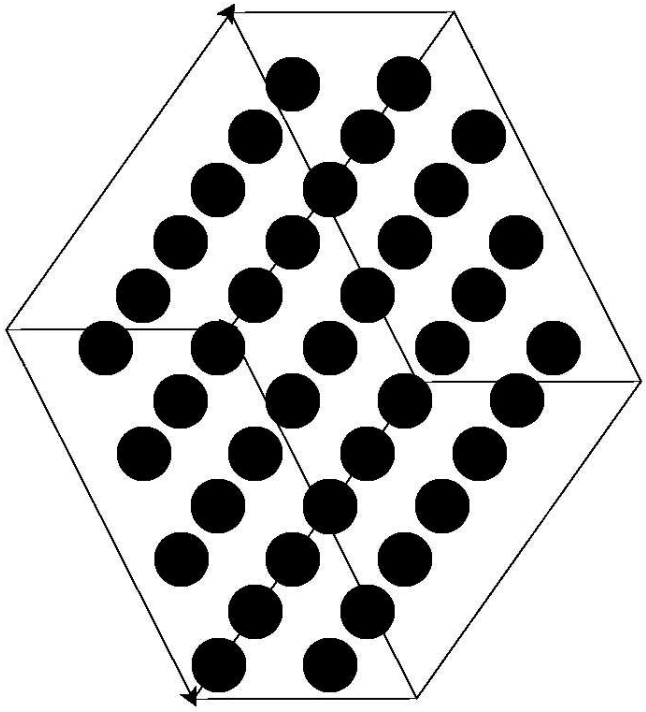

Figure 7.4: The 68–atom BCC–phase found in generation 11 of the calculation shown in figure 7.3.

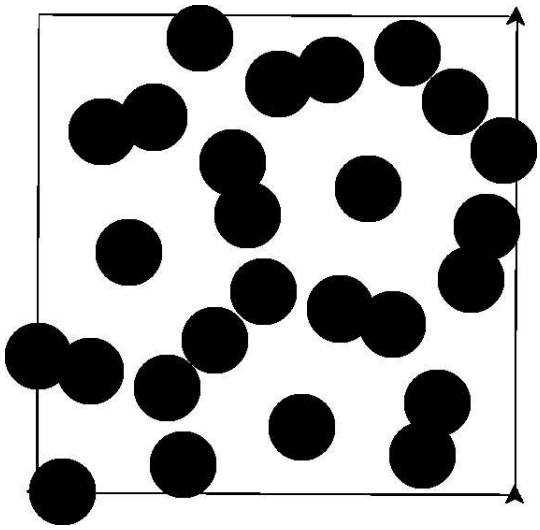

Figure 7.5: The 60–atom σ–phase found in generation 31 of the calculation shown in figure 7.3.

		Number of Each Phase Found				
Pressure (MPa)	Lowest Enthalpy Phase ‡	Higher Enthalpy Phase	BCC	σ	FCC	Lower Enthalpy Phase *
0	BCC	13	8	1	0	0
50	BCC	2	16	1	0	3
100	σ	1	9	11	0	1
150	FCC	1	0	0	15	6

‡ Data taken from Roth and Denton [2000].
* Where the term "Lower Enthalpy" refers to having lower enthalpy than the phase in column 2.

Table 7.3: Summary of results for 62–atom variable–cell, constrained variable–atom–number calculations.

7.3.2.1 Lower–Enthalpy Structures

In another of the 100MPa calculations performed a new structure emerged from the GA. This structure has 65 atoms and has a lower enthalpy to the σ–phase. A convergence plot of this calculation is shown in figure 7.6, where the population had evolved into a meta–stable (at this pressure) BCC–phase. The population then evolved into a lower–enthalpy phase than the σ–phase, which according to Roth and Denton should be the stable phase at this pressure.

Further lower–enthalpy phases were found, at all pressures other than zero pressure, although calculations on these phases at zero pressure found them all to be more stable than either BCC, σ or FCC. A summary of all the calculations performed is shown in table 7.3. As can be seen, the GA was successful at finding the previously found phases, as well as a number of higher–enthalpy phases.

I will be using the radial distribution function (RDF), $g(r)$, (also called the pair correlation function) as a useful quantity for using to distinguish between different solid structures. The RDF describes how many atoms may be found a distance r from any other atom in the material. Different solid structures have distinctive RDFs (which can be seen below in figure 7.10). The RDFs of solid structures such as BCC or FCC usually contain well defined peaks, whereas in the RDF of a more disordered structure the peaks become smeared, indicating the distortions to the lattice structure.

In total there were three possible new phases found, the RDFs of which are shown in figure 7.7. Structures "*c*", "*d*", "*g*" and "*h*" have similar radial distribution functions, and structures "*c*" and "*d*" have the same enthalpy per atom, despite having a difference of 6 atoms between these cells.

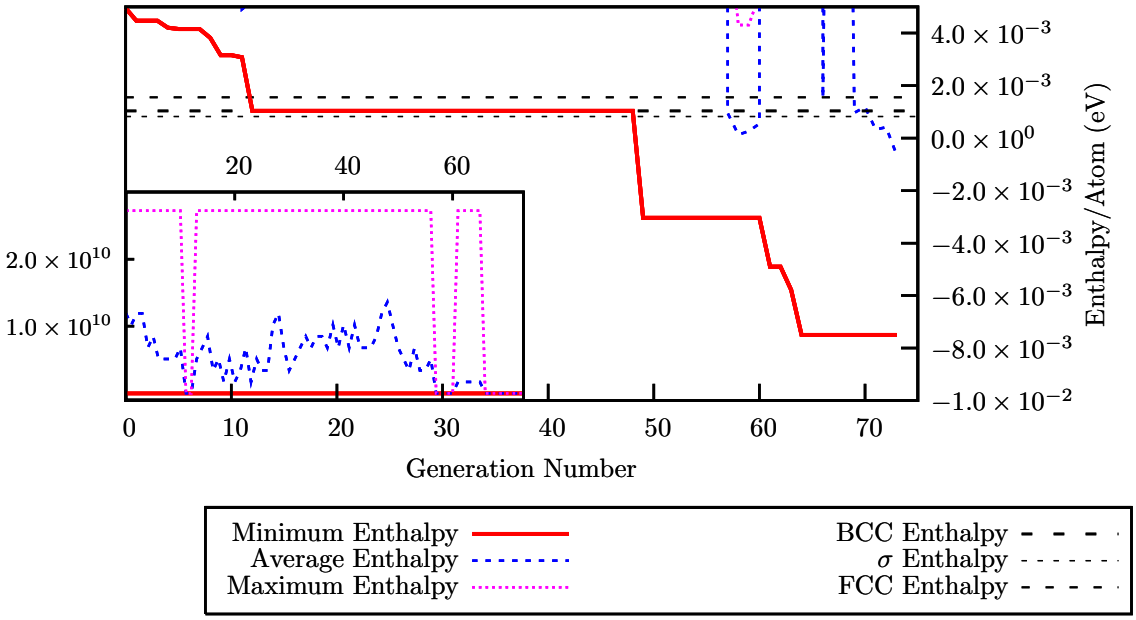

Figure 7.6: Convergence plot of a variable–atom–variable–cell 62–atom σ–phase calculation, giving rise to a previously unknown phase. The inset shows the complete calculation. The minimum–enthalpy structure found is shown in figure 7.8 and has 65–atoms.

Structures "*g*" and "*h*" have very similar RDFs which have peak positions similar to those of "*c*" and "*d*", but whose peak positions are slightly smeared. The higher enthalpy per atom for these structures indicates that these *are* the same structure as "*c*" and "*d*", but structures "*g*" and "*h*" have defect differences.

The number of atoms difference between structure "*g*" and structure "*h*" is 6 atoms, although these structures do not have the same enthalpy/atom, having 63 and 69 atoms respectively. This could indicate that the unit cell of this structure (characterised by the structures "*c*" and "*d*", and with "*g*" and "*h*" having higher–enthalpy defects) should contain either 2 or 6 atoms. A unit cell of 3 atoms would mean that a 63 or 69 atom structure should be able to have the same enthalpy per atom as a 60 or 66 atom structure. A view of structure "*c*" is shown in figure 7.9.

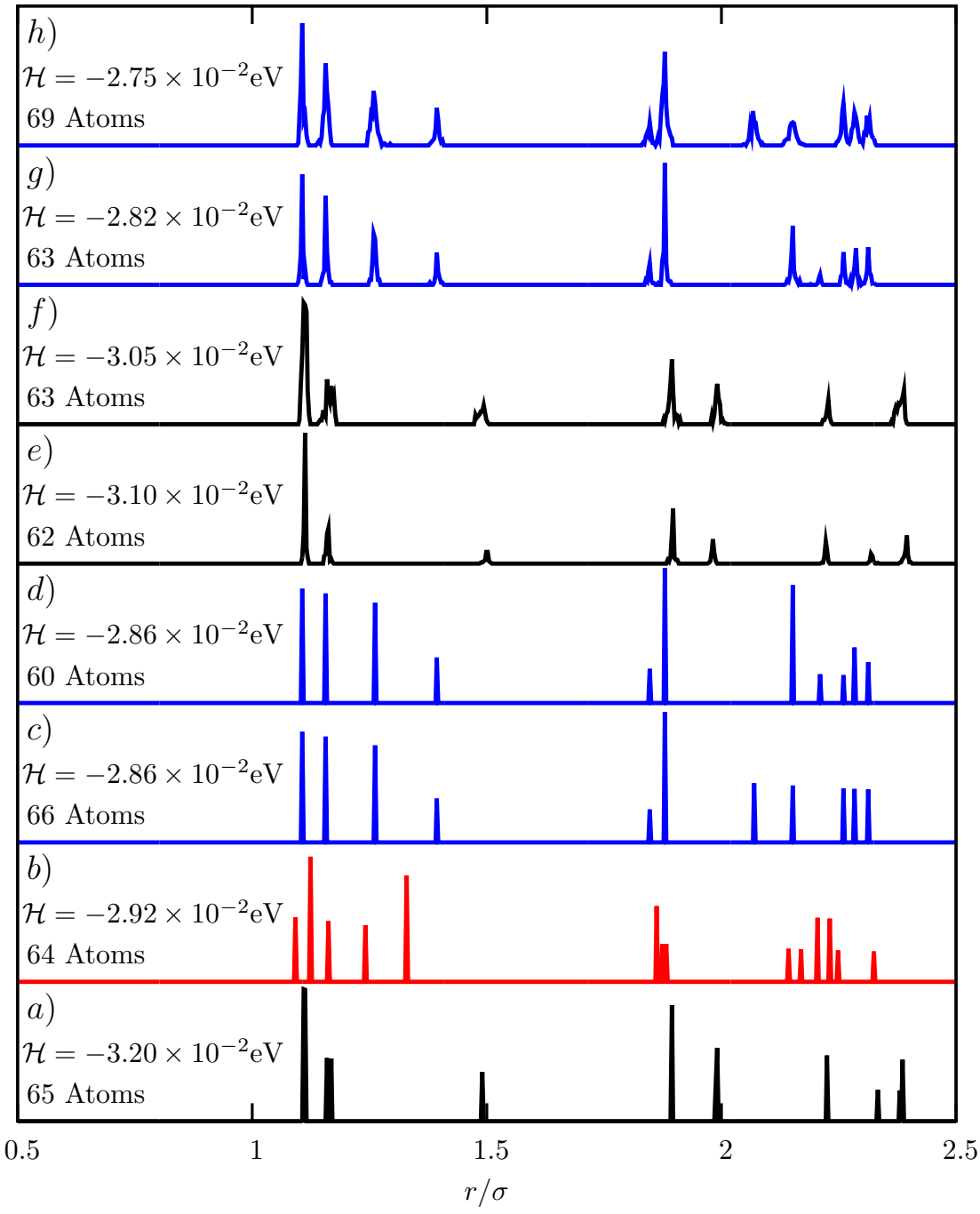

Figure 7.7: Comparison of the radial distribution function, $g(r)$, and enthalpy per atom for the lower–enthalpy structures found.

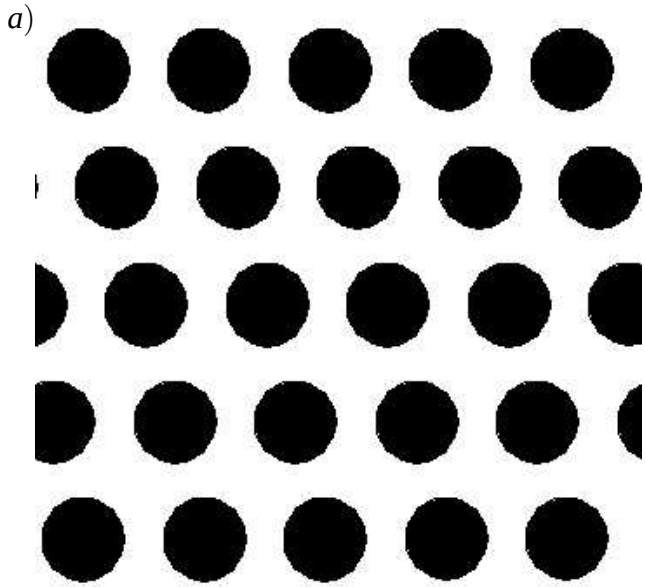

Figure 7.8: Structure "*a*" from figure 7.7: 65–atom phase found in generation 64 of the calculation shown in figure 7.6.

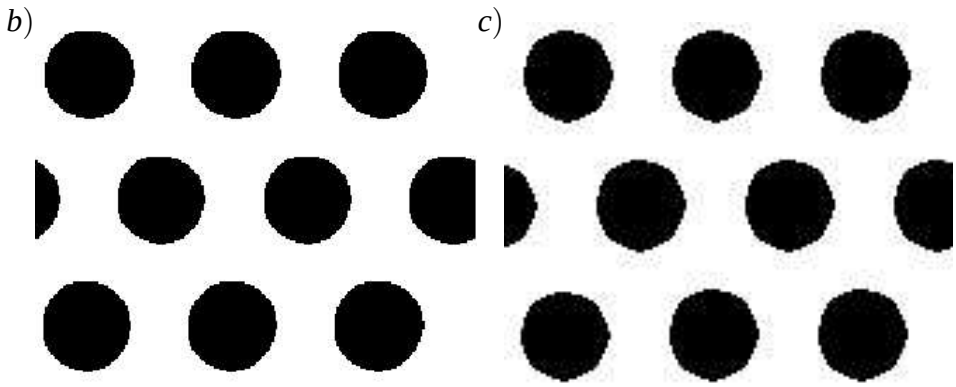

Figure 7.9: Structure "*b*" from figure 7.7, 64 atoms, and structure "*c*" from figure 7.7, 66 atoms

The radial distribution function of structure "*b*" is unique, although it does have some similarities to those of "*c*" and "*d*". The RDF of "*a*" is also different to that of "*b*", "*c*" and "*d*", although it is similar to "*e*" and "*f*", in which the peaks are also smeared compared to those in "*a*" (figure 7.8). Again I suggest that these are the same structures with only stacking fault or defect differences. Structure "*a*" can be seen in figure 7.8 and shows stacking which requires 5 layers until it repeats. The stacking of structures "*b*" and "*c*", shown in figure 7.9 are similar to each other, but different from structure "*a*". The ordering of figure 7.7 is such that distinct structures are placed in the "*a*", "*b*", "*c*" and "*d*" positions, even though structures "*e*" and "*f*" have a lower enthalpy per atom than "*b*", "*c*" and "*d*". In fact, in figure 7.6 the system passes through structure "*c*" before finding structure "*a*".

A plot comparing the radial distribution functions of BCC, FCC, HCP and σ–phase structures with these three phases is shown in figure 7.10. As can be seen, these new phases have distinct peaks that are not present in either the HCP, FCC or BCC plots. The RDF of a TCP–phase structure should appear as smeared out σ–phase structure [Roth and Denton, 2000]. The RDFs of phases "*a*", "*b*" and "*c*" are quite distinct and clear, indicating a small number of atoms in the unit cell.

Taking the cells of these structures and calculating the energy–volume curves for these structures leads to the plot in figure 7.11. As can be seen, structure "*a*" should be the most stable phase at all positive pressures. A summary of the enthalpies of these three phases at a number of pressures is given in table 7.4.

Pressure	Enthalpy/Atom (eV)		
(MPa)	New Phase "*a*"	New Phase "*b*"	New Phase "*c*"
0	-3.1968×10^{-2}	-2.9235×10^{-2}	-2.8630×10^{-2}
50	-1.9605×10^{-2}	-1.4941×10^{-2}	-1.4939×10^{-2}
100	-7.5010×10^{-3}	-3.0522×10^{-3}	-3.0337×10^{-2}
150	4.3883×10^{-3}	8.6854×10^{-3}	8.7353×10^{-2}

Table 7.4: Table showing the enthalpy per atom of the three distinct lower enthalpy phases suggested by the GA for the pressures studied.

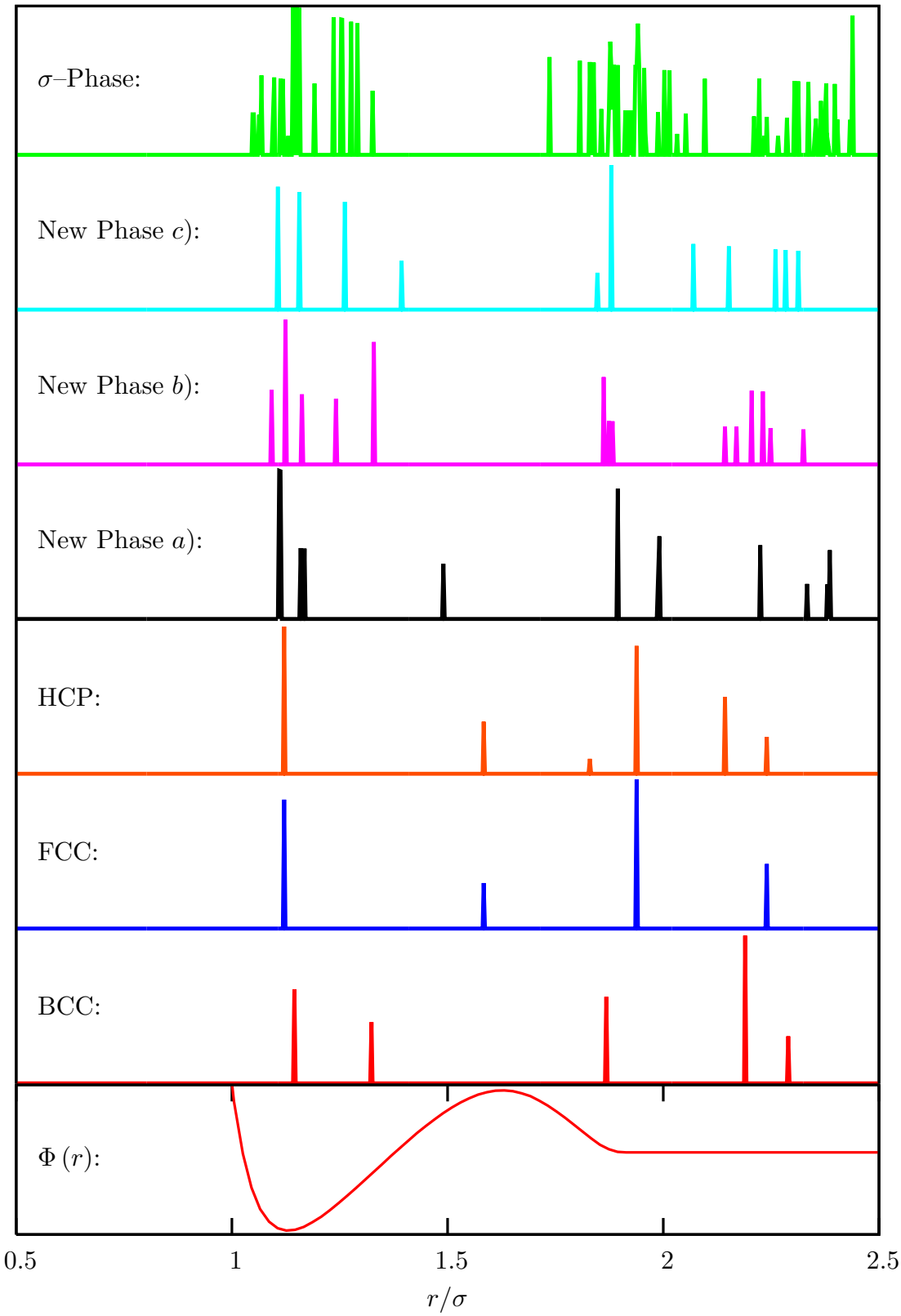

Figure 7.10: Comparison of the radial distribution function, $g(r)$, for the distinct lower–enthalpy structures found with BCC, FCC, HCP and the σ–phase. The Dzugutov Potential is also shown.

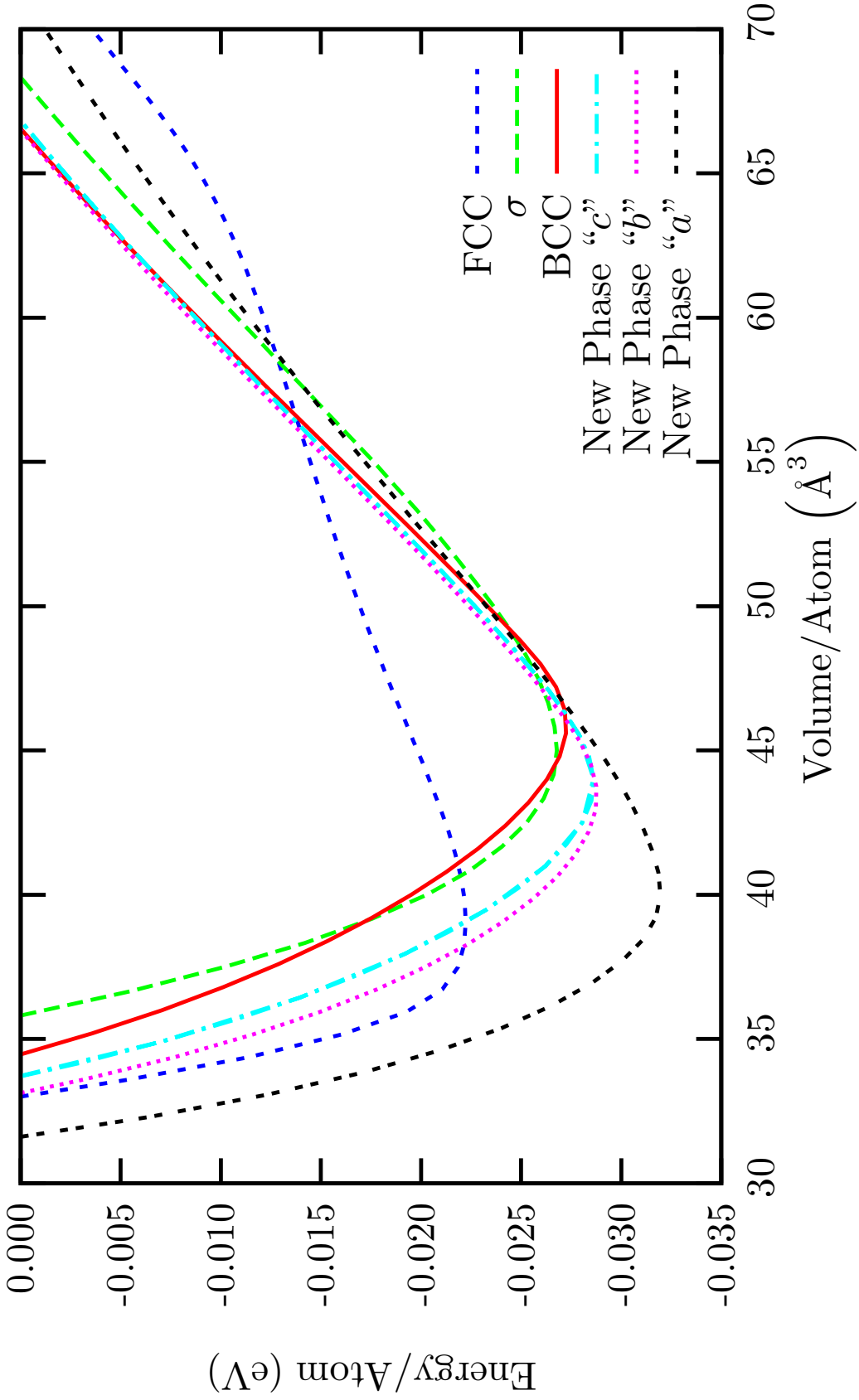

Figure 7.11: Energy–Volume curve for the Dzugutov potential showing the three new phases. The curves for the σ–phase and structures *"a"*, *"b"* and *"c"* were calculated assuming an isotropic expansion due to computational difficulties.

7.3.2.2 Characterisation

In an attempt to characterise these phases, I used the symmetry finder from Materials Studio Modelling (MSM) used in chapter 6 [Accelrys, 2001–] to attempt to find the unit cell of the three new phases. For phase "*a*" the symmetry finder suggested a symmetry of C2/m (12) at a very fine tolerance (0.001 Å). The symmetry of structure "*b*" was also suggested to be of space group C2/m (12) and both "*c*" and "*d*" were given the same space group of Fmmm (69) (giving the same suggested cell). A summary of these suggested cells is given in table 7.5. All suggested cells have 1 atom in the primitive cell. The cells were produced by calculating the symmetry and then creating the primitive 1–atom cell from this symmetry. These structures were then tessellated up to a $5 \times 5 \times 5$ super–cell before being relaxed in CASTEP. If the correct unit cell has been determined then this relaxation should leave the structure unchanged.

Phase	Symmetry	Fractional Co–ordinates	Unit Cell	Relaxation
a)	C2/m (12)	$\left(\frac{1}{2}, \frac{1}{2}, 0\right)$	Volume=40.0973 Å $a = b = 3.7960$ Å $c = 5.08812$ Å $\alpha = \beta = 132.323^\circ$ $\gamma = 63.2806^\circ$	BCC Im$\bar{3}$m (229)
b)	C2/m (12)	$(0, 0, 0)$	Volume=43.4450 Å $a = b = 3.83654$ Å $c = 3.72546$ Å $\alpha = \beta = 73.6831^\circ$ $\gamma = 113.005^\circ$	BCC Im$\bar{3}$m (229)
c)	Fmmm (69)	$(0, 0, 0)$	Volume=43.9087 Å $a = 3.77317$ Å, $b = 3.94809$ Å $c = 4.30249$ Å $\alpha = 54.2225^\circ$, $\beta = 58.0928^\circ$ $\gamma = 67.6846^\circ$	BCC Im$\bar{3}$m (229)

Table 7.5: Table showing the suggested structures from Materials Studio Modelling [Accelrys, 2001–]. When relaxed all suggested structures became BCC rather than the original phase.

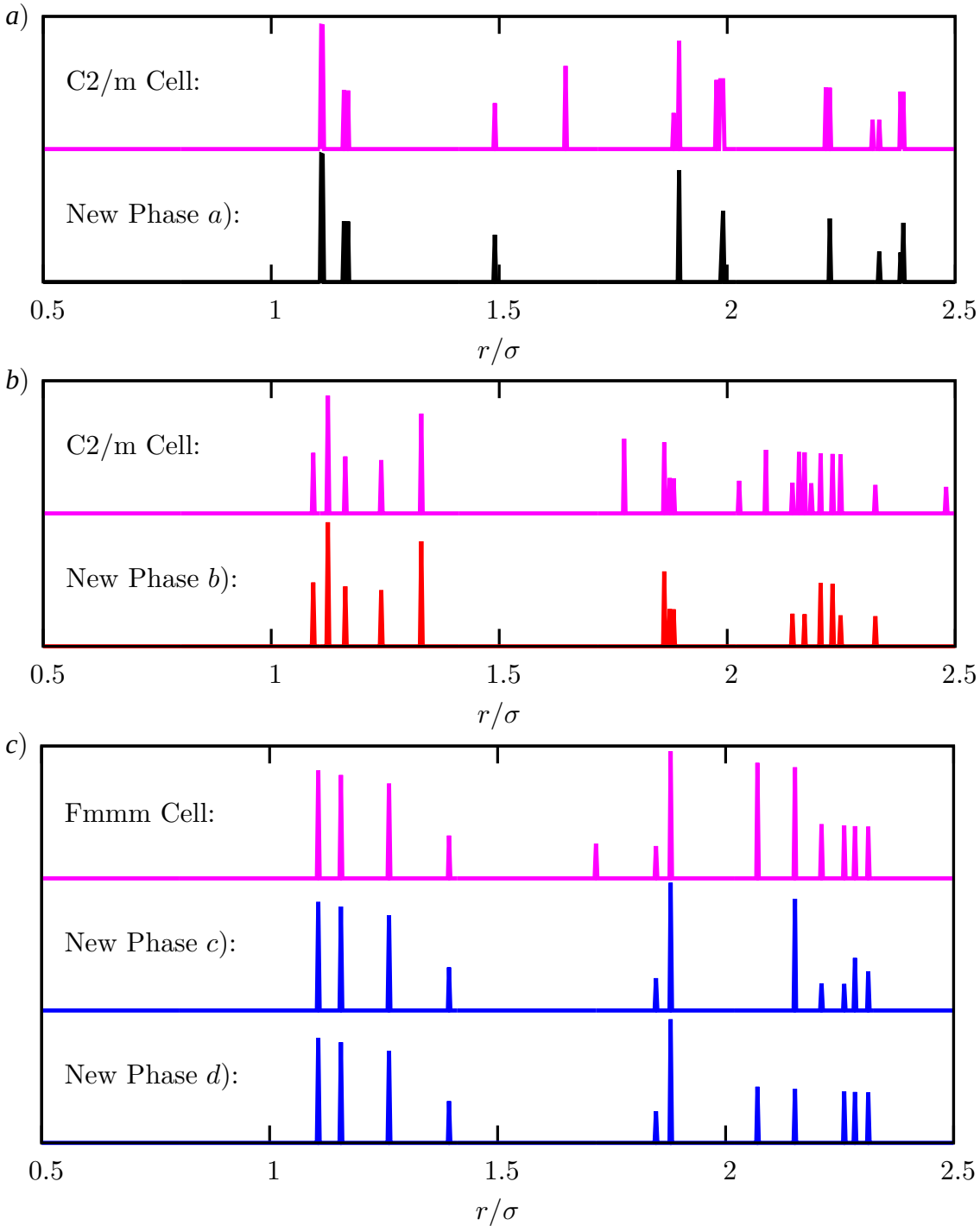

Figure 7.12: Comparison of the radial distribution function, $g(r)$, for suggested structure from the symmetry finder and *a*) Phase "*a*"; *b*) Phase "*b*"; *c*) Phases "*c*" and "*d*".

However, in all cases, creating a super–cell from the suggested primitive cell and relaxing the structure did not give the same results as before, and the structures relaxed into the BCC phase (space group $Im\bar{3}m$ (229)). Figure 7.12 shows a comparison of the RDFs of the GA determined phases and the structures suggested by MSM. While these RDFs are repeated almost exactly in all cases, there are some differences, most notably with extra peaks around the $1.7\,r/\sigma$ positions. The suggested cell for structures "*c*" and "*d*" not only contains the similar peaks between these two structures, but also has the separate peaks that appear in these structures (i.e. the peak at $2.1\,r/\sigma$ that is present in "*d*" but missing in "*c*"). It is clear that MSM is creating the wrong unit cells of these structures, despite these cells having some of the characteristics of the correct structures.

This extra structure within the super–cell seems to be forcing the system out of these low–enthalpy structures, and into the BCC–basin. These areas of the potential energy surface seem to be close, since these structures originally evolved from a system with a BCC meta–stable structure as the lowest enthalpy member (figure 7.6).

I also attempted to use the FINDSYM packing from the ISOTROPY [Stokes and Hatch, 1995–
] but this also produced structures ("found" using the highest tolerance settings) that minimised to BCC. It is possible that structure "*a*" has a 65–atom unit cell, just as structure "*b*" has a 64–atom unit cell, although I think this unlikely. The largest unit cell possible for structures "*c*" and "*d*" is 6 atoms, assuming that these structures are both defect–free.

All attempts so far to determine a smaller unit cell of these structures has failed, and I am unable to give the symmetry for a smaller unit cell for any of these lower enthalpy phases. The funnels that these structures exist within in the potential energy surface may be very narrow (as indicated by the study of Roth and Denton being unable to find these phases). If this is the case then the small perturbation made by the symmetry finder moving the atoms is enough to disrupt the lattice and move the structure into the much larger BCC basin.

7.4 Large Cell Calculations

I want to see how well the GA will operate with a large number of atoms in a variable cell calculation. As was shown in chapters 4 and 5 the GA could cope very well with 150–atoms using a Lennard–Jones potential in a fixed–cell case, and it is tempting to attempt minimisations with even larger numbers of atoms here.

For these reasons I also performed calculations using input cells (before randomisation) of a $5 \times 5 \times 5$–BCC cell of 250 atoms, a $4 \times 4 \times 4$–FCC of 256 atoms, and a $2 \times 2 \times 2$–σ–phase cell of 240 atoms. Allowing the number of atoms to vary by 15% on either side of 240, 250 or 256 atom input cells should mean that all structures are accessible.

As shown in the previous section, the GA is able to find the σ–phase using a cell of 62–atoms. The FCC–phase became accessible at higher pressures as was expected, and the BCC–phase appeared at lower pressure, and at pressures where the σ–phase has the lowest enthalpy.

A cell with $230 \rightarrow 260$ atoms has an incredibly complex potential energy surface (since the number of minimia increases exponentially with the number of atoms), with a large number of glassy high enthalpy states that the system would be required to pass though to access the σ–phase from a BCC–phase.

The external pressure for these calculations was set to 0.09 GPa at which the σ–phase is slightly more favourable in enthalpy than the BCC–phase. The structure factor weighting factor w was set to 0.75 as in previous calculations. However, I found that from a 240–atom cell the system was unable to find the σ–phase, with only BCC phases appearing. The convergence graphs for two independent calculations can be seen in figures 7.13 and 7.16.

Both the BCC structures found are in the space group $Im\bar{3}m$ (229) but have non–cubic cells. Both have 252–atoms but have different cells and are orientated differently within them, as can been seen in figures 7.15 and 7.17.

The calculation shown in figure 7.13 is interesting in that it appears that the FCC–phase was found as the systems minimised to the BCC phase in the generations 37–47. In fact this was a glassy structure more closely resembling the σ–phase (figure 7.14) but with stacking faults, dislocations, an incorrect number of atoms (251 compared to 240) and a non–tetragonal cell. Even though the GA found a σ–like structure, it was unable to minimise this further, and eventually the population evolved into the BCC–phase. Since the increase of one atom is easily done in crossover, and a 252–atom structure might easily fall into a BCC–phase part of the potential energy surface this could explain

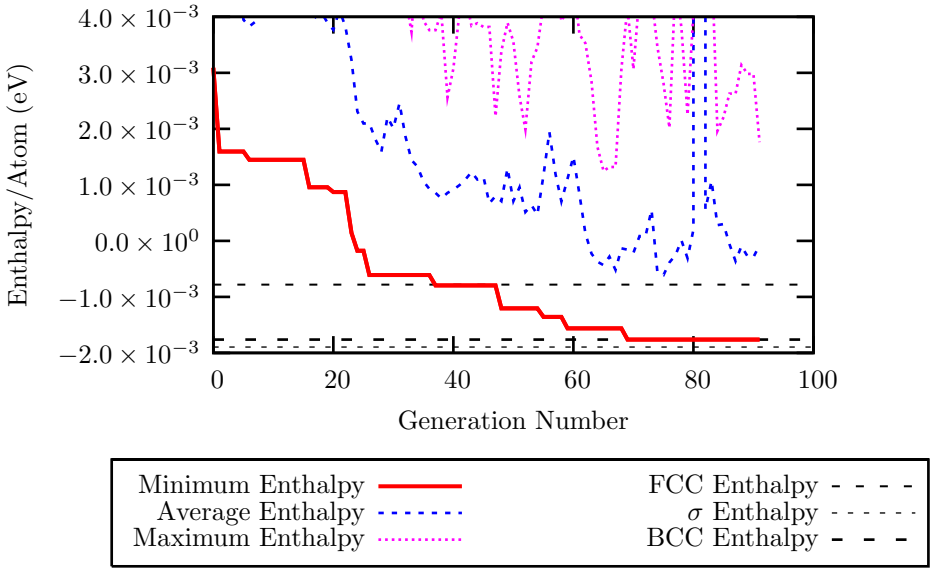

Figure 7.13: Convergence plot of a large variable–atom–variable–cell calculation. Two of the structures found are shown in figures 7.14 and 7.15.

why a structure with some σ–phase characteristics evolved into a BCC–phase structure.

I find it interesting that the BCC–phases found, while both having the same structure and number of atoms, have markedly different cells. Although the calculation has no knowledge of the super–cell used in the calculation, and so the atoms can have any orientation in space, the periodic cuts force a weak coupling between the atoms and the cell, as was seen in section 4.2.1.4. However, the variable–cell in this case has meant that the cell can be any periodic cell within the lattice, and so these cells are different. Although the more complicated σ–phase was not found it is gratifying to see that the GA can still find low–enthalpy meta–stable structures with a large number of atoms.

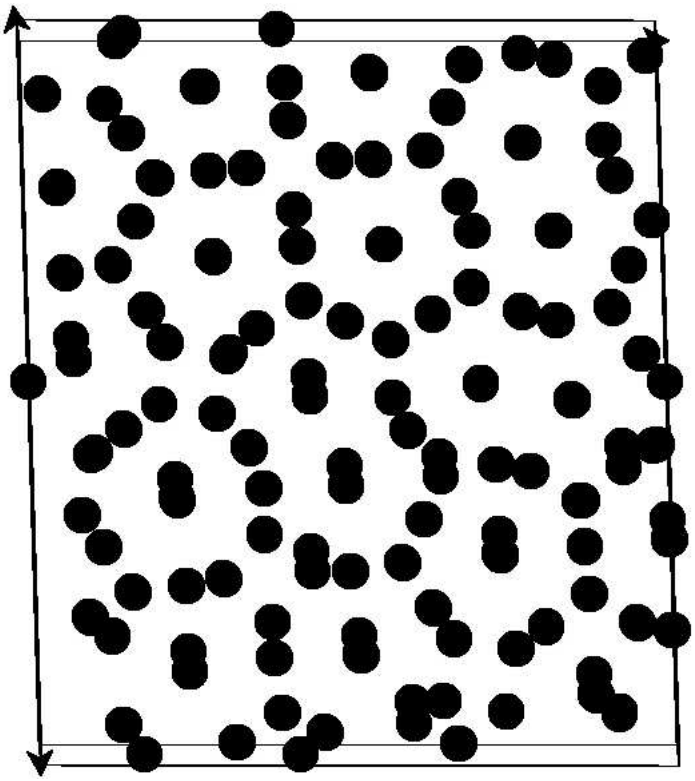

Figure 7.14: Glassy 251–atom structure with some σ–phase characteristics from the calculation shown in figure 7.13.

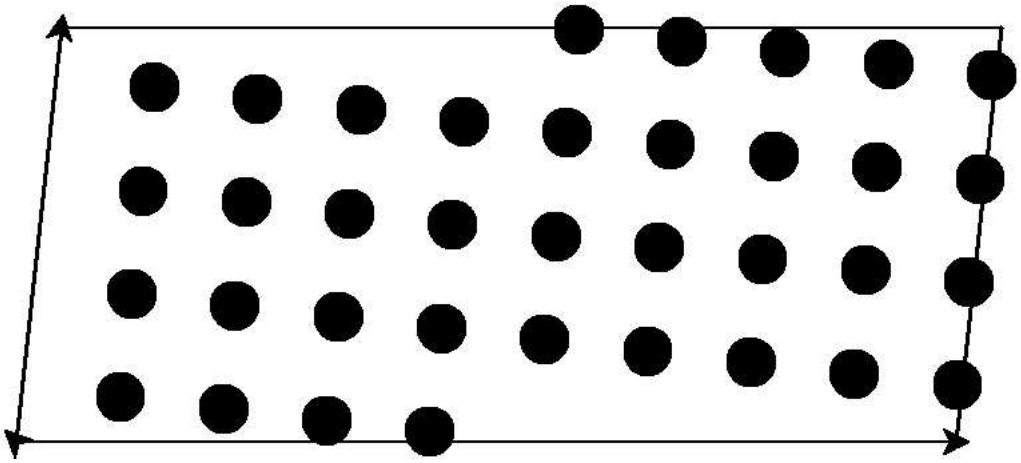

Figure 7.15: A 252–atom BCC–phase structure found from the calculation shown in figure 7.13.

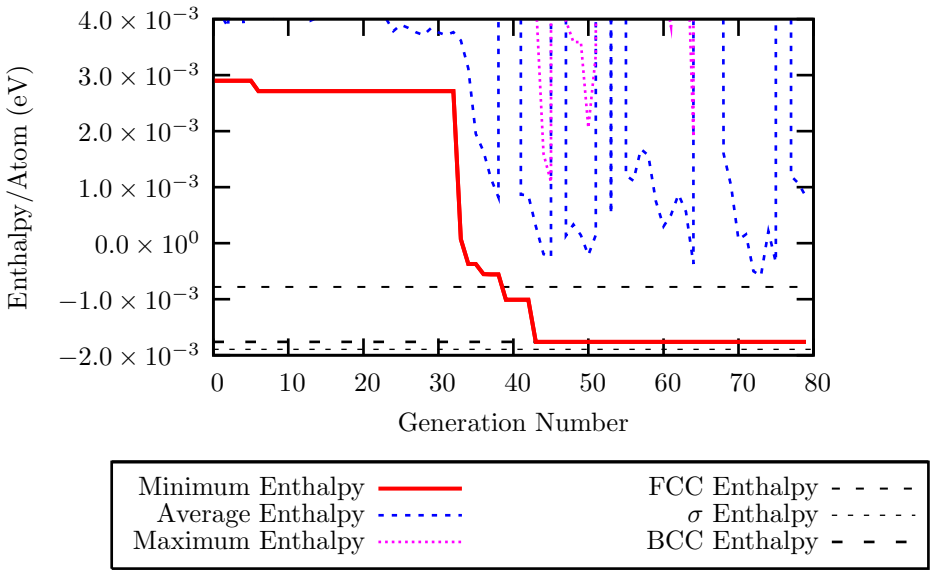

Figure 7.16: Convergence plot of a second large variable–atom–variable–cell calculation. The minimum–enthalpy structure found is shown in figure 7.17.

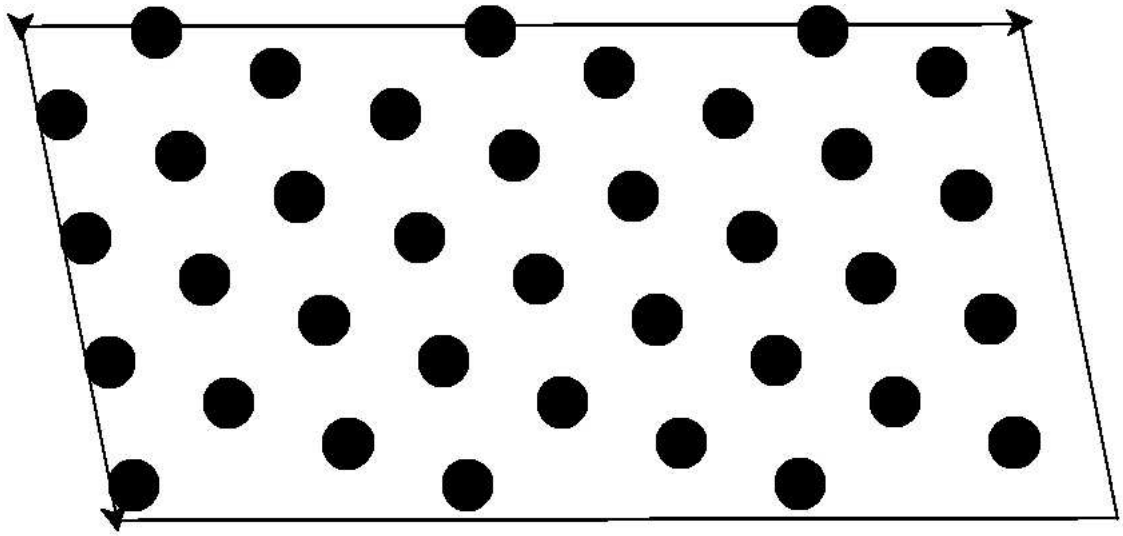

Figure 7.17: A 252–atom BCC–phase structure found from the calculation shown in figure 7.16.

7.5 Conclusions

The Dzugutov Potential is an interesting empirical potential which exhibits a number of different phases at higher pressures but at zero–temperature, making it a good test potential for the GA. Using 62–atom cells the GA found all previously reported structures by varying the number of atoms in the unit cell. During these calculations the GA also found three previously unreported phases, one of which has been shown to be the most stable phase at all positive pressures. However, attempts to determine the symmetry of smaller unit cells of these structures has so far proved frustrating, and any suggested unit cells have relaxed into BCC structures. The fact that these phases are so sensitive to the initial placement of atoms may mean that they are unstable at finite temperature, which may explain why they have not been reported previously.

Calculations performed on large cells of more than 230 atoms did not manage to repeat the successes of the 62–atom cells, although they were able to find BCC structures. This means that the GA can still operate effectively as the number of atoms increases, even in a variable–cell case.

The action of increasing pressure also reduced the number of higher–enthalpy local minima within which the system became trapped, and increased the number of lower–enthalpy phases found in the calculations. While the GA is not best suited to determine the pressure at which the phase–transition occurs, it can be used to suggest candidate phases. With these phases it is then possible to construct energy–volume curves which can be used to determine the phase–transition pressure.

Chapter 8

Conclusions

When I started this work, I hoped that I would be able to create a framework that would allow the prediction of crystal structures from first principles. In chapter 2 I outlined the methodology of Genetic Algorithms, and showed how they had already been applied to some problems in solid–state physics. While these methods had shown some excellent results there is always scope for improvement. I then outlined some methods for parameterising systems that fit within the GA approach in chapter 3, using both empirical potentials and the *ab initio* Density Functional Theory.

Following on from the real–space encoded GA methods of Deaven and Ho [1995], which cut the structure into halves and then swapped these halves to make new structures, in chapter 4 I outlined the methodology of a Genetic Algorithm that uses the periodic nature of the simulation system to aid the convergence to the minimum–enthalpy structure. The key feature of this method is the use of a crossover procedure which is also periodic with the system. Applying this method to the Lennard–Jones potential, this technique showed a faster convergence to the final structure that the use of a planar cut in the crossover procedure. This method does not require that the number of atoms in each population member be fixed during the course of the calculation, in fact, results are improved if the number of atoms is allowed to vary between members.

Building on this method, in chapter 5 I derived a new fitness function, based on the structure factor (equation 5.1), that could differentiate between structures during the course of a calculation, and attempt to force the system away from already found structures. Results were compared against those of the previous chapter and not only showed more structural diversity within the population, these results also showed a faster convergence with increasing weight being placed on the results of the structure factor comparison.

In chapter 6 I applied the GA method developed to search for carbon polymorphs. This search also involved the use of DFT that makes as few assumptions about the system as possible, although there are other methods, such as Quantum Monte Carlo [Foulkes *et al.*, 2001], that make even less assumptions. This chapter also introduced a vital piece of functionality within the method that I am developing: the ability to breed structures with different cells. Without this ability it would be necessary to perform this method over all possible cell types. This technique was compared against the scatter method of Pickard and Needs [2006] and showed that the distribution of structures using the GA was skewed towards lower–enthalpy structures, while still finding some higher–enthalpy ones. Out of the 56 structures created using both methods, a greater number of different symmetries was found using the GA. Further work will be required to show that the GA would be the preferred method in all cases over the scatter method.

These methods were then applied to search for phases of the Dzugutov potential at different pressures. Using input cells with approximately 60 atoms the method was able to find all previously known phases. Using these same input cells three new phases were also found, although these have yet to be fully characterised. Using input cells with approximately 250 atoms the BCC–phase was found. While these results for large cells showed that there is still some improvement that could be made to force the system to explore other possible structures, the fact that this method was able to find perfect lattice structures without defects and stacking faults in cells of this size is encouraging. While these results used an empirical potential they show that this method can be applied to large system successfully. This means that large–scale *ab initio* systems could also be studied using this method.

I believe that I have outlined a consistent framework that should allow the determination of crystal structures from first principles. The methods developed can be applied to systems without any bias towards the number of atoms in the cell or the cell shape. In the next chapter I will outline a number of possible extensions and improvements to the method. All the results presented here have been from cells with only one species, and I will outline a simple extension that will allow for multi–species calculations. I will also show how calculations on surface and interface systems can be performed using these techniques, before discussing the possible extension to large systems using Neural Networks, and the difficulties associated with extending the method to finite temperature systems.

Chapter 9

Future Applications

9.1 Introduction

While I consider that the method described in this study has been well validated on the systems described, the technique itself is also in its infancy, and is being improved continuously. I will now describe some ways in which this is currently being done, or how the method may progress in the future.

9.2 Extension to Multi–Species

Effectively multi–species calculations can be performed in exactly the same way as single–species calculations, but with a few minor modifications.

Algorithm 4 Multi–Species Crossover

1: A cut is defined by calculating a wavelength and amplitude.
2: Loop over the number of different species of atom.
3: Consider the atoms in species 1.
4: This cut is compared against the atoms in the current species:
5: **if** all required crossover constraints are fulfilled **then**
6: EXIT
7: Change to species 2 and goto line 4.
8: **else**
9: Start again with a new cut; goto line 3.
10: **end if**
11: The constraints must be fulfilled for the same cut for all species.

It is also necessary to consider what is meant by the crossover constraints; as well as fixing the atom–number it is also possible to fix the stoichiometry of the system. If doing a calculation on water for instance, it might be better to allow

an extra oxygen atom only if two extra hydrogen atoms were also allowed (or multiples thereof), rather than fixing the number of atoms per species at the start of the calculation. This is now comparable to a variable–atom number calculation.

In the mutation step it is also possible to add in a transmutation operation, where the species type of an atom is swapped with that of a different atom, i.e., if an atom of species A is at position $\mathbf{x}_1$ and an atom of species B is at position $\mathbf{x}_2$ then after the transmutation step the atom at $\mathbf{x}_1$ is of species B and the atom at position $\mathbf{x}_2$ is of species A. During this step the total numbers of species A and species B will not change.

This may be the preferred way of doing a variable–atom number calculation since a change in stoichiometry may change the physics or chemistry of a situation in unwanted ways. It also may be desired to allow different possibilities to be presented and considered by the algorithm which may be counter–intuitive, but could perhaps lead to interesting new structures. This also allows a greater ability to explore the PES. As was seen in chapters 4 and 7 even in a variable–atom calculation the number of atoms in the minimum enthalpy configuration was the correct one, even though the number of atoms changed through the generations, and the system converged faster than if the number of atoms were fixed at the start of the calculation. Allowing a change in stoichiometry may have similar benefits.

This method has been coded into the GA, but is still undergoing a testing phase, and so no results have been presented using this extension to the method. This technique will form the basis of future studies.

9.3 Improvements to the Fitness Function

As was shown in chapter 5 the use of the structure factor could improve the fitness calculation by weighting against similar structures. In the formalism shown it may be possible for two similar structures to have different structure factors. While this is unwanted, it is acceptable. However, it is also possible that two dissimilar structures would have the same structure factor. This is less acceptable - this means that one of the structures will be weighted against even though it should be weighted more highly.

Turning to the world of signal processing a suggestion presents itself, the *Bispectrum* function [Mendel, 1991];

$$B(\mathbf{k}_1, \mathbf{k}_2) = F(\mathbf{k}_1) F(\mathbf{k}_2) F^*(\mathbf{k}_1 + \mathbf{k}_2) \tag{9.1}$$

where $F(\mathbf{k})$ is as defined in equation 5.1. It is a statistical tool that is useful in searching for non–linear interactions, and it has been shown that this function is superior to the structure factor for differentiating between signals [Heikkilä, 2004]. While equation 9.1 is translationally invariant, in fact it is possible to take the diagonal component

$$B_{diag}(\mathbf{k}) = F(\mathbf{k}) F(\mathbf{k}) F^*(2\mathbf{k}) \tag{9.2}$$

which is also translationally invariant. Expanding this gives

$$B_{diag}(\mathbf{k}) = V^3 \sum_n \sum_m \sum_q \rho'(n) \rho'(m) \rho'(q) \exp(2\pi i \mathbf{k} \cdot \mathbf{r}_n) \exp(2\pi i \mathbf{k} \cdot \mathbf{r}_m) \exp\left(-4\pi i \mathbf{k} \cdot \mathbf{r}_q\right) \tag{9.3}$$

simplifying in terms of $\mathbf{r}_n$, $\mathbf{r}_m$ and $\mathbf{r}_q$ gives

$$B_{diag}(\mathbf{k}) = V^3 \sum_n \sum_m \sum_q \rho'(n) \rho'(m) \rho'(q) \exp\{2\pi i \mathbf{k} \cdot \left[\left(\mathbf{r}_n - \mathbf{r}_q\right) + \left(\mathbf{r}_m - \mathbf{r}_q\right)\right]\} \tag{9.4}$$

The vector $\left[\left(\mathbf{r}_n - \mathbf{r}_q\right) + \left(\mathbf{r}_m - \mathbf{r}_q\right)\right]$ contains two angles to average over and so it is not possible to spherically average equation 9.4 in the same way as equation 5.3. Calculating the average correctly requires a lengthy derivation and leads to an infinite series of spherical Bessel functions and so for this application I will approximate the average as

$$\langle B_{diag}(k') \rangle \approx V^3 \sum_n \sum_m \sum_q \rho'(n) \rho'(m) \rho'(q) \frac{\sin\left(k' \left|\left[\left(\mathbf{r}_n - \mathbf{r}_q\right) + \left(\mathbf{r}_m - \mathbf{r}_q\right)\right]\right|\right)}{k' \left|\left[\left(\mathbf{r}_n - \mathbf{r}_q\right) + \left(\mathbf{r}_m - \mathbf{r}_q\right)\right]\right|} \tag{9.5}$$

There is no requirement on the Bispectrum to be positive–definite, so this function can be either positive or negative. Using this function instead of $\Lambda(k_r)$ in the comparison function, shown in equation 5.14, gives

$$R\left(\langle B_{diag}(k') \rangle\right) = \frac{\Sigma_{k_r} \left|\left|\langle B'_{diag}(k') \rangle\right| - \left|\langle B_{diag}(k') \rangle\right|\right|}{\Sigma_{k_r} \left|\langle B'_{diag}(k') \rangle\right|} \tag{9.6}$$

The use of this function should hopefully improve on the results that have

already been shown using the structure factor as seen in chapter 5. A fitness function using the Bispectrum has yet to be coded into the GA.

9.4 Extension to Surfaces and Interfaces

The determination of surface structure is still one of the most complicated areas of physics. While large-scale *ab initio* calculations have been performed of complicated surfaces [Stich *et al.*, 1992; Brommer *et al.*, 1992], no unbiased global minimisations have been attempted on this scale. Large scale empirical calculations have been performed, with success [Chuang *et al.*, 2005, 2004], but empirical potentials are restrictive, and may not give accurate results, especially for these larger systems.

While the method utilising periodic cuts has been shown during the course of this thesis to be suitable for bulk crystallographic systems, some modification of the method is required to make this method suitable for studies on surface and interface systems. The crossover technique must be modified due to the change in periodicity, as seen in figure 9.1, where in figure 9.1 surface calculations would have species A as a vacuum.

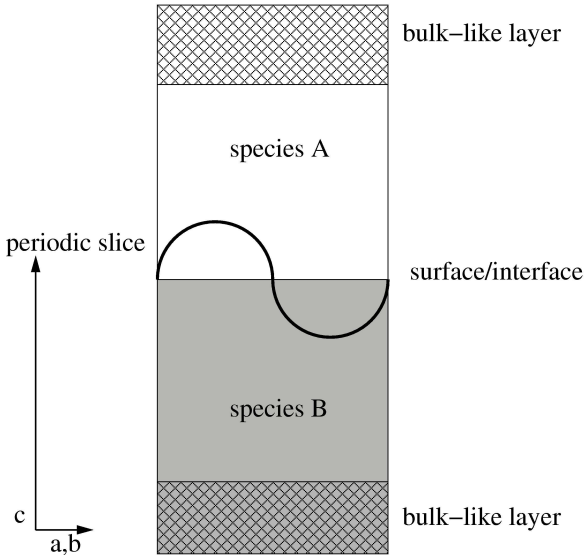

Figure 9.1: Diagram showing the super–cell to be used in surface/interface GA calculations, with periodic cut in the plane of the surface/interface.

Periodic cuts can still be used, but due to the fact that we are dealing with a 2–D system now, rather than the three–dimensional system that the method was originally designed to optimise, only one cut is required. The boundary in the direction perpendicular to the surface/interface will require some form of bulk termination, as is shown in figure 9.1. As was seen in figure 3.5 where the

k–point density and plane–wave cut–off energy had to be converged, in any surface or interface calculations the amount of vacuum–gap and the number of bulk–constrained atoms must also be converged to ensure the accuracy of the calculation.

With regards to the method itself, I believe that the fact that this cut is made in the plane of the surface gives this method advantages over the work of Chuang *et al*. [2005, 2004]. The cut should not only mix elements from both parents in plane, but also allow the addition of adatoms and other surface features, as shown diagrammatically in figure 9.2

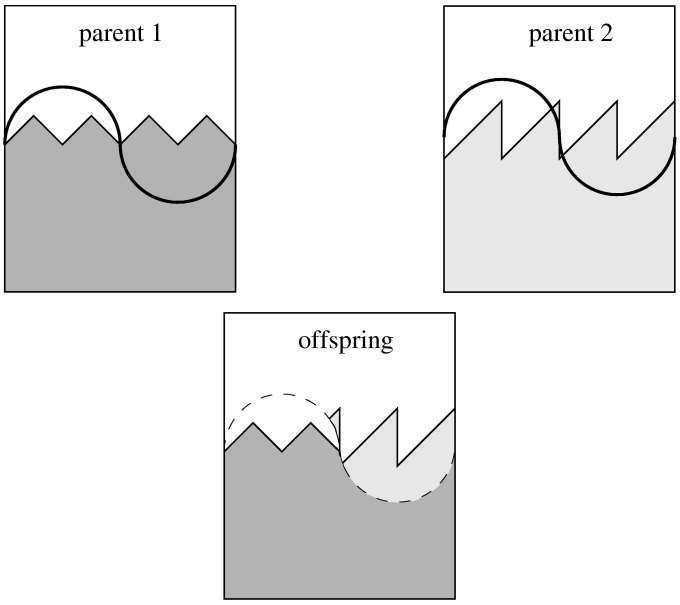

Figure 9.2: Diagram showing crossover between two idealised surfaces with different features. The crossover operation allows the creation of new surface features that may not be present in either parent.

When this method is extended to study interface systems, as figure 9.1 also describes, this method will allow a large degree of mixing in the initial population due to the randomness introduced for the 0th generation. This may have an advantage over proceeding from two idealised surfaces placed in contact and then directly minimising, or simulation the deposition of atoms onto the surface (as in Miyazaki and Inoue [2002]). This may mean that the formation and study of more interesting interface structures may be possible.

As with the extension to multi–species calculations, described in section 9.2, the GA is currently able to perform calculations using this method. However, no results have yet been obtained.

9.5 Extension to Large Systems

While studying large systems using empirical potentials as in chapter 4, 5, and 7 can be useful and does allow some interesting physical insight, it would be better to study all systems using an *ab initio* framework as in chapter 6. However, DFT calculations are incredibly expensive and time–consuming. An ideal solution would be to use an empirical method that was more closely related to the potential energy surface searched in DFT.

To this end I suggest using a combination of Genetic Algorithm methods and Neural Networks. The Neural Network method of Lorenz *et al*. [2006, 2004] would allow large scale calculations to be performed in a feasible timescale.

9.5.1 Neural Networks

It has been recently shown by Lorenz *et al*. [2006, 2004] that Neural Networks (NN) can be trained from *ab initio* calculations to accurately represent the Potential Energy Surface (PES) of the system being studied, and allow Molecular Dynamics (MD) simulations to be performed. In this method an approximation to the PES is found using an iterative technique which samples the *ab initio* PES calculated from Density Functional Theory (DFT). I suggest using these techniques and applying them for use within a GA framework. The NN approach can be easily tested against empirical potentials for improving the method, before being applied to more complicated quantum mechanical systems.

9.5.2 Combining with the Current Method

The NN method could be incorporated into the framework of the Genetic Algorithm, as it has already been shown that the GA makes no assumption as to how the energy of the system is calculated. The NN can be trained from within the course of the calculation, and could be saved and improved upon in subsequent calculations. While a NN PES may not be as accurate as DFT, it would give a good guess at an initial structure that could then be optimised within DFT, training the net as the calculation proceeds.

9.5.3 Other Possibilities

Other possible schemes which would allow the large systems to be studied within an *ab initio* context are the "learn–on–the–fly" method of Csanyi *et al*.

[2004] where the classical force–field is continually updated with *ab initio* calculations. This scheme is currently being incorporated into CASTEP as part of a large MD project. It may also be possible to use a linear–scaling *ab initio* code such as ONETEP [Haynes *et al.*, 2006] instead of CASTEP for the energy and force calculations.

9.6 Extension to Finite Temperature

While there is nothing to prevent the GA method from being used to study systems at finite temperature, the local minimiser would have to be able to search in this space. As was discussed in section 3.4, I was exploring the energy landscape defined by the enthalpy of the system. In a finite temperature situation it is the Gibbs free energy surface that must be explored, $\mathcal{G} = \mathcal{U} + \mathcal{PV} - \mathcal{TS}$. While it is possible to define a temperature, $\mathcal{T}$, it is more difficult to calculate the entropy, $\mathcal{S}$, of the system. For this type of calculation to be possible in a GA framework, it is necessary to be able to perform finite–temperature local–optimisation.

9.7 Conclusions

There are a number of ways in which the genetic algorithm method proposed here could be extended or improved, some of which should be easier to implement than others. Key extensions are those that allow variable–species calculations and surface/interface studies to be performed, and these have already been completed. The Bispectrum method of fitness determination should also be incorporated and test against current results to see if there is an improvement over the Structure Factor method already implemented.

The extensions that allow large–scale or finite temperature calculations will require extensive development and testing before they can be incorporated into the current scheme. However, it is these methods that may allow some interesting and commercially relevant systems to be studied.

Appendix A

Clusters

A.1 Introduction

While the chapters in this thesis have been concerned with bulk systems, I would also like to give the results of a study performed at the start of my PhD on Lennard–Jones clusters, which involved comparing the GA method of Deaven and Ho [1995] and more traditional Simulated Annealing methods. I was interested to see how the "physical" approach of Simulated Annealing compared with the nature–inspired approach of Genetic Algorithms, and hence decide which approach was best for crystal structure determination.

A.2 Simulated Annealing

Simulated Annealing [Kirkpatrick *et al.*, 1983] is a relatively simple technique which is analogous to cooling the structure into the minimum state. The method consists of two loops, the first controls the "temperature", T, and the second is a Metropolis loop [Metropolis *et al.*, 1953] (see algorithm 5). The "temperature" of the system is a parameter that controls how much of the potential energy surface can be explored from any point. The system will start in a random state, and then the atomic positions will be altered at random be a small amount. The energy of the new solution is then compared to the previous solution. The Metropolis loop will always accept lower energy solutions than the current one, but will only accept higher energy solutions with a certain probability that is dependent on this "temperature". The higher the "temperature" then higher energy solutions are more likely to be accepted. The last loop is performed as if $T = 0$ by only accepting lower energy solutions. This is known as a "greedy" step.

This technique can be applied to many problems such as microchip design, and the Travelling Salesman problem [Kirkpatrick *et al.*, 1983]. In microchip design it is the length of the connections between transistors that is optimised, taking into account any congestion on the chip. In the Travelling Salesman problem it is the total distance travelled between cities, where all cities are visited.

A number of different cooling schedules can be chosen, where the temperature is lowered, for example, in a linear or an exponential fashion. I found the linear cooling schedule to be the best in this case.

Algorithm 5 Basic Simulated Annealing

1: Begin with an initial temperature $T = T_i$ (cooling down to a final temperature T_f) and an initial structure with energy E
2: Start temperature loop
3: Start the Metropolis loop for a given number of steps
4: Generate a trial structure with energy E_{trial}
5: Set $\Delta E = E - E_{trial}$
6: **if** $\Delta E > 0$ **then**
7: $P = \exp(-\Delta E / k_B T)$
8: **else**
9: $P = 1$
10: **end if**
11: This value of P is then compared with a random uniform deviate $p \in [0, 1]$
12: **if** $p \leq P$ **then**
13: then the new configuration is accepted
14: **end if**
15: End Metropolis loop
16: The temperature is then lowered by whatever cooling schedule is chosen
17: **if** $T = T_f$ **then**
18: a greedy run through the Metropolis loop is performed
19: then EXIT temperature loop
20: **end if**
21: End temperature loop

A.2.1 Temperature Bouncing

Schneider *et al*. [1998] described the use of a modified cooling schedule known as temperature "bouncing" to improve the results of a simulated annealing approach to the PCB442 problem. At the start of the simulated annealing algorithm, the system is usually in a highly disordered state, but after the initial cooling stage the system is now in a lower–energy one. By increasing the temperature the system can now explore areas of the potential energy surface that were inaccessible at the end of the first quench, but could be lower in energy. Successive raising and lowering of the temperature gives rise to the term "bouncing". This scheme was included so lines $16-21$ of the algorithm are altered from the basic algorithm to those in algorithm 6.

Algorithm 6 Bouncing Alteration

16: The temperature is then lowered by whatever cooling schedule is chosen
17: **if** $T = T_f$ ***and*** $T_i/2 > T_f$ **then**
18: *a greedy run through the Metropolis loop is performed*
19: $T_i = T_i/2$
20: **else**
21: a greedy run through the Metropolis loop is performed
22: then EXIT temperature loop
23: **end if**
24: End temperature loop

A.2.2 Results

It was found that these bouncing steps markedly improved the results for Lennard–Jones clusters (see figure A.1). When the temperature is "bounced" this often results in a new minimum energy structure being found.

Despite the use of bouncing this method was unsuccessful for clusters of more than 12 atoms. The summarised results of performing 128 simulated annealing simulations for each cluster type is shown in figure A.2. These results were quite discouraging and from these results, combined with the success of the GA method (see section A.3.1) I decided to focus wholly on genetic algorithms.

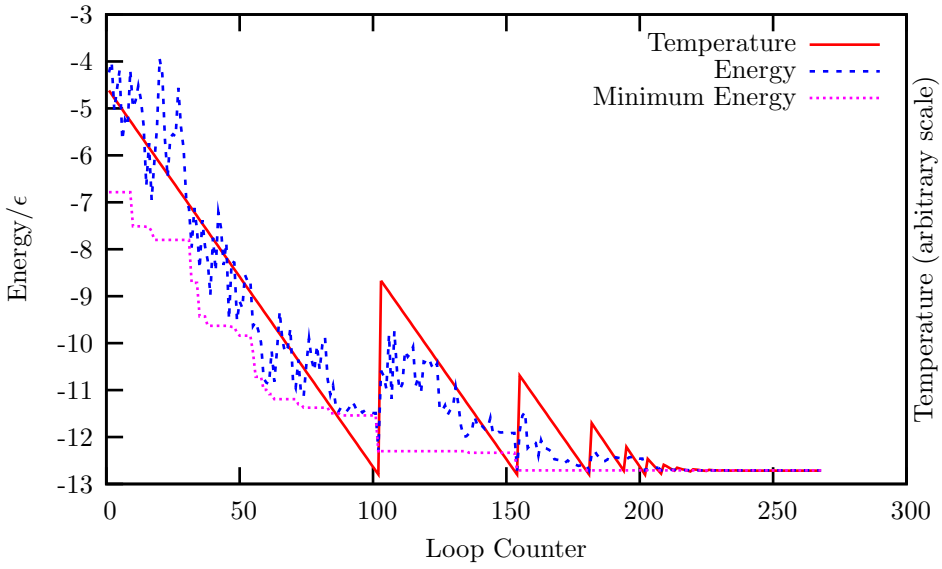

Figure A.1: The effects of temperature bouncing on energy for LJ_6 cluster

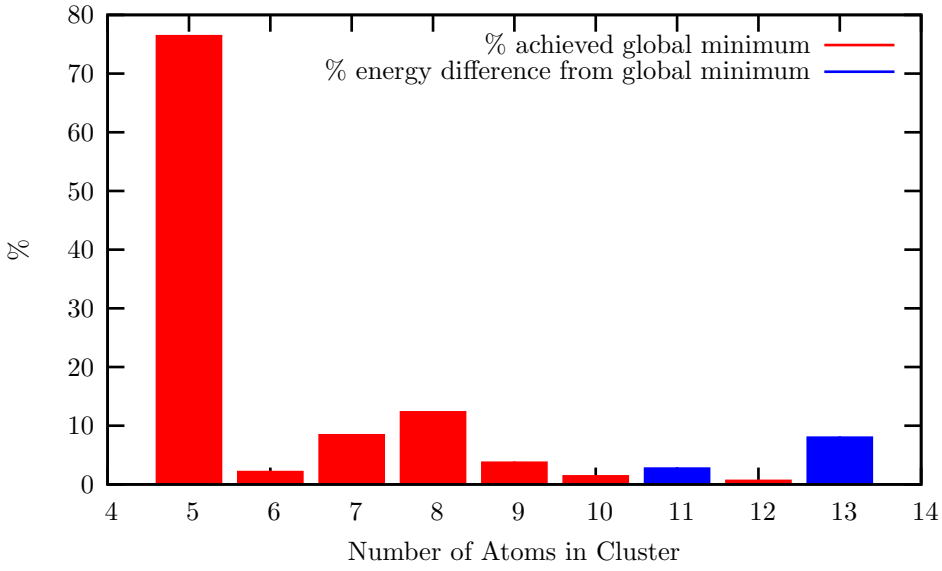

Figure A.2: Summary of Simulated Annealing simulations. There are 128 simulations for each cluster type. Results in red show the percentage of simulations that achieved the global minimum structure, and where this structure was not found, results in blue give the percentage difference of the lowest energy structure found from the global minimum.

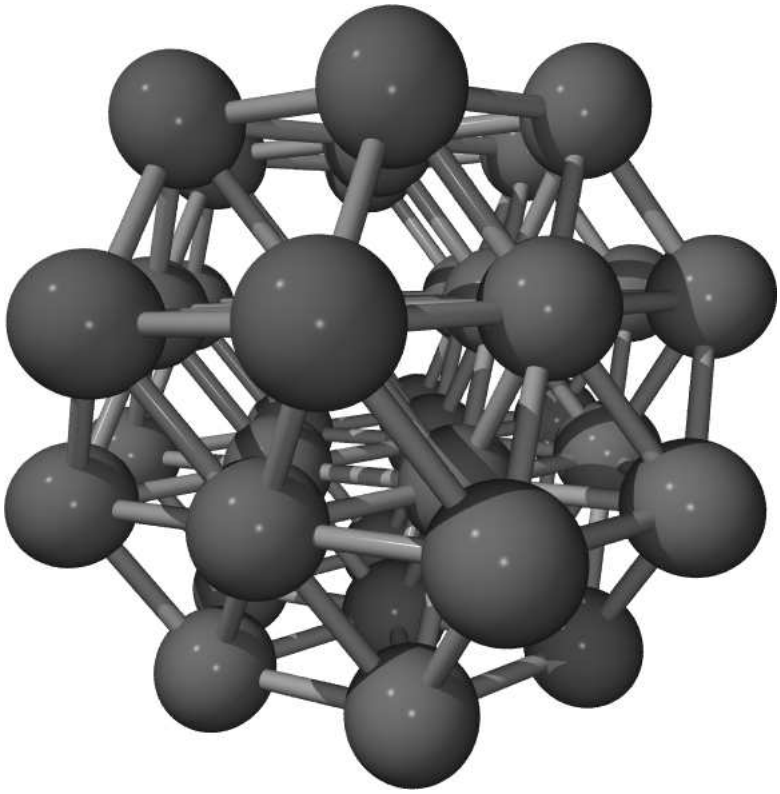

Figure A.3: The 38–atom Lennard–Jones cluster, LJ_{38}

A.3 Genetic Algorithms

The GA method used here is the one described by Deaven and Ho [1995], Johnston [2003] and in section 2.3 except that the plane of the cut is the same for both clusters, and two parents will produce two offspring. This work was performed concurrently with the simulated annealing results described above, and on the strength of the GA method it was decided to only follow this method of optimisation for the rest of this PhD. Update was only performed in an elitist manner however, rather than using a hybrid scheme, which was developed later.

A.3.1 Results

A good test case for any global optimisation technique is to successfully find the LJ_{38} cluster [Doye *et al.*, 1999] shown in figure A.3. This is a close–packed structure, which is different from the icosahedral symmetry of the minimum–energy clusters for most other Lennard–Jones clusters, which are typically based on Mackay icosahedra [Mackay, 1962].

This method had no problems in being able to find this structure, as can be

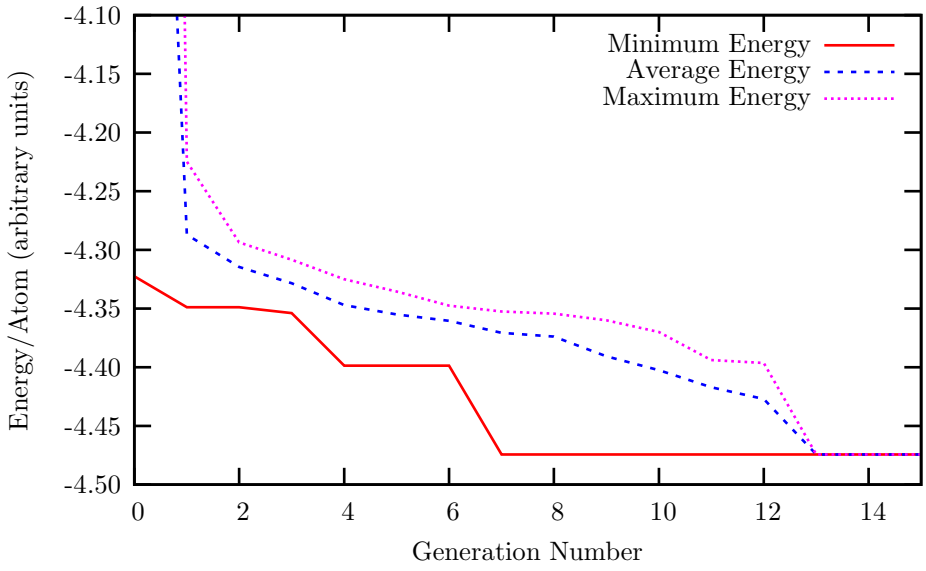

Figure A.4: Results from GA in the study of the LJ_{38} cluster. The minimum–energy configuration was found in 7 generations.

seen in figure A.4. The structure shown in figure A.3 was found after only 7 generations, with the all the structures becoming the same after 13 generations.

A.4 Conclusions

When comparing the methods of simulated annealing and genetic algorithms it is clear to see, from the results presented here, that GAs are more efficient than simulated annealing for determining the minimum energy configurations of atomic clusters. Having discarded simulated annealing as a method, I then moved on to expanding the GA cluster technique so that it would be suitable for studying systems with greater periodicity. The method developed has been described in detail in chapter 4 and extensions and improvements to this method are then described in subsequent chapters.

Appendix B

Recipe for Cinnamon Balls

Ingredient	Amount
egg whites	2
Castor (superfine) sugar	100 g (4 oz / $\frac{1}{2}$ cup)
ground almonds	200 g (1/2 lb / 2 cups)
cinnamon	1 level tablespoon
icing (confectioners') sugar	(approx 5mm deep in a plate or wide bowl)

Algorithm 7 The Perfect Cinnamon Balls [Rose, 2004]

1: Beat the egg whites till they form stiff peaks.
2: Fold in all the remaining ingredients.
3: Form into balls with wetted hands.
4: Bake on a greased tray at 170° C (Gas Mark 3 / 325° F) for 25 minutes, or until just firm to the touch.
5: Roll in icing sugar whilst warm.
6: Roll in icing sugar when cold.

I find it easier to mix the dry ingredients first, before adding them to the egg whites. This ensures a more even mixing.

It is important to bake the balls only as long as directed to ensure that the biscuits remain soft and moist inside. It may seem that they are still underdone, but it is important that they are not allowed to dry out.

These amounts make about 15-20 depending on the size of the cinnamon balls. I find it best *not* to pre–heat the oven otherwise they may burn. Also use a clean baking sheet, or one that has only been used for cakes/biscuits, to improve taste. Remove them from the baking sheet with a firm twist, or a thin spatula.

Bibliography

Accelrys, 2001–. URL `http://www.accelrys.com/`.

M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 1987. ISBN 0-19-855375-7.

V. E. Bazterra, M. B. Ferraro, and J. C. Facelli. Modified genetic algorithm to model crystal structures: Iii. determination of crystal structures allowing simultaneous molecular geometry relaxation. *International J. Quantum Chem.*, 96:312–320, 2004.

R. L Becerra and C. A. C. Coelloa. Cultured differential evolution for constrained optimization. *Comput. Methods Appl. Mech. Engrg.*, 195:43034322, 2006.

V Blum, G. L. W. Hart, M. J. Walorski, and A. Zunger. Using genetic algorithms to map first-principles results to model Hamiltonians: Application to the generalized Ising model for alloys. *Phys. Rev. B*, 72:165113, 2005.

M. Born and J. R. Oppenheimer. Zur quantentheorie der molekeln. *Ann. Phys. (Leipzig)*, 84:457–484, 1927.

W. H. Bragg and W. L. Bragg. *Nature*, 91:557, 1913a.

W. H. Bragg and W. L. Bragg. *Proc. Royal Soc.*, A89:277, 1913b.

T. Brodmeier and E. Pretsch. Application of genetic algorithms in molecular modeling. *J. Comput. Chem.*, 15:588–595, 1994.

K. D. Brommer, M. Needels, B. E. Larson, and J. D. Joannopoulos. Abinitio theory of the si(111)-(7x7) surface reconstruction - a challenge for massively parallel computation. *Phys. Review Letters*, 68:1355–1358, 1992.

M. J. Bucknum, I. Stamatin, and E. A. Castro. A chemically intuitive proposal for the structure of n–diamond. *Molecular Physics*, 103:2707–2715, 2005.

D. M. Ceperley and B. J. Alder. Ground state of the electron gas by a stochastic method. *Phys. Rev. Lett.*, 45:566, 1980.

E. Y. Cheung, E. E. Mccabe, K. D. M. Harris, R. L. Johnston, E. Tedesco, K. M. P. Raja, and P. Balaram. C-h center dot center dot center dot o hydrogen bond mediated chain reversal in a peptide containing a gamma-amino acid residue, determined directly from powder x-ray diffraction data. *Angewandte Chemie-International Edition*, 41:494–496, 2002.

S. Y. Chong and M. Tremayne. Combined optimization using cultural and differential evolution: application to crystal structure solution from powder diffraction data. *Chem. Commun.*, pages 4078 – 4080, 2006.

F. C. Chuang, C. V. Ciobanu, V. B. Shenoy, C. Z. Wang, and K. M. Ho. Finding the reconstructions of semiconductor surfaces via a genetic algorithm. *Surf. Sci.*, 573:L375, 2004.

F. C. Chuang, C. V. Ciobanu, C. Predescu, C. Z. Wang, and K. M. Ho. Structure of Si(114) determined by global optimization methods. *Surf. Sci.*, 578:183, 2005.

Crystal Lattice Structures, 1995–. URL `http://cst-www.nrl.navy.mil/lattice/`.

Gabor Csanyi, T. Albaret, M. C. Payne, and A. De Vita. "learn on the fly": A hybrid classical and quantum-mechanical molecular dynamics simulation. *Physical Review Letters*, 93(17):175503, 2004. URL `http://link.aps.org/abstract/PRL/v93/e175503`.

D. de Fontaine. Cluster approach to order–disorder transformations in alloys. *Solid State Phys.*, 47:33–176, 1994.

D. M. Deaven and K. M. Ho. Molecular-geometry optimization with a genetic algorithm. *Phys. Rev. Lett.*, 75:288–291, 1995.

D. M. Deaven, N. Tit, J. R. Morris, and K. M. Ho. Structural optimization of Lennard–Jones clusters by a genetic algorithm. *Chem. Phys. Lett.*, 256:195–200, 1996.

P. Debye. X-ray dispersal. *Annalen Der Physik*, 46:809–823, 1915.

R. Doll and M. A. Vanhove. Global optimization in leed structure determination using genetic algorithms. *Surface Science*, 355:L393–L398, 1996.

J. Donohue. *The Structures of the Elements*. John Wiley & Sons, 1974. ISBN 0-471-21788-3.

J. P. K. Doye and D. J. Wales. Surveying a potential energy surface by eigenvector–following. *Z. Phys. D.*, 40:194–197, 1997.

J. P. K. Doye, M. A. Miller, and D. J. Wales. The double-funnel energy landscape of the 38-atom Lennard- Jones cluster. *J. Chem. Phys.*, 110:6896–6906, 1999.

J. P. K. Doye, D. J. Wales, F. H. M. Zetterling, and M. Dzugutov. The favored cluster structures of model glass formers. *J. Chem. Phys.*, 118:2792–2799, 2003.

M. Dzugutov. Glass-formation in a simple monatomic liquid with icosahedral inherent local order. *Phys. Rev. A*, 46:R2984–R2987, 1992.

Mikhail Dzugutov. Formation of a dodecagonal quasicrystalline phase in a simple monatomic liquid. *Phys. Rev. Lett.*, 70(19):2924–2927, May 1993. doi: 10.1103/PhysRevLett.70.2924.

S. Ergun and L. E. Alexander. Crystallined forms of carbon: A possible hexagonal polymorph of diamond. *Nature*, 195:765, 1962.

R. P. Feynman. Forces in molecules. *Physical Review*, 56:340–343, 1939.

W. M. C. Foulkes, L. Mitas, R. J. Needs, and G. Rajagopal. Quantum monte carlo simulations of solids. *Rev. Mod. Phys.*, 73(1):33–83, Jan 2001. doi: 10.1103/RevModPhys.73.33.

R. T. Fu, K. Esfarjani, Y. Hashi, J. Wu, X. Sun, and Y. Kawazoe. Surface reconstruction of si (001) by genetic algorithm and simulated annealing method. *Sci. Rep. Res. Inst. Tohoku Univ. Ser. A-Phys. Chem. Metall.*, 44:77–81, 1997.

S Goedecker. Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. *J. Chem. Phys*, 120:9911, 2004.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison–Wesley Publishing Co. Inc., 1989. ISBN 0-201-15767-5.

K. D. M. Harris, R. L. Johnston, and B. M. Kariuki. The genetic algorithm: Foundations and applications in structure solution from powder diffraction data. *Acta Crystallographica Section A*, 54:632–645, 1998a.

K. D. M. Harris, R. L. Johnston, B. M. Kariuki, and M. Tremayne. A genetic algorithm for crystal structure solution from powder diffraction data. *J. Chem. Research-S*, pages 390–391, 1998b.

K. D. M. Harris, S. Habershon, E. Y. Cheung, and R. L. Johnston. Developments in genetic algorithm techniques for structure solution from powder diffraction data. *Z. Kristallogr.*, 219:838–846, 2004.

G. L. W. Hart, V. Blum, M. J. Walorski, and A. Zunger. Evolutionary approach for determining first-principles Hamiltonians. *Nature Materials*, 4:391, 2005.

B. Hartke. Siz–dependent transition from all–surface to interior–molecule structures in pure neutral water clusters. *Phys. Chem. Chem. Phys.*, 5:275–284, 2002.

B. Hartke. Global geometry optimization of clusters using genetic algorithms. *J. Phys. Chem.*, 97:9973–9976, 1993.

B. Hartke. Global geometry optimization of clusters guided by N–dependent model potentials. *Chem. Phys. Lett.*, 258:144–148, 1996.

B. Hartke. Global cluster geometry optimization by a phenotype algorithm with niches: Location of elusive minima, and low–order scaling with cluster size. *J. Comput. Chem.*, 20:1752–1759, 1999.

B. Hartke, H. J. Flad, and Dolg. M. Structures of mercury clusters in a quantum–empirical hybrid model. *Phys. Chem. Chem. Phys.*, 3:5121–5129, 2001.

I. Harvey. *The SAGA Cross: The mechanics of recombination for species with variable–length genotypes.*, volume 2 of *Parallel problem solving from nature*, pages 269–278. 1992.

B. Hayes. How to count. *American Scientist*, 89:110–114, 2001.

P. D. Haynes, C. K. Skylaris, A. A. Mostofi, and M. C. Payne. Onetep: linear-scaling density-functional theory with local orbitals and plane waves. *Physica Status Solidi*, 243:2489–2499, 2006.

J. Heikkilä. A new class of shift-invariant operators. *IEEE Signal Processing Letters*, 11:545–548, 2004.

M. R. Hoare and P. Pal. Physical cluster mechanics: statics and energy surfaces for monatonic systems. *Adv. Phys.*, 20:161–196, 1971a.

M. R. Hoare and P. Pal. Statics and stability of small cluster nuclei. *Nature*, 230: 5–8, 1971b.

M. R. Hoare and P. Pal. Geometry and stability of "spherical" f.c.c. microcrystallites. *Nature*, 236:35–37, 1972.

R. Hoffmann, T. Hughbanks, and M. Kertśz. A hypothetical metallic allotrope of carbon. *J. Am. Chem. Soc.*, 105:4831–4832, 1983.

P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136: B863–B871, 1964.

John H. Holland. *Adaption in Natural and Artificial Systems*. MIT Press/Bradford Books Edition, 1992. ISBN 0-262-58111-6.

J. D. Honeycutt and H. C. Andersen. Molecular-dynamics study of melting and freezing of small lennard-jones clusters. *J. Phys. Chem.*, 91:4950–4963, 1987.

G. H. Johannesson, T. Bligaard, A. V. Ruban, H. L. Skriver, K. W. Jacobsen, and J. K. Norskov. Combined electronic structure and evolutionary search approach to materials design. *Phys. Review Letters*, 88:10089–10095, 2002.

R. L. Johnston. Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries. *Dalton Trans.*, pages 4193–4207, 2003.

W Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32:922, 1976.

W Kabsch. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A34:827–828, 1978.

G. Kane and M. Goeppert-Mayer. Lattice summations for hexagonal close-packed crystals. *J. Chem. Phys*, 8:642, 1940.

B. M. Kariuki, H. Serrano-Gonzalez, R. L. Johnston, and K. D. M. Harris. The application of a genetic algorithm for solving crystal structures from powder diffraction data. *Chem. Phys. Letters*, 280:189–195, 1997.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

W. Kohn and L. J. Sham. Sef–consistent equations including exchange and correlation effects. *Phys. Rev.*, 140:A1133–A1138, 1965.

R. J. Lagow, J. J. Kampa, H. C. Wei, S. L. Battle, J. W. Genge, D. A. Laude, C. J. Harper, R. Bau, R. C. Stevens, J. F. Haw, and E. Munson. Synthesis of linear acetylenic carbon - the sp carbon allotrope. *Science*, 267:362–367, 1995.

A. Laio and M. Parrinello. Escaping free-energy minima. *Proc. Natl. Acad. Sci. U.S.A.*, 99:12562–12566, 2002.

A. Laio, A. Rodriguez-Fortea, F. L. Gervasio, M. Ceccarelli, and M. Parrinello. Assessing the accuracy of metadynamics. *J. Phys. Chem. B*, 109:6714–6721, 2005.

E. Landree, C. Collazo-Davila, and L. D. Marks. Multi-solution genetic algorithm approach to surface structure determination using direct methods. *Acta Crystallographica Section B-Structural Science*, 53:916–922, 1997.

J. E. Lennard-Jones and A. E. Ingham. On the calculations of certain crystal potential constants, and on the cubic crystal of least potential energy. *Proc. Royal Soc.*, A107:636, 1925.

S. Lorenz, A. Gross, and M. Scheffler. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Letters*, 395:210–215, 2004.

S. Lorenz, M. Scheffler, and A. Gross. Descriptions of surface chemical reactions using a neural network representation of the potential-energy surface. *Phys. Review B*, 73:115431, 2006.

A. L. Mackay. A dense non-crystallographic packing of equal spheres. *Acta Cryst.*, 15:916, 1962.

J. Maddox. Crystals from 1st principles. *Nature*, 335:201, 1988.

Richard M. Martin. *Electronic Structure: basic theory and practical methods.* Cambridge University Press, 2004. ISBN 0-521-78285-6.

D. B. McGarrah and R. S. Judson. Analysis of the genetic algorithm method of molecular conformation determination. *J. Comput. Chem.*, 14:1385–1395, 1993.

J. M. Mendel. Tutorial on higher-order statistics (spectra) in signal processing and system theory: theoretical results and some applications. *Proc. IEEE*, 79: 278–305, 1991.

N Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087, 1953.

Koji Miyazaki and Takayoshi Inoue. Genetic algorithm simulation for the deposited structure of atoms. *Surf. Sci.*, 501:93–101, 2002.

H. J. Monkhorst and J. D. Pack. Special points for Brillouin-zone integrations. *Phys. Rev. B*, 13:5188, 1976.

J. A. Niesse and H. R. Mayne. Global geometry optimization of atomic clusters using a modified genetic algorithm in space-fixed coordinates. *J. Chem. Phys.*, 105:4700–4706, 1996.

J. A. Niesse and H. R. Mayne. Global optimization of atomic and molecular clusters using the space-fixed modified genetic algorithm method. *J. Comput. Chem.*, 18:1233–1244, 1997.

A. R. Oganov and C. W. Glass. Crystal structure prediction using ab initio evolutionary techniques: Principles and applications. *J. Chem. Phys.*, 124: 175–185, 2006.

A. R. Oganov, C. W. Glass, and S. Ono. High-pressure phases of caco3: Crystal structure prediction and experiment. *Earth Planetary Science Letters*, 241:95–103, 2006.

J. D. Pack and H. J. Monkhorst. "special points for Brillouin-zone integrations"-a reply. *Phys. Rev. B*, 16:1748, 1977.

J. P. Perdew and Alex Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B*, 23(10):5048–5079, May 1981. doi: 10.1103/PhysRevB.23.5048.

John P. Perdew, J. A. Chevary, S. H. Vosko, Koblar A. Jackson, Mark R. Pederson, D. J. Singh, and Carlos Fiolhais. Atoms, molecules, solids, and surfaces: Applications of the generalized gradient approximation for exchange and correlation. *Phys. Rev. B*, 46(11):6671–6687, Sep 1992. doi: 10.1103/PhysRevB.46.6671.

B. G. Pfrommer, M. Cote, S. G. Louie, and M. L. Cohen. Relaxation of crystals with the quasi-newton method. *J. Comp. Phys.*, 131:233, 1997.

C. J. Pickard and R. J. Needs. High-pressure phases of silane. *Phys. Review Letters*, 97:045504, 2006.

E. Polak. *Computational Methods in Optimization*, volume 77. Mathematics in Science and Engineering, 1971.

G. L. Pollack. The solid state of rare gases. *Rev. Mod. Phys.*, 36:748, 1964a.

Gerald L. Pollack. The solid state of rare gases. *Rev. Mod. Phys.*, 36(3):748–791, Jul 1964b. doi: 10.1103/RevModPhys.36.748.

W. J. Pullan. Genetic operators for the atomic cluster problem. *Comp. Phys. Comm.*, 107:137–148, 1997.

D. Quigley and M. I. J. Probert. Phase behavior of a three-dimensional core-softened model system. *Phys. Review E*, 71:322–325, 2005a.

D. Quigley and M. I. J. Probert. Progression of phase behavior for a sequence of model core- softened potentials. *Phys. Review E*, 72:061202, 2005b.

Evelyn Rose. *The New Complete International Jewish Cookbook*, page 617. Robson Books Ltd, 2004. ISBN 1-86105-732-6.

J. Roth and A. R. Denton. Solid-phase structures of the dzugutov pair potential. *Phys. Rev. E*, 61:6845–6857, 2000.

A. Scemama, P. Chaquin, M. C. Gazeau, and Y. Benilan. Semi-empirical calculation of electronic absorption wavelengths of polyynes, monocyano- and dicyanopolyynes. predictions for long chain compounds and carbon allotrope carbyne. *Chem. Phys. Letters*, 361:520–524, 2002.

J. Schneider, I. Morgenstern, and J. M. Singer. Bouncing towards the optimum: Improving the results of monte carlo optimization algorithms. *Phys. Rev. E*, 58:5085–5095, 1998.

Peter A. Schultz, Kevin Leung, and E. B. Stechel. Small rings and amorphous tetrahedral carbon. *Phys. Rev. B*, 59(2):733–741, Jan 1999. doi: 10.1103/PhysRevB.59.733.

M. D. Segall, P. L. D. Lindan, M. J. Probert, C. J. Pickard, P. J. Hasnip, S. J. Clark, and M. C. Payne. First-principles simulation: Ideas, illustrations and the CASTEP code. *J. Phys.: Cond. Matt.*, 14(11):2717–2743, 2002.

David P. Shoemaker and Clara B. Shoemaker. Comment on ''filling three-dimensional space with tetrahedra: A geometric and crystallographic problem''. *Phys. Rev. B*, 38(9):6319–6321, Sep 1988. doi: 10.1103/PhysRevB.38.6319.

R. W. Smith. Energy minimization in binary alloy models via genetic algorithms. *Computer Phys. Comm.*, 71:134–146, 1992.

R. Srinivasan and S. Parthasarathy. *Some statistical applications in X-ray crystallography*. Pergamon Press, 1976. ISBN 0080180469.

I. Stich, M. C. Payne, R. D. Kingsmith, J. S. Lin, and L. J. Clarke. Abinitio total-energy calculations for extremely large systems - application to the takayanagi reconstruction of si(111). *Phys. Rev. Lett.*, 68:1351–1354, 1992.

S. D. Stoddard and J. Ford. Numerical experiments on stochastic behavior of a Lennard-Jones gas system. *Phys. Rev. A*, 8:1504, 1973.

H. T. Stokes and D. M. Hatch, 1995–
. URL `http://stokes.byu.edu/isotropy.html`.

R. T. Strong, C. J. Pickard, V. Milman, G. Thimm, and B. Winkler. Systematic prediction of crystal structures: An application to sp(3)-hybridized carbon polymorphs. *Phys. Rev. B*, 70:045101, 2004.

Z. H. Sun, Q. W. Liu, Y. F. Li, and J. Zhuang. Structures of adatom clusters on ag(iii) surface by genetic algorithm. *Chinese Phys. Letters*, 21:1604–1607, 2004.

K. Y. Szeto and J. Villain. Filling three-dimensional space with tetrahedra: A geometric and crystallographic problem. *Phys. Rev. B*, 36(9):4715–4724, Sep 1987. doi: 10.1103/PhysRevB.36.4715.

E. Tedesco, K. D. M. Harris, R. L. Johnston, G. W. Turner, K. M. P. Raja, and P. Balaram. Ab initio structure determination of a peptide beta-turn from powder x-ray diffraction data (august, pg 1460, 2001). *Chem. Comm.*, pages 494–496, 2001.

C. Tsoo and C. L. Brooks. Cluster structure determination using gaussian density distribution global minimization methods. *J. Chem. Phys.*, 101:6405–6411, 1994.

G. W. Turner, E. Tedesco, K. D. M. Harris, R. L. Johnston, and B. M. Kariuki. Implementation of lamarckian concepts in a genetic algorithm for structure solution from powder diffraction data. *Chem. Phys. Letters*, 321:183–190, 2000.

D. J. Wales. *Energy Landscapes*. Cambridge University Press, 2003. ISBN 0-521-81415-4.

D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *J. Phys. Chem. A*, 101:5111–5116, 1997.

D. J. Wales and H. A Scheraga. Global optimisation of clusters, crystals and biomolecules. *Science*, 285:1368–1372, 1999.

B. Wang, S. Yin, G. Wang, and J. Zhao. Structure and electronic properties of ultrathin titanium nanowires. *J. Phys.: Cond. Mat.*, 13:L403–L408, 2001a.

B. Wang, G. Wang, and J. Zhao. Magic structures of helical multishell zirconium nanowires. *Phys. Rev. B*, 65:235406, 2002.

B. L. Wang, S. Y. Yin, G. H. Wang, A. Buldum, and J. J. Zhao. Novel structures and properties of gold nanowires. *Phys. Rev. Lett.*, 86:2046, 2001b.

J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *J. Chem. Phys.*, 54: 5237–5247, 1971.

L. T. Wille. Minimum-energy configurations of atomic clusters - new results obtained by simulated annealing. *Chem. Phys. Lett.*, 133:405–410, 1987.

B. Winkler, C. J. Pickard, V. Milman, W. E. Klee, and G. Thimm. Prediction of a nanoporous sp(2)-carbon framework structure by combining graph theory with quantum mechanics. *Chem. Phys. Lett.*, 312:536, 1999.

B. Winkler, C. J. Pickard, V. Milman, and G. Thimm. Systematic prediction of crystal structures. *Chem. Phys. Lett.*, 337:36, 2001.

M. D. Wolf and U. Landman. Genetic algorithms for structural cluster optimization. *J. Phys. Chem. A*, 102:6129–6137, 1998.

S. M. Woodley. Prediction of crystal structures using evolutionary algorithms and related techniques. *Struct. Bonding (Berlin)*, 110:95, 2004a.

S. M. Woodley. Prediction of crystal structures using evolutionary algorithms and related techniques. *Applications Evolutionary Computation Chem.*, 110:95–132, 2004b.

S. M. Woodley. Prediction of inorganic crystal framework structures - part 2 - using a genetic algorithm and a direct approach to exclusion zones. *Phys. Chem. Chemical Physics*, 6:1823–1829, 2004c.

S. M. Woodley, P. D. Battle, J. D. Gale, and C. R. A. Catlow. The prediction of inorganic crystal structures using a genetic algorithm and energy minimisation. *Phys. Chem. Chemical Physics*, 1:2535–2542, 1999.

S. M. Woodley, P. D. Battle, J. D. Gale, and C. R. A. Catlow. Prediction of inorganic crystal framework structures - part 1: Using a genetic algorithm and an indirect approach to exclusion zones. *Phys. Chem. Chemical Physics*, 6: 1815–1822, 2004a.

S. M. Woodley, C. R. A. Catlow, P. D. Battle, and J. D. Gale. The prediction of inorganic crystal framework structures using excluded regions within a genetic algorithm approach. *Chem. Comm.*, pages 22–23, 2004b.

Y. Xiang, H. Jiang, W. Cai, and X. Shao. An efficient method based on lattice construction and the genetic algorithm for optmization of large Lennard–Jones clusters. *J. Phys. Chem. A*, 108:3586–3592, 2004.

Y. Xiao and D. E. Williams. Genetic algorithm: a new approach to the prediction of the structure of molecular clusters. *Chem. Phys. Lett.*, 215:17–24, 1993.

H. J. W. Zandvliet. Genetic algorithm for finding the reconstruction of semiconductor surfaces. *Surface Science*, 577:93–94, 2005.

Y. Zeiri. Prediction of the lowest energy structure of clusters using a genetic algorithm. *Phys. Review E*, 51:R2769–R2772, 1995.