# Modeling Metal Protein Complexes from Experimental Extended X-ray Absorption Fine-Structure using Computational Intelligence

**Collin Price**

Department of Computer Science

Submitted in partial fulfillment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The aim of this thesis is to find a better method for determining the atomic structure of a molecule using Extended X-Ray Absorption Fine Structure (EXAFS). The thesis uses the oxygen-evolving complex (OEC) in state $S_1$ as an example for structure refinement but the developed process can be applied to any given chemical structure that has undergone x-ray absorption spectroscopy experimentation. In this chapter, we introduce the biological background and terms, followed by the problem definition, and finally elaborate on the computer science theories applied to the problem.

## 1.1 Biological Background

Photosystem II [1] is the protein complex responsible for the first stage of photosynthesis. Photosynthesis is a process used by plants and other organisms to convert light (photons) into energy. Photons, that are captured from the Sun or other light sources, and water are processed through a water-oxidizing enzyme known as the oxygen-evolving complex (OEC) [2]. The water molecule ($H_2O$) is split into two parts, $O_2$ and $H^+$. The $O_2$ is released from the system, and the $H^+$ will be stored and used as a source of energy.

The OEC complex performs oxidation on two water molecules through a series of intermediary states. The "S-State Cycle" [2] consists of 5 states: $S_0$, $S_1$, $S_2$, $S_3$, and $S_4$. During the transistion between each state a hydrogen electron is released.

After $S_4$ concludes $O_2$ is formed. For the purpose of this work the resting or so-called storage state $S_1$ will be analysed. The atomic structure of the OEC molecule is altered between each state.

The most significant feature of this compound is its inorganic core $Mn_4Ca_1O_xCl_{1-2}(HCO_3)_y$. It is not found anywhere else in biology and offers the only biological blueprint for water splitting. By studying OEC the hope is to understand how the oxidation of water can occur at such a low energy cost. Aquiring a better understanding of how the water splitting process occurs will assist in creating biomimetic catalysts or engineered PSII enzymes for real world applications.

## 1.2 X-ray Absorption Spectroscopy

The following overview is based on information contained in Matthew Newvilles Fundamentals of XAFS (2004) [3]. X-Ray absorption fine structure (XAFS) is a method used to measure the absorption coefficient of a material as a function of energy. X-rays are part of the electromagnetic spectrum with wavelengths ranging from 25Å to 0.25Å. All atoms resonate at a specific wavelength. The x-ray is tuned to have the same wavelength as the target atom. A photon from an x-ray is absorbed by an electron in a tightly bound quantum core level of an atom. Absorption only takes place if the binding energy of the core level is less than the energy of the x-ray photon. At the time of absorption a core electron moves to an empty outer shell and another electron moves in to take its place. Eventually the affected electrons decay to their original state. During this time fluorescence energies are emitted that characterize a specific atom.

The absorption coefficients measured after the initial absorption are referred to as the EXAFS. During the decay of the electrons to their original state, oscillations occur in the measure of the absorption coefficient. The different frequencies found within the oscillations correspond to different near-neighbour coordination shells, which can be described and modeled according to the EXAFS equation. From the oscillations, the number of neighbouring atoms, the distances to the neighbouring atoms, and the disorder in the neighbour distances can be determined. The energy spectra for OEC in $S_1$ is shown in Figure 1.1.

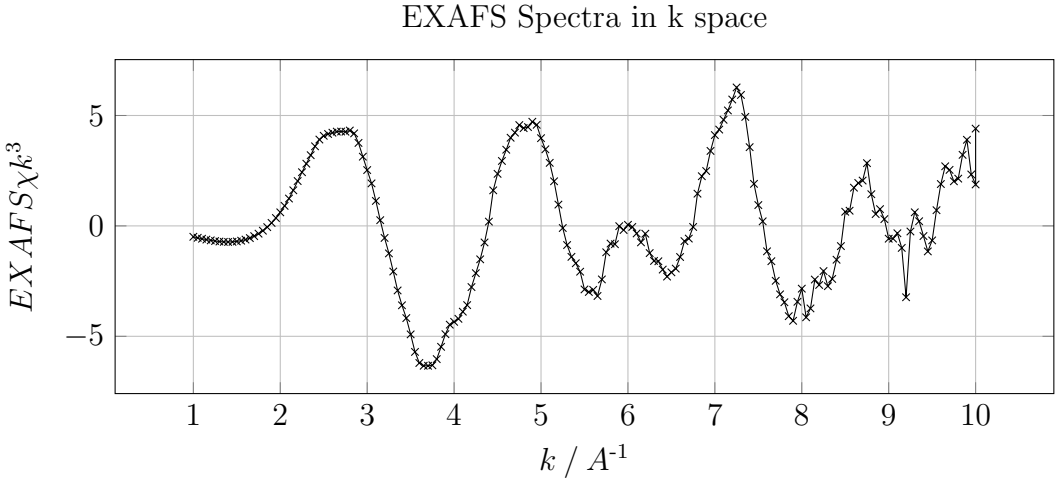


Figure 1.1: EXAFS Spectra of OEC in $S_1$

## 1.3 Force Fields

The atoms within a molecule are consistently interacting with each other. Atoms can directly and indirectly interact with neighbouring atoms. Atoms directly interact with neighbouring atoms with a bond or indirectly through van der Waals forces. Calculating the forces involved within the molecule would require a large amount of computing power to attain a high degree of accuracy. Instead simplier, classical formulas are used to calculate the energy within the system. There are several different formulas for calculating classical force fields. This work will utilize Assisted Model Building with Energy Refinement (AMBER) [4] force fields for the energy calculations. AMBER force fields are widely used with proteins and related systems [5]. Equation 1.1 shows the formula used when calcuting the energy of a system using AMBER force fields.

$$\begin{aligned} V(r^N) = & \sum_{\text{bonds}} k_b(l-l_0)^2 \\ & + \sum_{\text{angles}} k_a(\theta-\theta_0)^2 \\ & + \sum_{\text{torsions}} \sum_{n} \frac{1}{2} V_n[1+\cos(n\omega-\gamma)] \\ & + \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} f_{ij} \left\{ \epsilon_{ij} \left[ \left( \frac{r_{0ij}}{r_{ij}} \right)^{12} - 2 \left( \frac{r_{0ij}}{r_{ij}} \right)^{6} \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\} \end{aligned} \tag{1.1}$$

## 1.4 Problem Definition

The goal of this thesis is to examine different search heuristics to determine the best method of finding the theoretical atomic structure of a molecule using the molecule's EXAFS spectrum for comparison. This problem contains two important but unrelated goals. Firstly, the algorithm must be able to find an atomic structure whose EXAFS spectrum matches the experimental EXAFS spectrum, and secondly, creating an atomic structure whose potential energy is a low as possible.

EXAFS can be used to identify properties of a molecule, but they do not provide enough detail to determine the atomic structure of a molecule in 3-dimensional space. An EXAFS spectrum allows you to identify how far apart atoms are from each other, but does not give enough information to identify their dihedral angles. Fortunately, EXAFS can be used to assist in determining the atomic structure of a molecule. The energy spectrum given off by the molecule is unique to its structure, meaning that you can create an atomic structure, obtain its EXAFS spectrum, and compare the results. The hope is that if you create an atomic structure whose EXAFS spectrum closely matches the EXAFS spectrum of an actual model, then there is a high likelihood that the created structure will closely match the actual structure.

Using EXAFS spectrum comparison the goal is to obtain a set of candidate atomic structures. Atomic structures that generate similar EXAFS spectra may have different geometries. An expert will have to analyse the candidate solutions to determine if any of these atomic structures are actually chemically infeasible. **Having a set of candidate solutions will improve the odds of finding the actual solution.**

The IFEFFIT XAFS data analysis suite [6] is used to simulate the EXAFS experiments. This suite includes two applications that will be used FEFF6, and IFEFFIT. FEFF6 is used to simulate an XAFS experiment and IFEFFIT does post processing of the simulated EXAFS spectra. During the atomic structure refinement, the generated atomic structures will be run through these applications to obtain an EXAFS spectra.

NAMD [7] will be used for the energy calculations. The NAMD Energy Plugin [8] will calculate the potential energy of the generated atomic structure.

## 1.5 Thesis Organization

The remainder of this thesis is organized as follows.

# Chapter 2

# Background

The purpose of this chapter is to assist the reader in understanding the search techniques used in this work. Section 2.1 will explain the framework used for the algorithms described in Sections 2.2, 2.3, 2.4, and 2.5.

## 2.1 Evolutionary Algorithms

An evolutionary algorithm (EA) is a population-based metaheuristic optimization algorithm. An evolutionary algorithm is a search heuristic that is based on Darwin's theory of natural evolution. Darwin theorized that over a period of time a population of individuals would naturally mate and create offspring that were better than themselves. He suggested that not all individuals are created equally and that eventually the weaker individuals would die off. This same principle can be applied to a search algorithm as a heuristic. An EA contains a population of individuals that are evolved to find improved candidate solutions.

Figure 2.1 demonstrates how the basic EA operates. Initially a *population* of candidate solutions is generated. The individuals are evaluated based on an *evaluation function* and are checked against the *stopping criterion*. If the *stopping criterion* has not been reached the population goes through an *evolutionary period* where a new population of candidate individuals are created from the last population. This iterative process, also called a generation, is repeated until the *stopping criterion* is

reached. The following subsections will explain each of these parts.

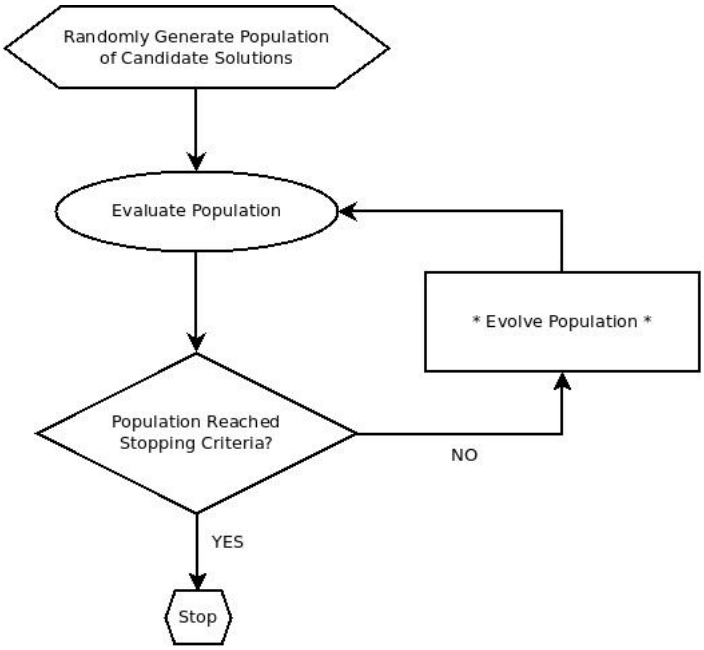


Figure 2.1: Population Individual Modification

### 2.1.1 Population

The population is a key piece to an EA. Each individual in the population represents a possible candidate solution to the problem that we are attemping to solve. The representation of the individual is usually unique to the problem. Generating the initial population can either be done randomly or by some procedural method. The goal of generating the initial population is to create a diverse**(CITE)** enough population to find improved solutions.

### 2.1.2 Evaluation Function

This operator determines the fitness of an individual. Each individual is evaluated and given a fitness score to represent how well the individual performed on the problem. This operation is problem specific and it can be very difficult to determine how a problem should be evaluated. The evaluation function is important for differentiating

individuals. A poor evaluation function can make each of the individuals appear to be similar when they actually have small key differences.

### 2.1.3 Stopping Criterion

Stopping criteria are used to determine when the EA should stop evolving. There are generally three ways stopping criteria can be reached: a maximum number of iterations is reached, the population has converged on the same solution, or the solution has been found.

### 2.1.4 Evolving the Population

The evolutionary process of an EA is what differs in each implementation of an EA. Each algorithm has a different interpretation of how the population should be evolved. Evolving the population consists of using the individuals in the population to create a new population. Later in this work the different interpretations will be explained.

## 2.2 Genetic Algorithm

A genetic algorithm (GA) is a search heuristic that mimic natural selection. In a GA individuals go through a *selection* process and are bred(*crossed*) with other individuals to create new individuals. Individuals in a GA are commonly refered to as *chromosomes*. Figure 2.2 depicts how evolution occurs in a GA.

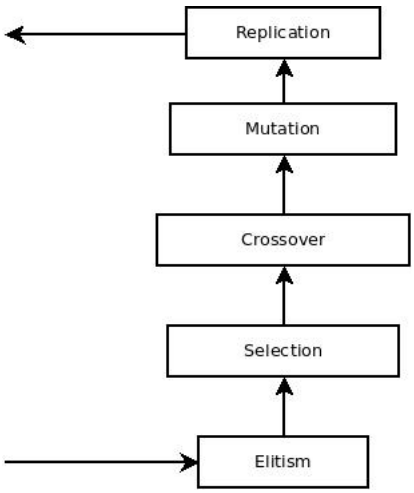


Figure 2.2: GA Evolution

During each generation a new population is created using the previous generation's population. Initially the best individuals from the previous population might be copied directly into the new population using an operator called *elitism*. To obtain the remaining individuals needed to fill the new population a *selection* process occurs. Two individuals are chosen using a *selection* method and then one of three options can occur: *crossover*, *mutation*, or *replication*. Crossover mixes two individuals together to create two new individuals, mutation randomly modifies each individual individually, and replication copies the individuals. The two individuals are then placed in the new population and the process is repeated until the new population is the same size as the previous population. Figure 2.2 depicts how evolution occurs in a GA. See Subsection 2.2.2 for more details on the operators discussed.

### 2.2.1 Chromosome

The individuals of a GA represent possible candidate solutions to the problem. Typically a chromosome is represented as an array where each index of the array represents a property of the candidate solution. There are no restrictions to the encoding of a chromosome but each property of the chromosome must be independant from the others. The simplest example of a representation is a binary array of 1's and 0's, as shown in Figure 2.2.1. In the sample chromosomes provided the 1's and 0's might represent whether a feature is enabled or disabled, 1 being enabled and 0 being disabled,

in the candidate solution.

| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|

Figure 2.3: Simple Chromosome Represention

### 2.2.2 Genetic Operators

Each of the following operators represent a piece of a genetic algorithm. They facilitate the evolutionary process in the effort to find better candidate solutions. Each of these operators has a unique purpose in the search algorthm but there are many different ways in which these goals can be carried out. Only a few of the different methods will be described in this work.

*Selection Operator*: This operator is very important to the *Crossover* and *Mutation* operators. The idea behind this operator is to put selection pressure on the population during the evolutionary process. Individuals with a better fitness score should be allowed a better chance of breeding to create the next population. During the selection process two individuals are chosen for breeding or reproduction. There are several varieties of selection methods but only *k-Tournament selection* will be explained as this is the method used in the algorithm in **(chapter goes here)**.

The $k$-Tournament selection method works by randomly selecting $k$ individuals from the population, where $k$ is less than the number of individuals in the population, and selecting the individual that has the best fitness score from the $k$ individuals. The value of $k$ should be relatively small compared to the size of the population. If the value of $k$ is too large it would defeat the purpose of this selection method. For example, if there is a population size of 100 a suitable value of $k$ is around 2-5.

*Crossover Operator*: This operator is essential to evolving the individuals of the population. Crossover is the mechanism by which two individuals breed to create two new individuals. With respect to the evolutionary process, crossover exploits the current information that is contained within the population in order to find improved individuals. The most widely used type of crossover is N-point crossover.

N-point crossover works by randomly selecting N cutting points and swapping the information between the two individuals along those N points. Figure 2.4 demonstrates

how the swapping of information occurs during 2-point crossover.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Parent 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| Parent 2 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

---

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Child 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| Child 2 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Figure 2.4: 2-Point Crossover

*Mutation Operator*: The mutation operator is used to introduce random changes to the individuals during evolution. Mutations to individuals are a way to explore the search space. Depending on how the initial population was created there may not be the necessary information in the population to find the optimal solution with crossover alone. Mutations allow for new information to possibly be introduced into the population. A common type of mutation is single-point mutation where a single index in your individual is modified. Figure 2.5 demonstrates single-point mutation.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Individual | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |

---

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Mutant | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |

Figure 2.5: Single-point Mutation

*Elitism Operator*: During each generation of the genetic algorithm a new population is created using the individuals from the population in the previous generation. The new population is bred from the previous individuals with the hopes of creating better individuals. Sometimes this is not the case and the population can end up losing valuable information from individuals that were not chosen during the selection process. To prevent this from happening the elitism operator was create. The elitism operator works by seeding the next generations population with the individuals with the best fitness score. Typically only the top 1% of individuals are copied into the next generation.

## 2.3 Recentering Genetic Algorithm

The recentering genetic algorithm (RGA) is a variation of the recentering-restarting genetic algorithm (RRGA) [9] [10] which has had success in avoiding local minima. The RRGA is used to avoid fixating on local optima. RRGA works by performing a series of standard GA runs. Each run uses the final population from the previous run as its starting population with some adjustments. At the beginning of a run the RRGA selects a center, which is a possible candidate solution to the problem, and at the end of each basic GA run the center is compared to the best individual in the population. If the best individual is better than the current center it is replaced with the best individual and the whole process is repeated. The center is used as a baseline for generating the population in the next run.

The RGA works similarly to the RRGA but there is no center for the population. Instead a basic GA is allowed to run until the population's fitness scores begins to converge. After the population has converged upon a minimum diversity, new individuals are introduced to the population. Duplicate individuals are removed from the population and new individuals that have not yet been in any population take their place. For example, if there is a population size of 100 and the convergence rate is 5% then after all the duplicates are removed there will only be 5 individuals remaining and 95 new individuals will be inserted into the population. Algorithm 1 shows the pseudo-code of the restarting method.

**Algorithm 1** Restarting the population

```
if population has converged to minimum diversity then
    remove all duplicate individuals;
    while population not full do
        insert random draw from generated individuals into population;
    end while
end if
```

## 2.4 Particle Swarm Optimization

## 2.5 Differential Evolution

# Chapter 3

# Previous Research

Before we can begin explaining the techniques we used in the next chapter it is necessary that we explain related research in this field.

## 3.1 Quantum Mechanics/Molecular Mechanics

In previous work [11] the authors used DFT-QM/MM and R-QM/MM techniques to find close approximations of the experimental EXAFS spectrum of OEC in $S_1$. The EXAFS spectrum used in their calculations was at a poorer resolution compared to the spectra used in the experiments in our work.

Density functional theory quantum mechanics/molecular mechanics (DFT-QM/MM) **??** uses the atoms spatially dependent electron density to determine the position of each atom. Since DFT largely uses function approximations this approach is very limited.

To increase their accuracy the researchers used a refined quantum mechanics/molecular mechanics (R-QM/MM) technique. This approach iteratively adjusted the molecular structure of the molecule and attempted to minimize a scoring function defined in terms of the sum of squared deviations between the experimental and calculated EXAFS spectra. A quadratic penalty was applied to each atom to ensure that their positions did not deviate too far from their original position in order to keep the energy of the system at a minimum.

The researchers speculated that even though the R-QM/MM technique was able to generate an EXAFS spectra closer to the experimental spectra their solution was only a local solution because it was based on their original DFT-QM/MM solution.

Later in [12] the same research group repeated their original experiments performed in [11] with updated X-ray diffraction (XRD) data that had a resolution of 1.9Å. They has success rerunning the DFT-QM/MM experiment followed by the R-QM/MM experiment but still had the same speculations about remaining in a local solution.

# Chapter 4

# Methodology

## 4.1 Problem Encoding

A molecule consists of a number of atoms. Each of these atoms has its own 3-dimensional position within the molecule. For the structure refinement problem the individual 3-dimensional position values are not important. The important information about this problem is how the atoms are positioned with respect to each other. Two different forms of representation were used in this work. For each of these representations the number values are shown in Angstroms (Å).

The initial run of experiments used a representation that maintained the initial atomic positions of each atom. The 3-dimensional coordinates were treated as a list of coordinates as shown in 4.1. Using this representation meant that during any form of crossover the tuple of X, Y and Z values would stay together.

| X | Y | Z |
|---|---|---|
| 14.451 | -13.346 | 1.133 |
| 15.336 | -13.488 | 2.014 |
| 13.005 | -13.364 | 1.452 |
| 0.019 | 0.011 | 0.045 |
| ... | ... | ... |

Figure 4.1: Representation 1

| 14.451 |
|---|
| -13.346 |
| 1.133 |
| 15.336 |
| -13.488 |
| 2.014 |
| 13.005 |
| ... |

Figure 4.2: Representation 2

Algorithms such as particle swarm optimization and differential evolution called for a more flexible representation. The other representation used was simply a list of values. The initial list of 3-dimensional coordinates was converted to a single list of decimal points as shown in 4.2. It is important to note that both of these representations are showing the same information. For fitness evaluation the list of number was converted back to a list of 3-dimensional coordinates by taking segments of three numbers to create a 3-dimensional position.

## 4.2 Population Generation

An initial population of different individuals needed to be created in order to begin refining the OEC atomic structure using an evolutionary algorithm. The initial OEC atomic structure came from the crystallographic photosystem II (PSII) structure [13]. It is available in the Protein Data Bank (PDB) [14] as PDB ID 3ARC. During the initial stages of experimentation the initial populations were created by randomly adjusting the atoms within this initial structure. To create a new individual each atom within the atomic structure would be randomly moved by 0.05-0.5Å. This form of population generation was quickly discarded because many of these individuals were either chemically infeasible or generated erroneous EXAFS spectra.

### 4.2.1 Molecular Dynamics Simulation

An alternative method of population generation was needed to generate individuals that were usable in the experiments. To ensure that the atomic structure was as stable

as possible, the structure was put into a molecular dynamics simulation. While in this simulation the molecule is allowed to act as if it were in the real world. The atoms were allowed to move freely in space until the overall temperature of the system was reasonably low. This acted as the baseline atomic structure for all tests. NAMD [7] was used to run the molecular dynamics simulations.

Once the atom structure was stable the temperature within the system was increased. The increased temperature causes the atoms to oscillate their positions but still remain chemically feasible. During this process stapshots of the molecules atomic structure were recorded. The simulation was allowed to run for 10 000 steps and 10 000 snapshots of the atomic structure were recorded. Each of these snapshots creates a feasible individual for the experiments.

Since 10 000 individuals is more than enough individuals to seed the populations the best individuals were picked. The generated atomic structures were run through IFEFFIT **??** and compared to the target EXAFS spectra. The top 3% (roughly 300) individuals were used to generate the initial populations in the evolutionary algorithms.

The atomic structures that were generated contained 1269 chemical elements. For the purposes of OEC structure refinement only 79 specific atoms were required for EXAFS analysis. The genetic algorithm only used the 79 atoms that required refinement.

## 4.3 Genetic Operators

### 4.3.1 Crossover

The basic one-point crossover operator was chosen for the experiments (Described in Subsection 2.2.2). One point crossover is generally less destructive to the individuals than other forms of crossover.

Table 4.1: Minimum Move Required at 1%

| Element | 1% Difference | 5% Difference |
|---|---|---|
| O | 0.025Å | 0.5Å |
| Mn | 0.01Å | 0.5Å |
| Ca | 1Å | 5Å |
| C | 0.5Å | 5Å |
| N | 0.5Å | 5Å |
| H | 5Å | 5Å |

### 4.3.2 Mutation

**Placeholder until represenation is decided.**

For the mutation operator a single atomic coordinate will be moved. A random atomic coordinate is selected from the individual and its position is moved randomly by 0.05Å using Euclidean distance. The resulting position will be 0.05Å away from its original position. In order to determine how much distance the atomic position should be moved, an analysis was needed to learn more about how changing atomic positions affects the calculated EXAFS spectra.

The analysis consisted of moving each atom, individually, in a variety of directions and calculating its RMSD score. Each atom was moved in a total of six directions (±X, ±Y, and ±Z), at a variety of distances (0.001Å, 0.005Å, 0.01Å, 0.025Å, 0.05Å, 0.1Å, 0.5Å, 1Å, and 5Å). This was done to determine how much movement was required of an atom to make a significant change to the RMSD score. Table 4.1 shows results of how much movement is required to produce a 1% and 5% change to their RMSD scores. Since there is more than one instance of each chemical element in OEC, the distance chosen was the first distance that produced the minimum change because the goal was to find the absolute minimum for each chemical element.

The value of 0.05Å was chosen for the experiments as a middle ground that could be applied to each chemical element. It should be noted that the value of 0.05Å is particular to OEC. A similar analysis could be done to determine the minimum move distance for each element in another chemical complex.

### 4.3.3 Selection

For the selection operator a 3-tournament selection was used.

## 4.4 Parameters

# Bibliography


[1] D. J. Vinyard, G. M. Ananyev, and G. C. Dismukes, "Photosystem II: The reaction center of oxygenic photosynthesis.," *Annual Review of Biochemistry*, vol. 82, pp. 577 – 606, 2013.

[2] J. Yano and J. Kern, "Manganese: The oxygen-evolving complex and models," *Encyclopedia of Inorganic and Bioinorganic Chemistry*, 2006.

[3] M. Newville, "Fundamentals of xafs," *Consortium for Advanced Radiation Sources, University of Chicago (USA)[http://xafs. org]*, 2004.

[4] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *Journal of the American Chemical Society*, vol. 117, no. 19, pp. 5179–5197, 1995.

[5] J. W. Ponder, D. A. Case, *et al.*, "Force fields for protein simulations," *Advances in protein chemistry*, vol. 66, pp. 27–86, 2003.

[6] T. U. of Chicago, "Ifeffit: Interactive xafs analysis." Accessed: 2014-04-01.

[7] Theoretical and C. B. Group, "Namd - scalable molecular dynamics." Accessed: 2014-04-01.

[8] Theoretical and C. B. Group, "Namd energy plugin." Accessed: 2014-04-01.

[9] J. Hughes, S. Houghten, and D. Ashlock, "Recentering, reanchoring & restarting an evolutionary algorithm," in *Nature and Biologically Inspired Computing (NaBIC), 2013 World Congress on*, pp. 76–83, IEEE, 2013.

[10] J. Hughes, J. A. Brown, S. Houghten, and D. Ashlock, "Edit metric decoding: Representation strikes back," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 229–236, IEEE, 2013.

[11] E. M. Sproviero, J. A. Gascón, J. P. McEvoy, G. W. Brudvig, and V. S. Batista, "A model of the oxygen-evolving center of photosystem II predicted by structural refinement based on exafs simulations," *Journal of the American Chemical Society*, vol. 130, no. 21, pp. 6728–6730, 2008.

[12] S. Luber, I. Rivalta, Y. Umena, K. Kawakami, J.-R. Shen, N. Kamiya, G. W. Brudvig, and V. S. Batista, "$S_1$-state model of the $O_2$-evolving complex of photosystem II," *Biochemistry*, vol. 50, no. 29, pp. 6308–6311, 2011.

[13] Y. Umena, K. Kawakami, J.-R. Shen, and N. Kamiya, "Crystal structure of oxygen-evolving photosystem ii at a resolution of 1.9 Å," *Nature*, vol. 473, no. 7345, pp. 55–60, 2011.

[14] "Crystal structure of oxygen-evolving photosystem ii at 1.9 angstrom resolution."