DEVELOPMENT OF DIFFERENTIAL EVOLUTION ALGORITHMS APPLIED TO CRYSTAL STRUCTURE SOLUTION FROM POWDER DIFFRACTION DATA

By

Duncan Bell

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Chemistry
College of Engineering and Physical Sciences
The University of Birmingham
July 2012

Abstract

An understanding of the crystal structure can aid in the rationalisation of physicochemical properties exhibited by a crystalline material. Advances in the area of direct space crystal structure solution means that it is becoming easier to determine crystal structures from powder diffraction data. However, due to the number of structural models generated during structure solution calculations, direct space methods are computationally demanding.

Work presented in this thesis reports the optimisation of a differential evolution (DE) algorithm and a cultural differential evolution (CDE) algorithm to reduce the computational demands of direct space methods. Characteristics particular to certain crystal structures are identified as having a significant effect on the efficiency and robustness of structure solution calculations by DE and CDE.

The development of a new algorithm that closely mimics the natural evolution of a species is discussed. Results presented in this thesis demonstrate that this new algorithm is significantly more efficient than the DE algorithm.

Despite the complexity of powder diffraction patterns recorded for biphasic crystalline materials, in this thesis, the successful development and application of a direct space method to the simultaneous structure solution of two crystals from a biphasic powder pattern is reported.

Contents

Chapter 1: Introduction

Chapter 2: Methodology

Chapter 3: Optimising DE and CDE

Chapter 4: Eugenic Differential evolution

Chapter 6: Conclusions and further work

Appendix A: Experimental method

Appendix C: Cooperative coevolutionary algorithms

Chapter 1 Introduction

1.1 Understanding crystal structure

In modern society organic crystalline materials are used in a wide range of applications including pharmaceuticals, dyes, lasers and nonlinear optics. [1-4] The desire to treat an increasing range of health problems means that research into organic crystalline materials that have beneficial pharmaceutical properties is becoming increasingly lucrative for industry. An organic crystalline material can be defined as a type of matter in which organic molecules are arranged in a three-dimensional translational periodic pattern. Polymorphism can be defined as the ability of a chemical compound to adopt more than one crystal structure. [3-6]

A significant proportion of the compounds that form the active component of modern pharmaceutical products are administered in the crystalline form. [1,3-5,7] Crystalline compounds are preferred because the act of crystallisation usually removes any impurities from the final product (although impurities such as unreacted reagents can cocrystallise with the desired product), and because the solubility and bioactivity of crystalline phases are generally more constant compared to other formulations such as amorphous materials, [7] which are often hygroscopic and can potentially crystallise in forms with undesirable bioactivity. However, many crystalline pharmaceutical compounds can occur in multiple polymorphic forms. Different geometric rearrangement of the atoms, ions or molecules of a crystal structure can have a considerable affect on the bioactivity of a crystalline compound. For example, mebendazole, 5-benzoyl-2-benzimidazolecarbamic acid methyl ester (a general purpose anthelminthic) [8, 9] can be recrystallised in three common anhydrous polymorphic forms. The solubility of the forms decreases in the order B > C > A. However, form B is toxic, form A is not bioactive and renders medication useless when it is present in concentrations >= 30%, only form C displays the desired bioactivity.

Additionally one polymorphic form may have physical properties that make it more suitable for delivery to patients. The commonly used antibacterial agent sulfamerazine [5] can be crystallised in two main polymorphic forms; polymorph [I] which can be recrystallised from methanol and polymorph [II] which can be recrystallised from acetonitrile. Figure 1.1 below (taken from reference 5) shows the different crystal structures adopted by polymorphs [I] and [II].

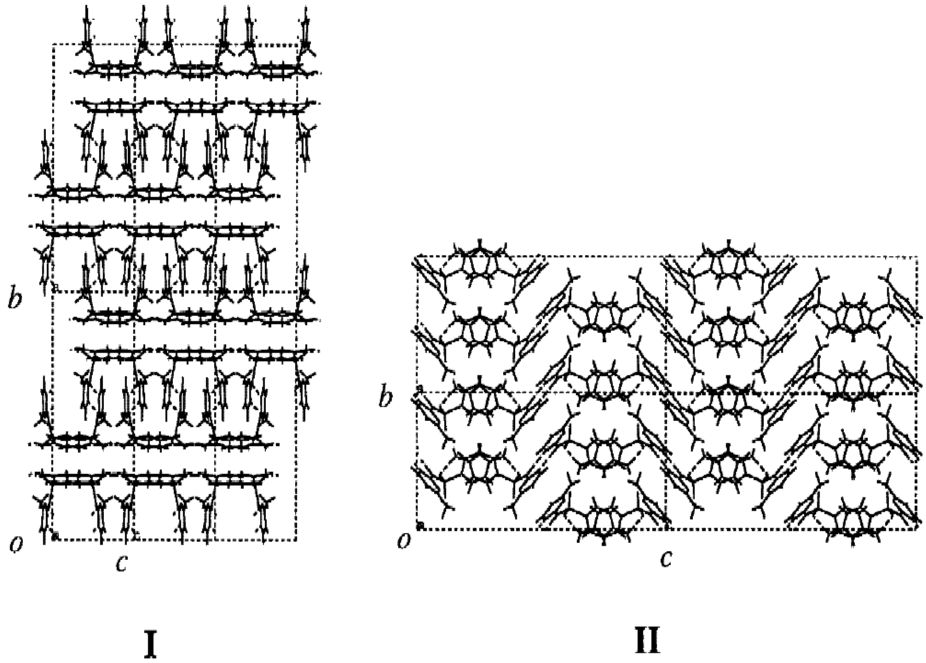


Figure **1.1**, the crystal structures of two polymorphic forms of sulfamerazine, polymorph [I] on the left and polymorph [II] on the right [5]

The figure shows that the crystal structure of both of the polymorphs is built up from stacked sheets of interlinked molecules of sulfamerazine. Whereas molecules in the same sheet are linked by relatively strong hydrogen bonds, the separate sheets are only held together by weak Van der Waals forces. The figure shows that whereas the sheets forming polymorph [I] are flat, the sheets forming polymorph [II] are puckered. This means that the sheets forming polymorph [I] can slide across each other more easily than the sheets forming polymorph [II]. This means that crystals of form [I] are more easily compressed than crystals of form [II]. The existence of slip planes in form [I] means that it can be compressed into more robust tablets than form [II].

Poor biopharmaceutical properties rather than toxicity or lack of efficacy mean that < 1% of potential drug compounds are finally marketed. [10] Alternative formulations can be used to address this, such as cocrystallisation which can be used to tune the biopharmaceutical properties of a bioactive crystalline compound.

A cocrystal can be defined as a neutrally charged supramolecular assembly of homo or heteromeric synthons, [7, 10] thus the components of a cocrystal occupy the same crystal lattice. To be considered as a cocrystal, the cocrystal and its individual components must all be in the solid state at standard temperature and pressure. [7] Thus a solvate (in which solvent molecules occupy sights within the lattice) cannot be classed as a cocrystal at room temperature and pressure. Similarly if the separate synthons become electrically charged for example through the transfer of a proton from one synthon to another, the solid supramolecular assembly is considered as a salt.

The physical and chemical properties of a specific crystalline compound can be manipulated in a systematic fashion by crystallising the compound with a homologous series of complementary supramolecular synthons. [7, 10] One example of property control through cocrystallisation is through the use of a family of dicarboxylic acids. The melting point of aliphatic dicarboxylic acids containing an even number of carbon atoms in the backbone of the molecule decreases monotonically as the length of the carbon backbone increases. [10] If a homologous series of aliphatic dicarboxylic acids is used to form a series of isostructural cocrystals with a specific bioactive compound, the melting point of the cocrystal is reflected in the melting point of the isolated dicarboxylic acid. Thus the melting point (thermal stability) of the cocrystal can be engineered. The solubility of a bioactive compound can also be controlled through cocrystallisation with different molecules as demonstrated by the solubility of the anticancer drug hexamethylenebisacetamide being improved by a factor of 2.5 [10] by cocrystallisation with different dicarboxylic acids. Since cocrystallisation does not involve changing the molecular structure of the participant molecules, the solubility of the drug molecule can be manipulated without altering the molecular structure of the bioactive molecule.

To maximise the utility of a compound that can crystallise in multiple forms it is necessary to control the production of the compound in order to maximise synthesis of the preferred form. Interactions between solvent and product molecules can promote or inhibit the crystallisation of a particular form. Sulphathiazole can be crystallised in four main forms [7] but the choice of solvent and the presence of a reaction precursor during the final crystallisation stage affect the mechanism of crystal growth and hence the form of sulphathiazole obtained. The use of propanol as solvent prevents the formation of a hydrogen-bond motif common to forms [II]-[IV] so only form [I] can be obtained. Although the growth of crystals of form [I]-[III] can continue in the presence of a reaction precursor the growth mechanism adopted by crystals of form [IV] is inhibited. Thus form [IV] can only be obtained from pure solutions of sulphathiazole. Unless it is possible to perform a systematic study of all the synthetic routes and crystallisation procedures that can be employed to synthesise a target material (thereby identify the most suitable combination of production procedures), it is useful to have an understanding of how the choice of reagents and solvent used to synthesise the target material affect the formation, the crystallisation and the structure of the target material. It is key to access the structural information in organic crystalline materials such as polymorphs and cocrystals so that synthetic routes and crystallisation conditions can be used in crystal design and favour the crystallisation of the compound in the desired form.

1.2 X-ray crystallography

“Crystallography as a discipline has its origins in the pre diffraction era when the term referred to the study of the morphology of crystals”. [2] Nineteenth century crystallographers could measure the angles between different faces of a crystal and the lengths of face edges to determine the symmetry of the crystal and assign relative lengths to the three crystal axes. At the beginning of the 20th century Bragg demonstrated a technique that could be used to determine the actual location of atoms inside the unit cell of a crystal. Since the atoms, ions or molecules that form the crystal structure are arranged in a three-dimensional periodic pattern, the crystal structure can be visualised as lattice planes that intersect each other at regular intervals. The length of atomic bonds is of the order of one angstrom (1×10^{-10} m), and hence the spacing between different planes is also of the order of one angstrom. Thus the crystal structure acts as a three-dimensional diffraction grating for electromagnetic radiation of an appropriate wavelength.

X-rays are high energy electromagnetic radiation with wavelengths in the range 1-10s of angstroms. Thus they diffract as they travel through a crystal. X-rays interact much more strongly with electrons than with atomic nuclei so when travelling through a crystal the X-rays are diffracted by areas of high electron density rather than the atomic nuclei themselves. However since the electrons within the crystal lattice are localised around atomic nuclei the X-rays can be used to determine the location of the atoms within the unit cell of the crystal. Bragg demonstrated that if a beam of monochromatic X-rays is directed through a perfect single crystal onto a photographic plate, a series of spots (diffraction maxima) is observed on the plate. The distances between the different spots can be used to calculate the distance between the different lattice planes, and along with intensity data, this information can be used to determine the position of individual atoms inside the unit cell of the crystal.

1.2.1 Bragg's Law

If a specific lattice plane is orientated at an angle (Q) to an incoming beam of X-rays, X-rays will be reflected from the plane at an angle Q. X-rays will also be reflected at an angle q from any other parallel lattice planes. If a monochromatic X-ray beam is directed at an angle Q onto parallel planes A and B that are separated by a distance (D) monochromatic X-rays will be reflected from both planes with an angle Q. X-rays that travel through plane A without being reflected and instead are reflected by plane B will travel further than X-rays that are reflected by plane A. The path length of X-rays that are reflected from plane B is (2 x D x sinQ) longer than the path length of X-rays that are reflected by plane A. If X-rays that are reflected by plane B are exactly in phase with X-rays reflected by plane A, complete constructive interference will occur and produce a single X-ray with twice the amplitude of the two reflected X-rays. Conversely if the X-rays reflected by plane B are exactly half of a wavelength (180°) out of phase with X-rays reflected by plane A, complete destructive interference will occur. If the two reflected X-rays are neither exactly in phase or 180° out of phase. less destructive interference will occur and a low intensity X-ray will be detected. Since the intensity of a wave is proportional to the square of the amplitude of the wave, complete constructive interference produces X-rays with high intensity that are observed as bright diffraction maxima on the photographic plate. If intense diffraction maxima are observed at a certain angle Q it can be assumed that complete constructive interference is occurring within the crystal structure. This means that the difference in the path lengths of two different X-rays reflected by different parallel planes is an integer multiple of the wavelength of the X-ray beam. Thus the wavelength of the X-ray beam is related to (D x sinQ) by an integer multiple N. For simplicity of calculation, crystallographers assign N a value of one. Thus if the X-ray wavelength and the angle Q are known, it is possible to calculate the distance D that separates two parallel lattice planes.

Bragg's law states that (N x L = 2 x D x sinQ) [11] where L is the X-ray wavelength in metres, N = 1, D is the interplanar separation in metres and Q is the angle at which the incoming X-ray beam strikes parallel planes of atoms. The factor of two accounts for the fact that one of the X-rays travels through plane A without reflecting to plane B, is reflected by plane B and travels back to plane A. Thus D = L/(2 x sinQ).

1.2.2 The Phase Problem

The diffraction maxima observed during an X-ray crystal diffraction experiment are a reciprocal representation of the crystal structure. The position of the observed diffraction maxima is determined by the unit cell parameters and the symmetry of the crystal. The intensity of the diffraction maxima is determined by the distribution of atoms in the unit cell. The intensity of maxima I(s) is proportional to the square of a complex number, the structure factor amplitude |F|. The scattering vector (s) is a point on the crystal lattice corresponding to specific diffraction maxima. The structure factor is determined by both the amplitude and phase of the detected X-ray and hence it is necessary to know both the amplitude and phase to determine the position of atoms inside the unit cell. [12] However it is the intensity of a reflection that is measured and this is related to the structure factor by equation 1.1. Each diffraction maxima has an associated structure factor F(s) with amplitude |F(s)| and phase α(s). The structure factor is related to the distribution of scattering matter by equation 1.2. Thus as the phase of a detected X-ray cannot be measured it is not possible to determine the crystal structure directly from a diffraction pattern.

$$I\,(\mathbf{s}) \propto |F\,(\mathbf{s})|^2 \tag{1.1}$$

$$F(\mathbf{s}) = |F(\mathbf{s})| exp[2\pi i\,\alpha(\mathbf{s})] = \int \rho(\mathbf{r}) exp(2\pi i\mathbf{s}\cdot\mathbf{r})d\mathbf{r} \tag{1.2}$$

A technique that can be employed to estimate the phases is the Patterson technique. [12-16] This technique involves determining the position of significantly heavy atoms inside the unit cell. Heavy atoms such as transition metals or halides with many electrons scatter X-rays more strongly than lighter atoms such as C, N and O with few electrons. The scattering of X-rays by these heavy atoms produces diffraction maxima with significantly high intensity. If the crystal structure only contains a relatively small number of heavy atoms it is possible to determine the position of the heavy atoms from the small number of highly intense diffraction maxima. Models can then be used to estimate the phase of these highly intense maxima. The best model can be used to assign appropriate phases to the remaining maxima that are caused by diffraction by the lighter atoms. However when the crystal structure contains no dominant scatterers, the Patterson technique cannot be used to estimate the phases. In these circumstances direct methods are used. Direct methods are commonly used to determine crystal structure when high quality data can be obtained from single crystal diffraction experiments. [12, 16, 17] Direct methods can be used to determine crystal structure by exploiting known phase relationships that occur between specific groups of maxima. Models are generated by assigning estimated phase values to individual maxima. Statistical analysis such as symbolic addition is used to evaluate each of the models and determine the best. The best model is then used to determine the crystal structure.

1.2.3 Powder Diffraction

In cases when it is not possible to grow a sufficiently perfect crystal to perform single crystal X-ray diffraction experiments, powder diffraction experiments can be used to determine the crystal structure. Powder X-ray diffraction involves passing a monochromatic X-ray beam through a powder sample of micro crystals (crystallites). The crystallites are packed in a sample holder and it is assumed that the orientation of each crystallite with respect to the sample holder is random. As the number of crystallites in the sample increases, the probability that each different lattice plane in the crystal structure is orientated in the same direction with respect to the sample holder increases. When a monochromatic X-ray beam is passed through a powder sample containing many crystallites diffraction occurs from all the lattice planes simultaneously. As the sample holder is rotated the angle between each set of parallel lattice planes with respect to the X-ray beam changes. At a certain angle of Q each set of parallel planes will satisfy (N x L=2 x D x sinQ) and produce an observable diffraction maximum. [11]

Single crystal and powder diffraction patterns both contain the same amount of structural information but this information is more easily extracted directly from a single crystal pattern. In a single crystal diffraction pattern, each diffraction maximum corresponds to one particular set of parallel lattice planes that satisfy (N x L=2 x D x sinQ). However, in a powder pattern, one peak may correspond to multiple different lattice planes that simultaneously satisfy (N x L=2 x D x sinQ). Powder diffraction results in the production of cones of intensity, [18] (figure 1.2) rather than the clearly defined diffraction maxima observed during single crystal diffraction experiments.

The powder diffraction pattern is obtained by sampling the diffraction cones along the 2Q axis. This results in diffraction data from all three dimensions being compressed into one dimension, causing significant overlap of diffraction peaks. This overlap makes it hard to

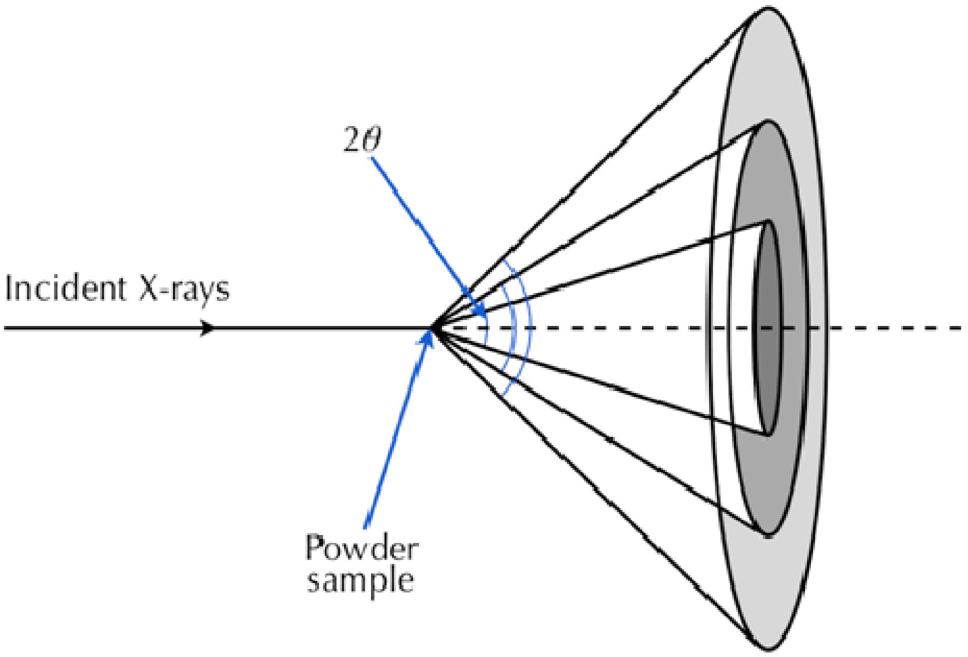


Figure 1.2, Diffraction cones resulting from simultaneous crystal diffraction by multiple sets of lattice planes from a powder sample (figure taken from [18]).

determine whether an observed peak results from the diffraction by a single set of planes or the overlap of multiple peaks that each result from diffraction by different sets of planes. Consequentially it is difficult to assign accurate integrated intensities to individual observed peaks. In addition, peak overlap significantly reduces the number of individual peaks that can be resolved in a powder pattern to the order of 200 compared to around 1500 resolvable maxima in a single crystal diffraction pattern. [3] Thus a powder pattern readily yields < 10% of the resolvable information that can be extracted from a single crystal diffraction pattern.

Despite the reduction in the amount of structural information that can be extracted directly from a powder pattern, it is still possible to index the powder pattern using resolvable peaks to determine the unit cell parameters and crystal symmetry, [3,4] and to determine the distribution of atoms in the unit cell using direct space techniques to interpret the intensity data.

The crystal structures of cimetidine [19] and polymorph V of sulphathiazole [20] have been determined by application of direct methods to powder diffraction data. However when direct methods are used to determine the crystal structure of molecular crystals it is often necessary to use diffraction data recorded at a synchrotron source rather than a conventional laboratory diffractometer. This is because determination of crystal structure using direct methods requires the measurement of accurate peak intensities from the diffraction pattern. Molecular crystals usually have large unit cells and low crystal symmetry which results in severe peak overlap in the diffraction pattern. [12] The severe peak overlap frustrates the accurate measurement of individual peak intensities. The peaks observed in diffraction patterns recorded at synchrotron sources are often more clearly defined and less overlapped than peaks observed in diffraction patterns recorded using conventional laboratory diffractometers. Thus it is less difficult to accurately measure the intensity of diffraction peaks recorded at synchrotron sources and use direct methods to determine the crystal structure. Direct methods are however significantly more appropriate for structure determination of inorganic crystals that usually have small unit cells and high symmetry resulting in diffraction patterns that suffer less from peak overlap.

The crystal structure of molecular crystals is increasingly being determined by the direct space technique. This is because the ability of the technique to determine crystal structure is not significantly reduced by peak overlap and the direct space technique requires little intensity data to be extracted directly from the diffraction pattern. Only the position of diffraction peaks needs to be measured to determine the unit cell parameters and crystal symmetry.

1.3. The direct space technique

The direct space approach involves the computer generation of a three-dimensional model of the crystal structure in a unit cell that has been derived using the resolvable peak positions in the powder diffraction pattern. The plausibility of the computer-generated model is then evaluated using various "cost functions". [21-23] The idea behind this approach is that the best model as defined by the best cost function will be representative of the real crystal structure, [24] thus the crystal structure can be determined without having to ‘extract’ a significant amount of structural information directly from the powder diffraction pattern.

1.3.1. Cost functions

One of the cost functions used in direct space methods is the R_{wp} factor, also commonly used in the Rietveld refinement process. [25,26] This technique involves simulating a diffraction pattern for a computer generated model of a crystal structure and quantitatively comparing the simulated pattern with the real diffraction pattern recorded for the sample. To quantitatively compare the simulated pattern with the real pattern, the Rietveld procedure digitises the two diffraction patterns. Each digitised point in each of the patterns is considered as an individual intensity measurement. The goodness of fit between the simulated and real patterns is determined by measuring the difference in the intensity of equivalent points in the simulated and real diffraction patterns. Equation 1.3 below shows the most common quantity used to compare digitised simulated and real diffraction patterns, R_{wp}.

$$R_{\mathrm{wp}} = \frac{\sum w(I_{\mathrm{obs}} - I_{\mathrm{calc}})^2}{\sum w(I_{\mathrm{obs}})^2} \qquad (1.3)$$

Thus the R_{wp} cost function does not directly assess the integrated intensity of peaks in the experimental diffraction pattern. The Rietveld technique can be used to simulate a powder diffraction pattern for a computer generated model of a crystal structure and compare it with an experimental powder diffraction pattern recorded for a real powder sample of the material. Thus peak overlap is taken into account and the need to assign individual peak intensities is negated by matching the more complex whole profile shape [4, 21, 27] of the simulated and experimental powder diffraction patterns.

Additionally, the R_{wp} factor cost function is tolerant with respect to the quality of the experimental data, as long as the pattern simulated from the computer generated model is of similar quality. A study in which the quality of the experimental powder diffraction pattern recorded for a test compound was manually varied, [28] showed that the correct structure could be consistently found using R_{wp} to locate the structure, despite significant reduction in data quality, although the difference in R_{wp} values between the correct structure and other possible but incorrect structures became smaller as the quality of the experimental pattern decreased. This means that data recorded using laboratory X-ray diffractometers can be used for direct space methods rather than relying on data recorded at synchrotron facilities, which tends to be used for traditional structure solution techniques involving accurate measurement of intensities of individual peaks.

The chi-squared cost function [29,30] can also be used to quantitatively compare simulated and experimental diffraction patterns. However the chi-squared technique does not digitise the entire simulated and experimental patterns, instead the difference in the integrated intensity of peaks in the simulated and experimental patterns is measured. This technique assigns the best cost function value to the simulated pattern that has peaks with the same integrated intensities as equivalent peaks in the experimental pattern. As this technique does not evaluate every point in the diffraction pattern it is a much less computationally demanding cost function than the R_{wp} function, which digitises complete simulated and experimental patterns and compares them point by point.

The crystal structure of form B of famotidine [29] (figure 1.3) has been determined by the direct space technique using a cost function based on chi squared.

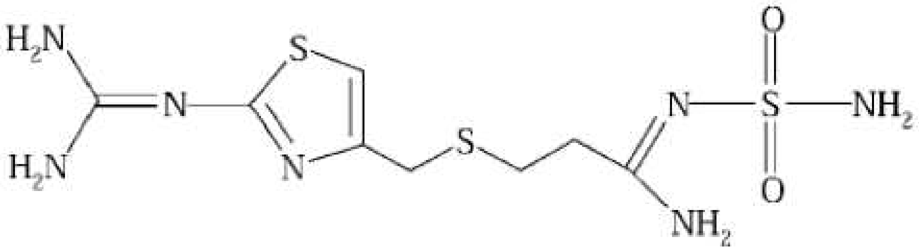


Figure 1.3, The crystal structure of form B of famotidine (figure taken from [29]).

However as the cost function based on chi-squared only uses peak intensities to assess the fitness it can be significantly less accurate than the cost function based on R_{wp}. [28]

The cost function can also take the form of a crystal lattice or potential energy calculation [28,31,32] in which the best solution corresponds to the most energetically favourable structural arrangement. In some cases [28,33] both types of cost functions (potential energy and R_{wp} fit) have been used as a combined weighted function in an attempt to improve the accuracy of direct space crystal structure solution.

It is possible for different crystal structures to have different lattice energies and physical properties but to have similar diffraction patterns. For example; a carboxylic acid and organic amide adduct could either form a cocrystal in which each of the two components are neutrally charged, or as a salt involving the transfer of a proton from the carboxylic acid to the amide. It is likely that the cocrystal and salt forms would have different lattice energies and solubilities. Since the salt contains charged adducts, the energy of solvation is likely to be significantly greater for the salt than for the neutral cocrystal. However the crystal packing of the cocrystal would be very similar to that of the salt. Indeed, the powder diffraction patterns of the two structures would be very similar as the non-hydrogen atom positions in the structure are essentially identical, with the difference based entirely on the position of the proton which is a weak X-ray scatterer.

Conversely it is possible for similar structures with similar lattice energies [28] to produce different diffraction patterns. Molecules with long alkyl chains can form crystal structures that are potentially complex to determine by direct space techniques that only use an R_{wp} cost function to assess the quality of model structures. [28] The flexibility of the alkyl chain means that many plausible model crystal structures have similar lattice energies but different X-ray diffraction patterns. Thus computer generated models of molecules with a high degree of intramolecular flexibility can adopt the same crystal packing motif and have similar crystal lattice energies but produce different simulated diffraction patterns. However, it is also important that direct space techniques should not rely on cost functions that only use potential energy calculations to assess the quality of model structures as many theoretically plausible structures can have energies within a few kJ/Mol of the correct structure. [31]

1.3.2. Describing the computer generated model

Traditional structure determination methods rely on the intensity of diffraction peaks to provide information on the number and location of atoms present in the unit cell, making them most applicable for small molecules with few atoms, or non-molecular crystalline materials containing heavy atoms that scatter X-rays strongly. [3] Direct space methods however, are most useful for crystal structure solution of large structurally rigid molecules. This is because direct space methods can make use of any inherent knowledge about the structure to be solved, and then apply this knowledge to the structure solution process.

If the structure under study is molecular and the connectivity of the molecule is already known, the structure solution problem becomes a matter of placing the component molecular fragments in the unit cell in the correct position and orientation, [22] rather than the traditional method of correctly locating a collection of unconnected atoms in the unit cell. Under these circumstances, the complexity of the problem is not defined by how many atoms the molecule contains, but how many ways its structurally rigid fragments may orientate with respect to one another and within the unit cell. [3]

Every rigid molecule can be defined as having six degrees of freedom (DOF) in three dimensional space; three components of translation along the x, y and z axes, and three mutually perpendicular axes about which the molecule or fragment can rotate. Additional degrees of freedom such as torsion angles accompany intramolecular rotation, making direct space structure solution of non-rigid molecules more complex. [34] Each individual molecule in a unit cell must be able to move independently, thus if the unit cell contains more than one molecule (a cocrystal for example) [22] each molecule must be described by a separate set of structure parameters, making the structure solution more complex.

1.4. Crystal structure determination by global optimisation

1.4.1. Random search

Many techniques have been developed to tackle the "trial and error" nature of direct space methods, optimising the movement and hence the calculation needed as each trial structure is evaluated. The simplest search technique is a completely random search involving the calculation of the fitness of randomly generated crystal structures within the unit cell. The search is likely to be quicker than an exhaustive search of all possible crystal structures. However, as not every structure is evaluated and the search is not guided in a logical progression from model structures that are poor representations of the actual crystal structure towards better ones, it is difficult to identify whether the structure that is assigned the highest fitness in a single search progression is the global optimum (real crystal structure).

1.4.2. Grid search

The simplest "controlled" search method is the grid search method in which every possible structural arrangement of the molecule is systematically investigated. A grid search has the advantage that as long as the translation and rotation step sizes between successive structures are sufficiently small, every possible arrangement that is defined by the “grid” will be evaluated and the best not missed as the fitness function is run after each structural perturbation. [4] However, it is the investigation of all possibilities that makes the grid search hugely inefficient. [35] For example, in a 10Å side length cube with a coarse translation step of 1Å, 10^3 positions are created on which the molecule can be placed, each of which must be evaluated. In addition, a rigid molecule also has three rotational degrees of freedom, each of which must be evaluated over 360°. If each plane of rotation is split into one degree steps, 359^3 orientation combinations would accompany each grid point in the unit cell, resulting in a $10^3 \times 359^3 = 4.63 \times 10^{10}$ point grid search for even the simplest rigid molecule in a relatively small unit cell.

The time required to conduct a grid search can be reduced by reducing the number of model structures evaluated during the search. This can be done by increasing the distance between the grid points. The above cubic unit cell could be divided into a three-dimensional grid where the distance between grid points is two Ångstroms, thus reducing the number of grid points on which the molecule could be placed to 5^3. The rotation step size could also be increased to five degrees, reducing the total number of combinations and evaluations to $5^3 \times 72^3 = 4.7 \times 10^7$. However increasing the distance from one structure to the next reduces the accuracy of the grid search and increases the probability that the best structure may fall between grid points and is not located or evaluated.

To improve the efficiency of a grid search, additional factors such as potential energy calculations, [36] or symmetry considerations [37] are taken into account to guide the search away from senseless structures, or to reduce the degrees of freedom needed.

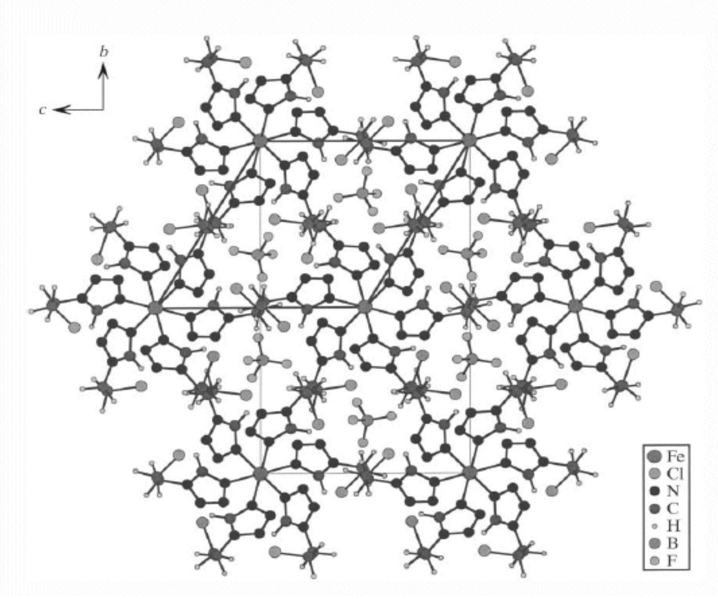


Figure 1.4, Crystal structure of the [Fe(TEEC)$_6$] [II] ion (figure reproduced from [37]).

The [Fe(TEEC)$_6$](BF$_4$)$_2$ complex (figure 1.4) theoretically has 36 degrees of freedom, so its structure would not have been solved successfully without a constrained grid search. Each of the TEEC ligands has 3 x 360° internal rotational DOF, and the [Fe(TEEC)$_6$] [II] unit shares the unit cell with two BF$_4$ counter-ions. [37] In this case, constraints were introduced as the central Fe [II] ion was fixed at (0,0,0), leaving only half of the symmetrical [Fe(TEEC)$_6$] [II] ion with half the number of intramolecular parameters and with no translational parameters to be determined. In addition, the number of grid points used in the search space was reduced by increasing the rotational step size to ten degrees.

Various other methods for guiding the grid search exist, including interatomic potential energy calculations. As most electrostatic potentials between atoms can be calculated with reasonable accuracy, [24] a guided search using this technique may avoid the evaluation of unfeasible structures. [36]

1.4.3. The landscape of global optimisation

To improve the efficiency of the search, it is essential to reintroduce some randomness in order to avoid evaluating every structure. The art is not to use the cost function purely to evaluate the plausibility of the proposed structure, but also as a method for controlling where the search may look next, i.e incorporation of randomness with rules. The cost function (in this case based on the R_{wp} factor between observed and calculated powder patterns), can be used to create a "fitness landscape" or "hypersurface" corresponding to the structural search space (figure 1.5). The hills or maxima in the landscape represent structural arrangements that differ significantly from the real structure; whereas the depressions or minima in the landscape correspond to arrangements that correspond to the correct structure solution that enables a successful Rietveld refinement against the experimental powder data.

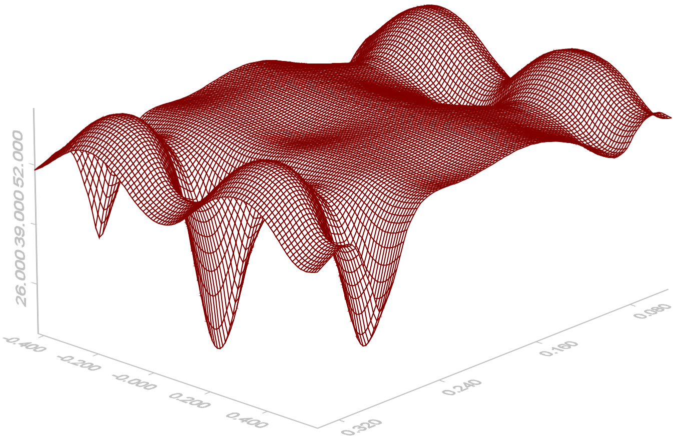


Figure 1.5, Two dimensional representation of an R_{wp} fitness landscape.

A grid search can be used to map out the complete fitness landscape defined by the cost function, whereas the more advanced optimisation algorithms do not explore the whole landscape and are tooled up to explore the depressions most thoroughly. [38] This is still a complex problem, with the landscape having as many dimensions as the model has parameters.

1.4.4. The Monte Carlo Method

The Metropolis Monte Carlo (MC) method [39] is a common approach that can be considered as the next step up from the grid search, in that it is a random search that seeks improvements with each progressive structure. The algorithm begins by randomly placing the molecule or structure under consideration in the unit cell, and calculates the fitness, e.g. R_{wp}, for that particular arrangement. Each parameter of the problem is then altered by a random amount, but constrained by a user defined maximum step size. This produces a "child" structure which is evaluated by the cost function. This process involves calculation of the value Z defined as the R_{wp} fitness of the child - the R_{wp} fitness of the parent. If Z <= 0 this indicates that the child is fitter than the parent so the child is immediately accepted and the parent discarded. If Z > 0 this indicates that the child is less fit than the parent. However the child is not automatically discarded. The child can still replace the parent with a degree of probability that can be influenced by the user (otherwise this would be a straightforward cost function minimisation). If Z > 0 a random number between zero and one is automatically generated, the child is then accepted if the random value is <= exp(-Z/S), where S is a scale factor that can be considered akin to temperature and is assigned a value by the user. [14] If the random number is > exp(-Z/S) the child is rejected, the parent is retained and an alternative child created. To prevent the Monte Carlo search becoming trapped in the first minima that it finds, it has the ability to escape by acceptance of children with lower fitness than their parents, (when the random number <= exp(-Z/S)). As the value of S is manually decreased by the user the probability that a child structure with a low fitness value is able to replace a parent structure with a higher fitness value decreases. Thus as the search progresses the search becomes confined in the minima. This step-by-step search generates a ‘Markov chain’ of structures and the Metropolis Monte Carlo search is driven to move down hill towards the deepest depression on the landscape, "the Global Minimum”, representing the structure with a simulated powder pattern that is the best match with the experimental data.

All global optimisation techniques are susceptible to becoming trapped in local minima, [40] a problem only avoided implicitly by the grid search method. For the Monte Carlo search, the value S and maximum step size are initially set high, so that the search may easily escape from minima. These values are often reduced as the calculation progresses, until the child structures have a 40% acceptance rate. [14] The search is allowed to proceed for a set number of generations as there is no termination criterion to stop the search once a structure with high fitness has been located. The global minimum cannot be determined with certainty unless S is decreased to 0, because until this point, the search still has some hill climbing ability. However, if S is set to 0, the search will become trapped in the first minima it finds, making this setting impractical.

Some crystal structures that have been solved using the MC search include the organic cocrystal 1,2,3-trihydroxybenzene-HMTA [22] (figure 1.6) and the red polymorph of fluorescein. [41] Although the MC search is capable of solving structures of significant complexity such as an organic cocrystal, a MC search can require the computation of many model structures before the best is located. The Markov chain of the MC search used to solve the structure of 1,2,3-trihydroxybenzene-HMTA was 500,000 structures long.

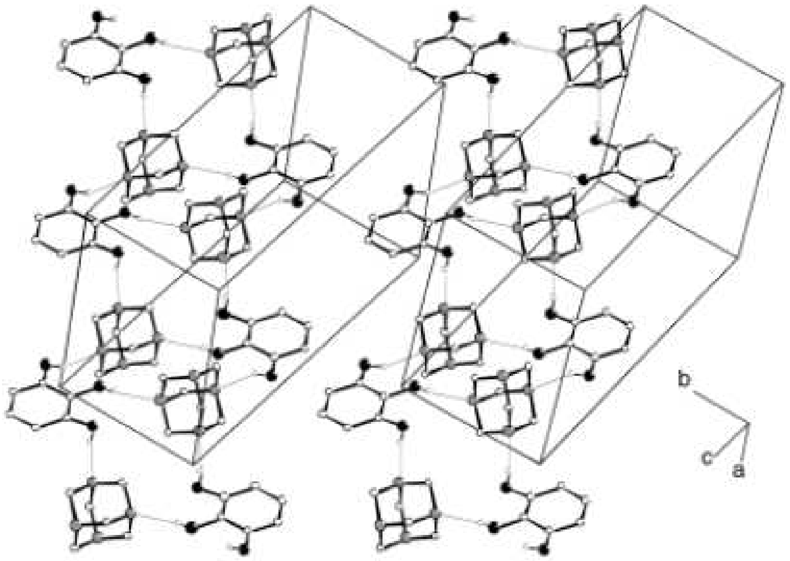


Figure 1.6, Stereoview of the crystal structure of the cocrystal 1,2,3-trihydroxybenzene-HMTA (figure taken from [22]).

1.4.5. Simulated Annealing

A variant of the Metropolis Monte Carlo technique is the simulated annealing (SA) search, in which the value S is decreased automatically, reducing the hill climbing ability of the search as it reaches convergence. [4, 24] The value S (or temperature) is initially set so that the child structures have a 90% survival rate, but reduced so that only 70% of child structures are accepted over their parents. [24] Crystal structures that have been determined using the SA method include: $AlVO_4$, $K_2HCr_2AsO_{10}$ and $[Co(NH_3)_3CO_3]NO_3.H_2O$, [24] the red polymorph of tetrahexylsexithiophene, [42] the pigment p-haematin (figure 1.7), [43] lactose and paracetamol [44] and the crystal structure of form B of famotidine. [29] The SA search is potentially more efficient than the MC search but it can still require many model structures to be generated and evaluated before the best structure is located. The Markov chain of the SA search used to solve the crystal structure of the red polymorph of tetrahexylsexithiophene, described by 13 structure parameters, was 1,200,000 structures long.

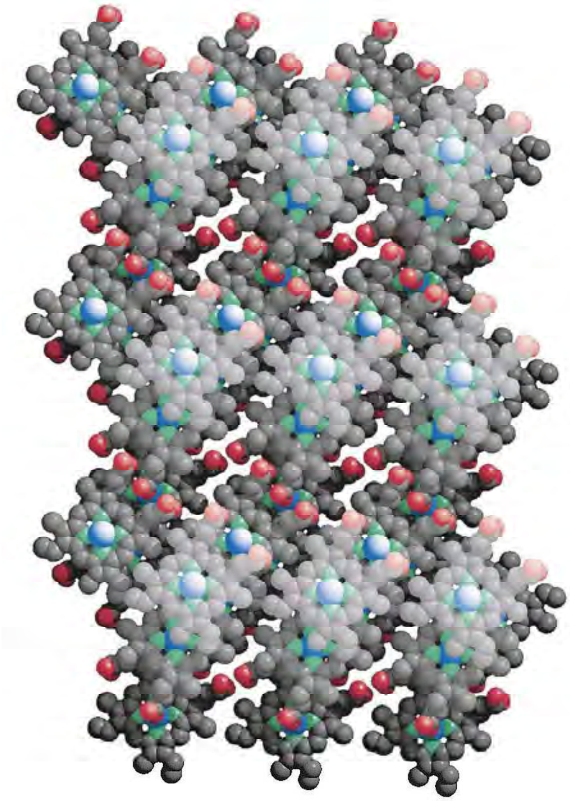


Figure 1.7, The crystal structure of the pigment p-haematin (figure taken from [43]).

1.4.6. Parallel Tempering

The landscape representing particularly complex crystal structures with many degrees of freedom may have numerous local minima besides the global minimum. The existence of the local minima frustrates the search process because the search will explore many local minima before it locates the global minimum. Initially assigning a high value to the temperature parameter of the MC or SA searches encourages the search to initially explore the whole landscape and increases the probability that the search escapes local minima and locates the global minimum. Decreasing the temperature either manually as in the MC search or automatically in the SA search encourages the search to only explore the minima and increases the probability that the search terminates inside the global minimum. However if the temperature is reduced too rapidly the search can become trapped in local minima and fail to reach the global minimum. Parallel tempering, [45] a variant of the MC and SA search, maintains the ability to explore the whole landscape during a single search progression whilst simultaneously the parallel tempering search is encouraged to thoroughly explore any minimum that is located.

The parallel tempering search involves performing multiple MC searches simultaneously, however each individual search is assigned a different temperature. Periodically the individual searches exchange structures. Thus if a low temperature search becomes trapped in a local minimum the structure can be released if it is exchanged into a search operating at a higher temperature. Conversely if a search operating at a high temperature locates a deep minimum it is unlikely that the structure will fully descend into the minimum before the search selects a structure with a lower fitness value and moves to a different area of the landscape. However if the structure is exchanged into a search operating at a lower temperature the structure is more likely to remain in the minimum, facilitating a more thorough exploration of the minimum. Although parallel tempering requires multiple individuals to simultaneously explore the same landscape, it can be a more computationally efficient search than the MC or SA searches. [45]

1.4.7. Constrained searches

A technique used to rein in MC or SA searches so that the search only explores areas of the landscape near the global minimum is to coarsely map the electron density of the unit cell to a chosen resolution, for example 2.5Å, and then to orientate the molecule within this volume of high electron density. [46] This technique not only reduces the area of landscape to be searched, but also greatly constrains the orientation of the molecule as it now has to fit into the volume of high electron density, which is based on the molecular shape, restricts intramolecular rotations and the overall molecular orientation. When this structure envelope approach was applied to a complex peptide, [46] nine of the seventeen intramolecular rotations were constrained to 60° (rather than the full 360), greatly reducing the amount of landscape needed to be explored by the simulated annealing search.

1.5. Population based searches

For all the guidance that can be given to grid search, MC and SA methods, the fact that only one individual is exploring the landscape at any one time means that they are inherently slow searches. As the initial structure is created with random parameters, the search will more than likely start far from the global minimum with its progression towards the global minimum frustrated by the presence of hills and local minima. A search technique that placed a number of individuals upon a landscape, allowing them to communicate data with each other concerning the relative merits of their surroundings, would not only sample a far larger area of landscape for a given time period, but would also put the relevance of any minima found into perspective, as the relative position of the minima on the landscape could immediately be calculated, thus avoiding unnecessary exploration of all but the deepest minimum. Evolutionary algorithms do all this, but concomitantly are able to move their populations of individuals across the landscape to cluster around likely sites for the global minimum without the individuals having to navigate their own way there.

1.5.1. Genetic Algorithms

The processes of evolution and survival of the fittest are harnessed in genetic algorithms GAs, [47] in which a population of individuals is created and allowed to breed, with little supervision, so that the process of natural selection will guide the population towards creating ideal individuals. [48,49] GAs are more complex than the previous global optimisation methods discussed in this chapter, and are more efficient when used to solve complicated search problems. [50] When applied to the problem of direct space structure solution, each member of the population represents a possible structural conformation, each with a fitness value assessed by the cost function and a position on the fitness landscape. This means that the genetic algorithm carries out a "parallel search" [49,51] in which many dissimilar conformations may be investigated simultaneously, unlike the MC and SA searches which consider only one individual and move from one conformation to the next. The MC and SA methods have only one starting point and one landscape explorer which must travel to the global minimum for success: the population based design of genetic algorithms means that the landscape explorer can move quickly to explore likely areas of the landscape. In this way, both the speed of the GA search, and the chance that an initial structure is near the global minimum, is greatly increased in comparison. The fact that GAs usually evolve populations containing more individuals than the total number of individuals used in a typical parallel tempering search means that GAs are potentially more computationally efficient than parallel tempering searches.

The GA creates child structures from the initial population by "cross-over" - the sharing of genetic information between a number of parent individuals, and "mutation" - random variation to the genetic information of an individual. These operations have opposing effects; whilst cross-over encourages homogeneity in the population in which individuals congregate around a single point, mutation forces individuals to be different, effectively pushing the search back out to explore more of the landscape to look for a better solution to the problem. Once a new population of children has been created, the algorithm will perform a selection process on the original parents and the new children, based upon their fitness, so that the best individuals proceed into the next generation, and in turn create the next generation.

There are many ways in which the processes of population member selection, parent selection, mutation and genetic representation can be performed; these will now be discussed.

1.5.1.1. Genetic Representation

In a similar way to the methods discussed previously, the parameters that describe and quantify the degrees of freedom of a model used for structure solution from powder diffraction data define the input for the algorithm. The translation, orientation and intermolecular rotation parameters are stored as a data string often referred to as a chromosome. [48,52] It is worth noting that the way in which the chromosome is written can have a significant impact on the efficiency of the search process. [48] Traditional GAs use binary encoding for the chromosome, but this introduces "Hamming cliffs" [48] in the problem parameters. These occur when real numbers of similar value are represented by very different binary numbers; for example seven (binary 0111) and eight (binary 1000). This large binary difference means that it is unlikely for a chromosome to undergo small mutations near the end of the search, as a parameter with the value of seven may not easily be changed to eight. Thus the search can frequently overstep the correct solution by consistently making excessively large variations to individuals. It is therefore far more effective where possible, to represent the chromosome as real numbers, [47] preventing the occurrence of "Hamming cliffs" and allowing small perturbations to individuals and overall a more efficient search.

1.5.1.2. Parent Selection

Crossover is the process of sharing genetic material between parents to produce children. In doing so, individuals communicate genetic information to each other and reflect the relative quality of the area of the fitness landscape that they inhabit. [51] As a child structure formed from two fit parents is more likely to be fitter than a child formed from one fit and one unfit parent, the search can quickly move towards the better structures by selection of the fitter children to become the parents in the next generation. This also excludes unfit individuals from parenthood so that the search only explores preferred areas of the landscape. If all individuals could be equally selected for parenthood, poor quality individuals would be chosen as often as fit individuals and the search process would become less efficient. [48]

1.5.1.2.1. Roulette Wheel Selection

For Roulette Wheel selection, the fitness of all individuals is "normalised" [48] in proportion to the fittest member of the population. This is done by dividing the fitness value of each individual by the maximum fitness value and creating a distribution of values ranging between zero and one. An individual is then selected at random, and becomes a parent if it has a normalised fitness value equal to or larger than a random number between zero and one. [48,49,52] Therefore the fitter an individual is, the more chance it has of becoming a parent. Population diversity is not unnecessarily reduced during the initial stages of the search, allowing a wide area of the landscape to be explored, as even unfit individuals have a chance of being selected. However, reduction in population diversity does become an issue when the search reaches a local minimum in the landscape when many individuals are clustered together with similar gene and fitness values.

1.5.1.2.2. Tournament Selection

Tournament selection is not as susceptible to the influence of extremely fit or unfit individuals. Rather than comparing one individual against the whole population, pairs of individuals are selected to compete against each other. [52] Each pair is chosen at random so that all individuals, even unfit ones, have a chance of becoming parents. As only the fittest of the pair is selected as a parent, the exclusion of the poorest individual from the next generation is assured. [48]

1.5.1.3. Elitism

Population growth would be an inherent feature of genetic algorithms, as the children created by crossover do not directly replace the parents. To avoid rampant population growth, individuals from the current population can be culled in order to maintain a given population size; this process is referred to as elitism. [49,53] A selection of the best individuals, both parents from previous generations and children, are ranked in order of fitness but only a portion of the best are chosen to go on to the next generation. Elitism therefore not only keeps down population size, but provides only the best individuals for the parent selection process. As only the best individuals are chosen to enter the potential parent population, the overall quality of fitness of the population will not decrease from generation to generation. [49,54]

1.5.1.4. Crossover

Child structures can be created by splicing together genetic material from the parents in a number of ways. The simplest and most applicable for individuals with a small number of genes is single point crossover. [49] Each parent chromosome is cut at the same place and the halves are swapped between the parents, producing two children. For direct space structure solution, this crossover process can be implemented by swapping over the translational and rotational genes of the parents. In this example (figure 1.8), two models of rigid molecules undergo single point crossover: each model structure has three translational degrees of freedom X, Y and Z and three rotational degrees of freedom represented by T, F and S.

This process produces two children from one pair of parents, causing inefficiencies if that particular pair is picked more than once. To increase the number of different children that can be produced from a given pair, two point crossover can be implemented, in which

Parent (A) X1,Y1,Z1,T1,F1,S1 Parent (B) X2,Y2,Z2,T2,F2,S2

--------------Crossover--------------

Child (A) X1,Y1,Z1,T2,F2,S2 Child (B) X2,Y2,Z2,T1,F1,S1

Figure 1.8, Single-Point Crossover

each chromosome is randomly cut at two places and the intervening genes exchanged. [38] To improve single point crossover the two ends of each chromosome string can be joined together, forming two chromosome loops. In this way many different children are produced when corresponding gene sets are randomly swapped between parents as the chances of splicing the same gene for each pair is greatly decreased. [38]

The GA now has a system for selecting individuals that represent good solutions to the problem, by combining genetic material from parents with high fitness, to create better children and moving towards the global minimum. However, by swapping parameters from only good parents, the GA is susceptible to becoming trapped in local minima. When all selected individuals have similar parameters representing a solution to a non-global minimum, the search "stagnates" [52,54] as any variation in the children created during crossover from these individuals merely explores the landscape within the minimum.

To escape from a local minimum, an individual must develop enough genetic variation so that it no longer occupies this area of the landscape. Clearly the creation of an outlier individual requires some other process than crossover, hence GAs need the ability to mutate the genetic material of individuals.

1.5.1.5. Mutation

Mutation involves the active alteration of the genes of an individual by either a completely random "static mutation", [49] or a more guided "dynamic mutation" approach. [48, 50, 52] The individual chosen for mutation can either be a new child structure [51] or an existing parent. [50] However, mutation often produces unfit individuals, so all mutated individuals are automatically placed into the next generation without undergoing any selection process: [49,50,55] this ensures transmission of the new genetic material.

Individual X1,Y1,Z1,T1,F1,S1

-----------Mutation-----------

Mutant X1,Y(JJ),Z1,T1,F1,S1

Figure 1.9, Mutation of an Individual.

For a model of a rigid molecule with three translational and three rotational degrees of freedom (using the notation in previous sections), Figure 1.9 demonstrates the system undergoing mutation.

1.5.1.5.1. Static mutation

Static mutation is more appropriate for initial generations and involves the change of one or more genes in the selected individual by a randomly generated value. [50] Although simple to implement, mutants can be created far from their parent individual, which towards the end of a search progression, means that the search is continuously widened rather than being allowed to converge on the optimal solution.

1.5.1.5.2. Dynamic mutation

Dynamic mutation involves the change of one or more of the parental genes by a random but limited step size to produce the child. [50] Ideally, a scaling mutation function is used, that will produce large mutation variations for initial generations encouraging exploration across the landscape but produce small mutation variations as the population converges permitting careful exploration around minima. Thus the genetic diversity of a population remains high during initial generations, reducing the probability that the population becomes trapped in local minima, whilst only small mutations occur when the population is close to converging on the global minimum. To achieve an efficient scaling mutation function for Genetic Algorithms, it is necessary to use an annealing function similar to that used in a Simulated Annealing search. [48] Unfortunately, to do this, it is necessary to have an idea of how many generations are likely to be required to get the search in the general region of the global minimum. This behaviour is addressed by the autonomous scaling search of the Differential Evolution algorithm (chapter 2).

Crystal structures that have been solved using genetic algorithms include: the structure of a polypeptide, [56] a lead pyridine-3,4 dicarboxylate complex, [57] ortho-thymotic acid [58] and a cocrystal formed between benzoic acid and pentafluorobenzoic acid, containing four independent molecules in the asymmetric unit (figure 1.10). [59]

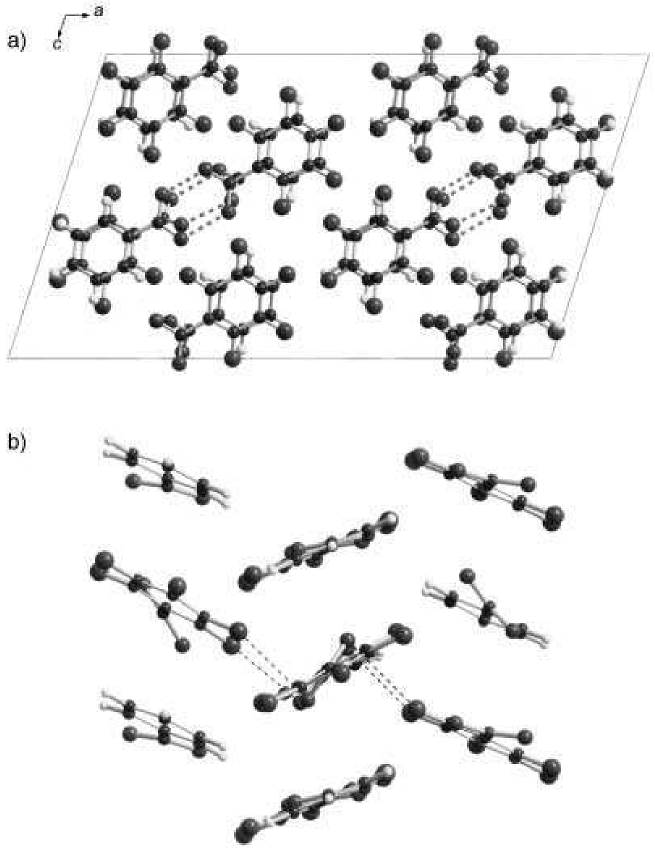


Figure 1.10, The structure of the cocrystal formed between benzoic acid and pentafluorobenzoic acid (figure taken from [59])

1.6. Differential evolution

Like GAs, differential evolution (DE) is a population based search method that applies the natural processes of mating, mutation and natural selection to solve an optimisation problem. However unlike GAs that use crossover and mutation functions to combine genetic material from multiple parents to create child structures, DE is a vector based search. To create a child structure, DE computes the vectors that join a group of individuals on the landscape and combines these vectors to produce a child structure. Thus in differential evolution the processes of crossover and mutation are performed simultaneously over all dimensions or structural parameters, each time a child is created.

This is in direct contrast with GAs that perform crossover and mutation operations separately.

The nature of a vector based search technique means that the crossover and mutation vectors are scaled automatically as the population converges. During the initial generations of a search the genetic diversity of the population is likely to be high causing individuals to be spread widely across the landscape. Thus the vectors joining a group of individuals are likely to be long, resulting in the generation of a child at a considerable distance from the parent. As individuals cluster together the vectors joining a group of individuals become much shorter, resulting in the generation of a child significantly closer to the parent. Thus individuals that are clustered together automatically conduct a local search of the immediately surrounding landscape.

One of the other main features of DE is that it is controlled by a small number of arithmetic functions in contrast to GAs that use probability functions to control the various evolutionary operations: during each generation, DE systematically selects every individual in the population to act as the parent rather than using probability functions to select the fitter individuals in the population. The features of DE and its application to the solution of crystal structures from powder diffraction data will be discussed in detail in the following chapters of this thesis.

1.7. Thesis Overview

- Previous work [60-63] has demonstrated that DE can be used to solve crystal structures from powder diffraction data. DE is governed by a small number of arithmetic expressions and hence a human operator is only required to define the value of a small number of DE control parameters to influence the evolution of a population and therefore the efficiency and reliability of an optimisation. In chapter three, this thesis explores how different combinations of DE control parameters can affect the computational efficiency and the probability that an accurate crystal structure is located by a single DE search. This was done primarily by performing multiple crystal structure determination calculations using different combinations of these DE control parameters deducing the optimal combination of control parameters from these results. The simplicity of DE algorithm also means that it is relatively simple to apply additional evolutionary control strategies to the population. Cultural DE [23, 64] uses information gathered by ancestral individuals during previous generations to influence the evolution of individuals in the current generation. This use of individuals to explore the landscape in order to locate the global minimum and simultaneously gather and pass information about the landscape onto future generations can significantly reduce the number of generations required by a population to converge in the global minimum. This information can be used to discourage current individuals from exploring regions of the landscape which have already been found to represent poor solutions to the problem. Chapter three explores how the computational efficiency of DE and the probability that an accurate crystal structure is located by a single DE search is affected by applying cultural control strategies. The application of different variants of cultural strategy and their effect on the evolutionary process is also explored. Chapter three demonstrates that influencing the evolution of a population by applying cultural guidance in an arbitrary manner can significantly reduce the efficiency of cultural DE.

- The size of the landscape representing a particular optimisation problem is defined by the number of structure parameters needed to describe the crystal structure. Thus when larger populations are initialised inside a particular landscape the landscape will have a higher population density than when a smaller population is initialised. The higher population density increases the probability that an initial individual is close to the global minimum. The reduced distance between this initial best individual and the global minimum reduces the number of generations required for evolutionary processes to move the best individual into the global minimum. Thus larger populations can locate the optimal structure in fewer generations than smaller populations. However, each generation every individual in the population is systematically selected to act as parent and produce a child structure. Thus the number of child fitness evaluations computed during the evolution of a population increases as the size of the population increases. Evaluating the fitness of an individual is a significantly computationally demanding procedure, thus the “rate determining step” of DE is the evaluation of the child structures. As a result, larger populations have a higher computational demand and take longer to converge in real time than smaller populations. Chapter four explores the concept of “eugenic” DE. To increase population density and the probability that an initial individual is near the global minimum, the eugenic DE initially explores the landscape with a large population. However to reduce the computational demand the eugenic DE reduces the size of the population by pruning out the most unfit individuals as the search converges.

- If X-ray powder diffraction is used to record data for a multiphasic sample (containing different crystalline phases), peaks resulting from each of the different crystal structures will be observed in the diffraction pattern. The relative intensity of a peak corresponding to one phase is proportional to the relative abundance of that phase in the mixture of phases. [65-67] If the crystal structures are known, structural data for each phase can be combined to simulate a multiphasic powder diffraction pattern. Rietveld refinement can then be used to refine the relative intensities of peaks corresponding to the different phases and hence determine the relative abundance of each of the phases. [66-70] This technique for quantitative phase analysis has been applied to various industrial applications where the necessary crystal structure data is freely available.

  Previously, iron was produced by mixing and crushing appropriate amounts of iron ore, calcium carbonate and coke into a coarse ‘lumpy’ mixture which was fed into a blast furnace. However, owing to the depletion of ‘lump’ iron ore and the necessary use of ‘fine’ ores this method of production is becoming impractical. [68] In order to process fine ore, appropriate amounts of fine ore and calcium carbonate are combined, finely crushed and ‘baked’ at high temperature producing a physically strong sinter which can be crushed into suitable lumps that can be mixed with coke and fed into a blast furnace. Silicoferrites of calcium and aluminium are an important sinter bonding phase and their composition, structural type and texture greatly affect the physical properties of the sinter which in turn impact the production of iron. In situ X-ray powder diffraction experiments have been used [68] to study the mechanism of formation of the silicoferrite phase in order to determine the combination of reaction conditions that produce the best sinter.

  Bauxite, (the chief ore from which aluminium is extracted) is a mixture of minerals including gibbsite $Al(OH)_3$, boehmite AlOOH, kaolinite $Al_2Si_2O_8(OH)_4$, hematite Fe_2O_3 and goethite FeOOH. Bauxite is processed in the Bayer process [69] to produce alumina (Al_2O_3) which when molten yields aluminium via electrolysis. The Bayer process involves dissolving bauxite in concentrated sodium hydroxide solution in the temperature range 423-523 K producing a solution of sodium aluminate and a suspension of insoluble iron oxides and oxyhydroxides. The insoluble material is precipitated in settling tanks and the clarified liquor is cooled, precipitating gibbsite which is calcined to produce alumina. However, iron oxides and oxyhydroxides seed premature gibbsite precipitation in the settling tanks and the formation of various aluminium containing scales on the process equipment. This reduces the amount of useful gibbsite that can be collected and impedes the flow of liquor through the equipment as well as the transfer of heat to and from the liquor, reducing the efficiency of the Bayer process. In situ X-ray diffraction experiments [69] have been used to study the mechanism of goethite seeded gibbsite precipitation. An understanding of this reaction could be used to optimise the reaction conditions of the Bayer process, minimising gibbsite losses.

Portland cement is an important building material and various chemical reactions occur during its production and use. [71] Initially appropriate quantities of limestone, quartz sand, shale and iron oxide [70] are combined and heated in a kiln at ca 1650 K producing a clinker material. The clinker is composed of four major phases; alite Ca_3SiO_5, belite Ca_2SiO_4 aluminate $Ca_3Al_2O_6$ and ferrite Ca_2AlFeO_5. [72] The clinker is combined with gypsum and additional limestone in a mill [70] to produce Portland cement. Knowledge of the phase composition of the clinker and cement can be used to predict the physical properties of the cement [70,71] as well as providing information about the efficiency of the manufacturing process. Knowledge of the clinker and cement phase composition can be used to control production such as kiln temperature, kiln and mill retention time and the feed rate of clinker and gypsum into the mill. [70] Previously, analytical techniques such as the Bogue method [73] were used to analyse cement composition. However this method involves a lengthy analytical procedure [70] which prevents the acquired knowledge being used to control the manufacturing process in real-time. X-ray powder diffraction combined with Rietveld refinement has been used to analyse the clinker and cement phase composition. [70,71,74] The advantage of an in situ X-ray powder diffraction/Rietveld refinement procedure [70] is that analytical results can be obtained within minutes and used in real time to control the manufacturing process.

However, if a multiphasic powder pattern is recorded and the crystal structures are unknown it is not possible using the Rietveld method [66,67] to perform quantitative phase analysis. Similarly, if the abundance of each phase in the mixture is unknown it is not possible using current techniques to perform crystal structure determination. Chapter five explores the technique of simultaneous direct space crystal structure solution of two different crystalline phases in a bi-phasic mixture using the DE technique and simultaneous quantitative phase analysis by Rietveld refinement.

- Finally, appendix (C) explores coevolution and how coevolutionary strategies can be applied to global optimisation algorithms to increase their efficiency at locating the optimal solution to a problem. [75,76] A number of strategies that could be used to create a cooperative coevolutionary DE suitable for solving the structure of molecular crystals are identified and discussed.

References

[1] N. V. Phadnis, R. K. Cavatur and R. Suryanarayanan. Identification of Drugs in Pharmaceutical Dosage Forms by X-ray Powder Diffractometry. *J. Pharm. Biomed. Anal.* (1997). **15**. 929.

[2] L. Brammer. Developments in Inorganic Crystal Engineering. *Chem. Soc. Rev*. (2004). **33**. 476.

[3] M. Tremayne. The Impact of Powder Diffraction on the Structural Characterization of Organic Crystalline Materials. Phil. *Trans. R. Soc. Lond. A*. (2004). **362**. 2691.

[4] K. D. M. Harris, M. Tremayne and B. M. Kariuki. Contemporary Advances in the Use of Powder X-Ray Diffraction for Structure Determination. *Angew. Chem. Int. Ed.* (2001). **40**. 1626.

[5] C. Sun and D. J. W. Grant. Influence of Crystal Structure on the Tableting Properties of Sulfamerazine Polymorphs. *Pharm. Res.* (2001). **18**. 273.

[6] N. Blagden. Crystal Engineering of Polymorph Appearance, the case of Sulphathiazole. *Powder Tech.* (2001). **121**. 46.

[7] O. Almarsson and M. J. Zaworotko. Crystal Engineering of the Composition of Pharmaceutical Phases. Do Pharmaceutical Co-crystals Represent a New Path to Improved Medicines? *Chem. Commun.* (2004). 1889.

[8] K. Kachrimanis, M. Rontogianni and S. Malamataris. Simultaneous Quantitative Analysis of Mebendazole Polymorphs A-C in Powder Mixtures by DRIFTS Spectroscopy and ANN Modeling. *J. Pharm. Biomed. Anal.* (2010). **51**. 512.

[9] M. M. de Villiers, R. J. Terblanche, W. Liebenberg, E. Swanepoel, T. G. Dekker and M. Song. Variable-Temperature X-ray Powder Diffraction Analysis of the Crystal Transformation of the Pharmaceutically Preferred Polymorph C of Mebendazole. *J. Pharm. Biomed. Anal.* (2005). **38**. 435.

[10] C. B. Aakeroy, S. Forbes and J. Desper. Using Cocrystals to Systematically Modulate Aqueous Solubility and Melting Behavior of an Anticancer Drug. *J. Am. Chem. Soc.* (2009). **131**. 17048.

[11] S. A. Nelson. Tulane University, Department of Earth & Environmental Sciences. (2010). http://www.tulane.edu/~sanelson/eens211/x-ray.htm

[12] K. D. M. Harris and E. Y. Cheung. How to Determine Structures When Single Crystals Cannot be Grown: Opportunities for Structure Determination of Molecular Materials using Powder Diffraction Data. *Chem. Soc. Rev.* (2004). **33**. 526.

[13] G. Reck, R. G. Kretschmer and L. Kutschabsky. POSIT, a Method for Structure Determination of Small Partially Known Molecules from Powder Diffraction Data. *Acta. Cryst. A.* (1988). **44**. 417.

[14] K. D. M. Harris, M. Tremayne, P. Lightfoot and P. G. Bruce. Crystal Structure Determination from Powder Diffraction Data by Monte Carlo Methods. *J. Am. Chem. Soc.* (1994). **116**. 3543.

[15] K. D. M. Harris and M. Tremayne. Crystal Structure Determination from Powder Diffraction Data. *Chem. Mater*. (1996). **8**. 2554.

[16] W. I. F. David and K. Shankland. Structure Determination from Powder Diffraction Data. *Acta. Cryst. A.* (2008). **64**. 52.

[17] C. Giacovazzo. Direct Methods and Powder Data: State of the Art and Perspectives. *Acta. Cryst. A.* (1996). **52**. 331.

[18] S. Y. Chong and M. Tremayne. Development of Novel Evolutionary Algorithms for Crystal Structure Determination from Powder Diffraction Data. School of Chemistry, University of Birmingham UK., (2006).

[19] R. J. Cernik, A. K. Cheetham, C. K. Prout, D. J. Watkin. A. P. Wilkinson and B. T. M. Willis. The Structure of Cimetidine (C10H16N6S) Solved from Synchrotron-radiation X-ray Powder Diffraction Data. *J. Appl. Cryst*. (1991). **24**. 222.

[20] F. C. Chan, J. Anwar, R. Cernik, P. Barnesc and R. M. Wilson. Ab initio Structure Determination of Sulfathiazole Polymorph V from Synchrotron X-ray Powder Diffraction Data. *J. Appl. Cryst.* (1999). **32**. 436.

[21] A. Altomare, R. Caliandro, C. Giacovazzo, A. Grazia, G. Molitcrni and R. Rizzi. Solution of Organic Crystal Structures from Powder Diffraction by Combining Simulated Annealing and Direct Methods. *J. Appl. Cryst*. (2003). **36**. 230.

[22] M. Tremayne and C. Glidewell. Direct-Space Structure Solution from Laboratory Powder Diffraction Data of an Organic Cocrystal, 1,2,3-trihydroxybenzene. *Chem. Comm*. (2000). 2425.

[23] S. Y. Chong and M. Tremayne. Combined Optimisation using Cultural and Differential Evolution Application to Crystal Structure Solution from Powder Diffraction Data. *Chem. Comm*. (2006). 4078.

[24] A. A. Coello. Whole-Profile Structure Solution from Powder Diffraction Data using Simulated Annealing. *J. Appl. Cryst*. (2000). **33**. 899.

[25] H. M. Rietveld. A Profile Refinement Method for Nuclear and Magnetic Structures. *J. Appl. Cryst*. (1969). **2**. 65.

[26] L. B. McCusker, R. B. Von Dreele, D. E. Cox, D. Louer and P. Scardi. Rietveld Refinement Guidelines. *J. Appl. Cryst.* (1999). **32**. 36.

[27] N. Masciocchi. P-Riscon, A Real-Space Scavenger for Crystal Structure Determination from Powder Diffraction Data. *J. Appl. Cryst*. (1994). **27**. 426.

[28] G. E. Engel, S. Wilke, O. K. Nig, K. D. M. Harris and F. J. Leusen. Powdersolve, a Complete Package for Crystal Structure Solution from Powder Diffraction Patterns. *J. Appl. Cryst.* (1999). **32**. 1169.

[29] K. Shankland, L. McBride, W. I. F. David, N. Shankland and G. Steel. Molecular, Crystallographic and Algorithmic Factors in Structure Determination from Powder Diffraction Data by Simulated Annealing. *J. Appl. Cryst*. (2002). **35**. 443.

[30] K. Shankland, A. J. Markvardsen, C. Rowlatt, N. Shankland and W. I. F. David. A Benchmark Method for Global Optimization Problems in Structure Determination from Powder Diffraction Data. *J. Appl. Cryst*. (2010). **43**. 401.

[31] J. P. M. Lommerse, W. D. S. Motherwell, H. L. Ammon, J. D. Dunitz, A. Gavezzotti, D. W. M. Hofmann, F. J. Leusen, W. T. M. Mooij, S. L. Price, B. Schweizer, M. U. Schmidt, B. P. van Eijck, P. Verwer and D. E. Williams. A Test of Crystal Structure Prediction of Small Organic Molecules. *Acta. Cryst. B*. (2000). **56**. 697.

[32] B. R. van Eijck, A. L. Spek, W. T. M. Mooij and J. Kroon. Hypothetical Crystal Structures of Benzene at 0 and 30 kbar. *Acta. Cryst. B.* (1998). **54**. 291.

[33] H. Putz, I. C. Schon and M. Jansen. Combined Method for Ab Initio Structure Solution from Powder Diffraction Data. *J. Appl. Cryst*. (1999). **32**. 864.

[34] Z. Pan, E. Y. Cheung, K. D. M. Harris, E. C. Constable and C. E. Housecroft. A Case Study in Direct-Space Structure Determination from Powder X-ray Diffraction Data: Finding the Hydrate Structure of an Organic Molecule with Significant Conformational Flexibility. *Cryst. Gro. Des*. (2005). **5**. 2084.

[35] S. M. Woodley. Prediction Of Crystal Structures using Evolutionary Algorithms and Related Techniques. Applications of Evolutionary Computation in Chemistry. (Ed. R. L. Johnston). Struct. Bond., Springer-Verlag. Heidelberg. (2004).

[36] V. Brodski, R. Peschar and H. Schenk. Organa, A Program Package for Structure Determination from Powder Diffraction Data by Direct Space Methods. *J. Appl. Cryst*. (2005). **38**. 688.

[37] E. Dova, A. F. Stassen, R. A. J. Driessen, E. Sonneveld, K. Goubitz, R. Peschar, J. G. Haasnoot, J. Reedijk and H. Schenk. Structure Determination of the $[Fe(Teec)_6](Bf_4)_2$ Metal Complex from Laboratory and Synchrotron X-Ray Powder Diffraction Data with Grid-Search Techniques. *Acta. Cryst. B*. (2001). **57**. 531.

[38] H. M. Cartwright. An Introduction to Evolutionary Computation and Evolutionary Algorithms. Applications of Evolutionary Computation in Chemistry. (Ed. R. L. Johnston). Struct. Bond., Springer-Verlag. Heidelberg. (2004).

[39] 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*. (1953). **21**. 1087.

[40] B. V. Babu and S. A. Munawar. Differential Evolution Strategies for Optimal Design of Shell-and-Tube Heat Exchangers. *Chem. Eng. Sci*. (2007). **62**. 3720.

[41] M. Tremayne, B. M. Kariuki and K. D. M. Harris. Structure Determination of a Complex Organic Solid from X-Ray Powder Diffraction Data by a Generalized Monte Carlo Method: The Crystal Structure of Red Fluorescein. *Angew Chem*. (1997). **109**. 188.

[42] M. A. Neumann, C. Tedesco, S. Destri, D. R. Ferro and W. Porzio. Bridging the Gap, Structure Determination of the Red Polymorph of tetrahexylsexithiophene by Monte Carlo Simulated Annealing, First-principles DFT Calculations and Rietveld Refinement. *J. App. Cryst*. (2002). **35**. 296.

[43] S. Pagola, P. W. Stephens, D. S. Bohle, A. D. Kosar and S. K. Madsen. The Structure of Malaria Pigment P-haematin. *Nature*. (2000). **404**. 307.

[44] A. Altomare, R. Caliandro, C. Cuocci, C. Giacovazzo, A. G. G. Moliterni, R. Rizzi and C. Platteau. Direct Methods and Simulated Annealing: a Hybrid Approach for Powder Diffraction Data. *J. App. Cryst*. (2008). **41**. 56.

[45] V. Favre-Nicolin and R. Cerny. FOX, 'Free Objects for Crystallography', a Modular Approach to ab initio Structure Determination from Powder Diffraction. *J. Appl. Cryst.* (2002). **35**. 734.

[46] S. Brenner, L. B. McCusker and C. Baerlocher. The Application of Structure Envelopes In Structure Determination from Powder Diffraction Data. *J. Appl. Cryst*. (2002). **35**. 243.

[47] Y. Zeiri. Prediction of the Lowest Energy Structure of Clusters using a Genetic Algorithm. *Phys. Rev. E*. (1995). **51**. 2769.

[48] A. P. Englelbrecht. Computational Intelligence: An Introduction. John Wiley and Sons LTD. Chester. (2002).

[49] 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. Cryst. A*. (1998). **54**. 632.

[50] K. D. M. Harris, R. L. Johnston and S. Habershon. Applications of Evolutionary Algorithms in Structure Determination from Diffraction Data. Applications of Evolutionary Computation in Chemistry. (Ed. R. L. Johnston). Struct. Bond., Springer-Verlag. Heidelberg. (2004).

[51] S. Darby, T. V. Mortimer-Jones, R. L. Johnston and C. Roberts. Theoretical Study Of Cuau Nanoalloy Clusters using a Genetic Algorithm. *J. Chem. Phys*. (2002). **116**. 1536.

[52] R. L. Johnston. Evolving Better Nanoparticles: Genetic Algorithms for Optimising Cluster Geometries. *Dalton Trans*. (2003). 4193.

[53] E. Landree, C. Collazo-Davila and L. D. Marks. Multi-Solution Genetic Algorithm Approach To Surface Structure Determination using Direct Methods. *Acta. Cryst. B*. (1997). **53**. 916.

[54] C. Roberts, R. L. Johnston and N. T. Wilson. A Genetic Algorithm for the Structural Optimisation of Morse Clusters. *Theor. Chem. Acc*. (2000). **104**. 123.

[55] A. Rapallo, G. Rossi, R. Ferrando, A. Fortunelli, B. C. Curley, L. D. Lloyd, G. M. Tarbuck and R. L. Johnston. Global Optimisation of Bimetallic Cluster Structures [I] Size-Mismatched Ag-Cu, Ag-Ni, and Au-Cu Systems. *J. Chem. Phys*. (2005). **122**. 194308.

[56] S. Habershon, K. D. M. Harris and R. L. Johnston. Development of a Multipopulation Parallel Genetic Algorithm for Structure Solution from Powder Diffraction Data. *J. Comp. Chem*. (2003). **24**. 1766.

[57] Z. J. Feng and C. Dong. GEST, a Program for Structure Determination from Powder Diffraction Data using a Genetic Algorithm. *J. Appl. Cryst*. (2007). **40**. 583.

[58] 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. Lett*. (1997). **280**. 189.

[59] D. Albesa-Jove, B. M. Kariuki, S. J. Kitchin, L. Grice, E. Y. Cheung and K. D. M. Harris. Challenges in Direct-Space Structure Determination from Powder Diffraction Data, A Molecular Material with Four Independent Molecules in the Asymmetric Unit. *Chem Phys Chem*. (2004). **5**. 414.

[60] D. E. McRee. Differential Evolution for Protein Crystallographic Optimizations. *Acta. Cryst. D*. (2004). **60**. 2276-2279.

[61] M. Tremayne, C. C. Seaton and C. Glidewell. Structures of Three Substituted Arenesulfonamides from X-ray Powder Diffraction Data using the Differential Evolution Technique. *Acta. Cryst. B.* (2002). **58**. 823.

[62] C. C. Seaton and M. Tremayne. Differential Evolution, Crystal Structure Determination of a Triclinic Polymorph of Adipamide from Powder Diffraction Data. *Chem. Comm.* (2002). 880.

[63] S. Y. Chong, C. C. Seaton, B. M. Kariuki and M. Tremayne. Molecular Versus Crystal Symmetry in Tri-substituted Triazine, Benzene and Isocyanurate Derivatives. *Acta. Cryst. B.* (2006). **62**. 864.

[64] M. Tremayne, S. Y. Chong and D. Bell. Optimisation of Algorithm Control Parameters in Cultural Differential Evolution Applied to Molecular Crystallography. *Front. Comput. Sci. China*. (2009). **3**. 101.

[65] A. W. Hull. A New Method of Chemical Analysis. *J. Am. Chem. Soc*. (1919). **41**. 1168.

[66] R. J. Hill and C. J. Howard. Quantitative Phase Analysis from Neutron Powder Diffraction Data using the Rietveld Method. *J. Appl. Cryst.* (1987). **20**. 467-474.

[67] D. L. Bish and S. A. Howard. Quantitative Phase Analysis using the Rietveld Method. *J. Appl. Cryst*. (1988). **21**. 86.

[68] N. V. Y. Scarlett, I. C. Madsen, M. I. Pownceby and A. N. Christensen· In situ X-ray Diffraction Analysis of Iron Ore Sinter Phases. *J. Appl. Cryst.* (2004). **37**. 362.

[69] N. A. S. Webster, I. C. Madsen, M. J. Loan, R. B. Knott, F. Naim, K. S. Wallwork and J. A. Kimpton. An Investigation of Goethite-Seeded Al[OH]3 Precipitation using In Situ X-ray Diffraction and Rietveld-Based Quantitative Phase Analysis. *J. Appl. Cryst.* (2010). **43.** 466.

[70] N. V. Y. Scarlett, I. C. Madsen, C. Manias and D. Retallack. On-Line X-ray Diffraction for Quantitative Phase Analysis, Application in the Portland Cement Industry. *Powder Diff*. (2001). **16.** 71.

[71] L. Leon-Reina, A. G. De la Torre, J. M. Porras-Vazquez, M. Cruz, L. M. Ordonez, X. Alcobe, F. Gispert-Guirado, A. Larranaga-Varga, M. Paul, T. Fuellmann, R. Schmidt and M. A. G. Arand. Round Robin on Rietveld Quantitative Phase Analysis of Portland Cements. *J. Appl. Cryst*. (2009). **42**. 906.

[72] L. P. Aldridge. Accuracy and Precision of Phase Analysis in Portland Cement by Bogue, Microscopic and X-ray Diffraction Methods. *Ceme. Conc. Res*. (1982). **12**. 381.

[73] R. H. Bogue. Calculation of the Compounds in Portland Cement. *Ind. Eng. Chem*. (1929). **1**. 192.

[74] F. Guirado and S. Gali. Quantitative Rietveld Analysis of CAC Clinker Phases using Synchrotron Radiation. *Ceme. Conc. Res*. (2006). **36**. 2021.

[75] M. A. Potter and K. A. De Jong. A Cooperative Coevolutionary Approach to Function Optimization. Third Parallel Problem Solving From Nature. Jerusalem, Israel. (1994). 249.

[76] Z. Yang, H. Tang and X. Yao. Large Scale Evolutionary Optimization using Cooperative Coevolution. *Info. Sci.* (2008). **178**. 2985.

Chapter 2 Methodology

2.1 Crystal Structure Determination from Powder Diffraction Data

The process of crystal structure determination from powder diffraction data can be broken down into three distinct primary activities: indexing of the powder pattern, structure solution and structure refinement. [1] The three stages are carried out sequentially, thus the success of the latter stages depends on the success of the preceding ones. [2] Figure 2.1 shows a schematic diagram of the process of crystal structure determination.

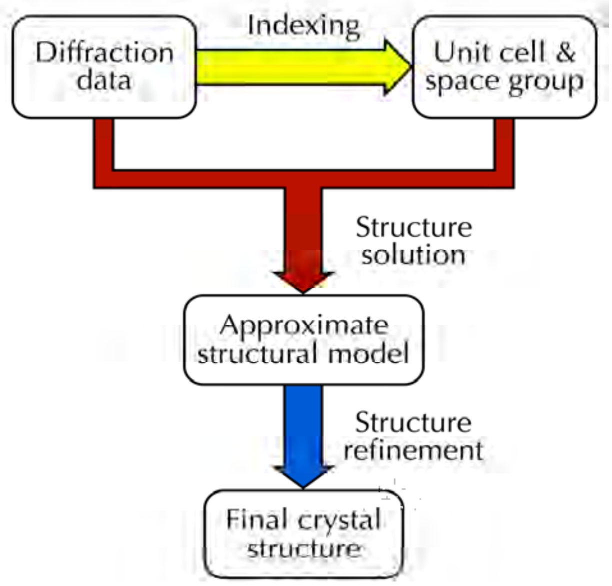


Figure 2.1, The Stages of Crystal Structure Determination. (Figure taken from [3]).

2.1.1 Preparation for Structure Solution

2.1.1.1 Indexing

Indexing involves determination of the unit cell (or lattice) parameters *a*, *b*, *c*, *alpha*, *beta* and *gamma* by analysis of the peak positions in the diffraction pattern. Nowadays indexing is largely performed automatically, relying on computer programs such as ITO, [4] TREOR, [5] DICVOL [6] and Crysfire [7] (which brings together many of the programs in the form of a suite) to identify potential lattice parameter solutions for a powder diffraction pattern. These programs usually require a minimum of 20 non-overlapped diffraction peaks to generate possible reliable unit cells. However, due to the peak overlap intrinsic to powder diffraction, indexing of powder patterns can often be difficult, and as it is not possible to perform structure solution and structure refinement until the pattern has successfully been indexed, severe peak overlap which prevents indexing can make determination of the crystal structure impossible.

The indexing of a powder diffraction pattern can also be treated as a global optimisation problem. A genetic algorithm [8] has been successfully used to index powder diffraction patterns. The GA generates a population of possible unit cells. Unit cells are generated with random lattice parameters and the value of the parameters define the chromosomes of each unit cell. A powder diffraction pattern is simulated for each unit cell and compared using a cost function based on R_{wp} with the powder pattern recorded for the real sample. In this way the fitness of each unit cell in a population is assessed. The unit cell that is assigned an *R* factor with the lowest value represents the current best model unit cell. By evolving the population of unit cells the GA can optimise the lattice parameters and determine a more realistic unit cell. Because a cost function based on R_{wp} is used to compare simulated and real powder patterns peak overlap is taken into account. Thus this technique can be used to index powder diffraction patterns that display considerable peak overlap.

2.1.1.2 Determination of Space Group

The space group is determined by identifying peaks that are absent from the diffraction pattern. If groups of peaks are systematically absent from the pattern it suggests that the crystal structure adopts a specific space group or choice of space groups. [1] Knowledge of the space group can be used to determine the contents (the number of independent molecules) in the asymmetric unit and symmetry elements that can aid the structure solution. If a molecule is located on a ‘special’ position (for example a molecular centre of inversion coincides with a crystallographic centre of inversion) the molecule has no translational degrees of freedom and only half the structure need be solved as the structure of the other half can be inferred through symmetry. Determination of the space group reduces the complexity of the structure solution even when the molecule is not located on a ‘special' position. If the unit cell contains multiple asymmetric units, structure solution can be used to locate the molecule within one asymmetric unit and the position of the molecules in the other asymmetric units determined through symmetry.

2.1.1.3 Profile fitting

Once the lattice parameters and space group have been determined, the Le Bail profile fitting technique [9] or Pawley fitting approach [10] is used to prepare the intensity data needed to solve the crystal structure. The purpose of the Le Bail technique is to fit (generate a mathematical description of) the diffraction pattern by refinement of profile variables that describe the pattern. These variables include peak position (defined by lattice and zero point parameters), background intensity distribution, peak width and peak shape and peak intensity. Unlike the structure refinement process where peak intensity is generated by atomic positions, the Le Bail technique treats these as variables. The values of the lattice parameters determined by the indexing procedure are initially used to fit the pattern, however, as the Le Bail fitting procedure considers peak shapes as well, it determines the lattice parameters with greater accuracy, thus the subsequent direct space structure solution stage uses these more accurate values to generate the unit cell. Once a good Le Bail fit (usually indicated by an R_{wp} factor of < 10%) has been achieved, the values of the refined lattice and profile parameters can be used in the structure solution stage to simulate accurate powder diffraction patterns for each computer generated model of a crystal structure. In traditional crystal structure solution approaches, this stage also generates a set of extracted integrated intensities through the peak fitting routine. These intensities are then used to generate a structure solution through direct methods or can be used to calculate chi-squared through direct-space methods.

2.1.2 Structure solution

The purpose of crystal structure solution is to develop a sufficiently accurate model of the crystal structure that can be refined, during the structure refinement stage, to a good representation of the real crystal structure.

The Le Bail fit is achieved without placing any scattering matter inside the unit cell and hence the *R* factor assigned to the Le Bail fit indicates how well that pattern can be fitted. The *R* factor assigned to the Le Bail pattern will be less than the *R* factor assigned to a model that is a good representation of the real crystal structure, but will be significantly less than the *R* factor assigned to a model that is a poor representation of the real structure. Thus the *R* factor assigned to a model found during the structure solution stage can be used to indicate the quality of the model.

Due to the peak overlap in powder diffraction patterns it is not always possible to identify one group of systematically absent peaks and assign a unique space group to the crystal structure. [1,2] If multiple space groups are possible, it is necessary to perform the structure solution stage with each possible space group in order to determine which space group gives the best model.

Once a good Le Bail fit has been achieved, scattering matter is introduced into the refined unit cell through the structure solution process. In the work discussed in the following chapters, structure solution is performed using the DE algorithm implemented in the POSSUM package [11] to generate model structures, and the Rietveld refinement [12] feature of the GSAS package [13] to evaluate the models.

The POSSUM package uses the provided information about the molecular connectivity to assemble the appropriate number and type of atoms into a model structure. This model is generated and sometimes optimised in terms of molecular geometry within ChemOffice Chem 3D. [14] Although in reality sigma and pi bonds are of different lengths, during this model generation step one ‘standard’ bond length is specified. Thus for example the standard C-C bond length used in POSSUM = 1.6Å, the standard C-O bond = 1.45Å and the standard C-N bond = 1.55Å. This model is placed in a random position and orientation, and if appropriate, conformation inside the unit cell. The individual parameters quantifying the degrees of freedom describing the model are assembled into a chromosome string and stored in the differential evolution program. The Rietveld refinement application of GSAS is then used to simulate a powder diffraction pattern for the model and compare it with the real powder pattern. The *R* factor value is then associated with the chromosome string and represents the fitness of that model.

The POSSUM package continues to generate models in this way until the differential evolution has stored a population containing the designated number of models, each with an associated fitness value. Once the population contains the required number of models, evolution of the population is initiated and continues until convergence of the population and completion of the optimisation so that each model in the population has an *R* factor with the same value. At this point it is assumed that all model structures are identical and that the optimum structure solution has been found. This crystal structure can then be used as a starting point in the structure refinement stage to determine the final real crystal structure. However, this final stage of the process is often only pursued if the model makes structural ‘sense’ (if intermolecular distances or intramolecular torsions are as expected), if the model has a reasonable *R*-factor (based on the Le Bail fit value) and a similar optimum structural model has been located in more than one DE run.

2.1.3 Structure refinement

During structure refinement, the variables defining the simulated powder diffraction profile and structural variables defining the model are refined to achieve an optimal fit between the diffraction pattern simulated for the model and the pattern recorded for the real sample. The fit between the simulated and real patterns is refined using the least squares technique. [12] The two patterns are digitised so that the difference between the simulated and real intensities at each point is calculated. The $\mathrm{R_{wp}}$ cost function is commonly used to quantify the difference between the simulated and real patterns using the following equation.

$$R_{\mathrm{wp}} = \frac{\sum w(I_{\mathrm{obs}} - I_{\mathrm{calc}})^2}{\sum w(I_{\mathrm{obs}})^2} \tag{2.1}$$

During least squares minimisation, the variables defining the simulated profile and structural model are adjusted to minimise the total difference in the intensities at each point in the corresponding real and simulated patterns. In this way the model is refined to a more accurate representation of the real crystal structure. [1,2,12] Although the fit between the simulated and real patterns is commonly quantified by $\mathrm{R_{wp}}$ [12] other *R*-factors can be used such as $\mathrm{R_B}$. [15] Some packages that can be used to perform Rietveld refinement include; GSAS, [13] RIETAN, [16] PROFIL, [17] FULLPROF, [18] DBW [19] and TOPAS. [20]

The structural variables refined during this least squares minimisation are not the same variables optimised during the structure solution approach used in this work. During structure solution, interatomic bond lengths and bond angles are fixed at predetermined values and the model is treated, where possible, as a collection of rigid fragments that are moved about the unit cell. During structure refinement, the constraints on bond lengths and geometry are relaxed so that the model can be refined to actual real values of the structure under consideration. [1, 2] This relaxation has a major effect on the improvement of the profile fit, but often relies on restraints being used in the structure refinement. [21] Despite this significant effect on the fit, it is still more efficient to reduce the number of variables in the structure solution process by consideration of rigid fragments where possible, although a combination of the two has been used. [22] This structure determination strategy uses a genetic algorithm to perform structure solution and Rietveld refinement to refine each model in the population after each generation. The GA explores the R_{wp} fitness landscape using a population of model structures and controls the evolution of the population using conventional (crossover, mutation and Darwinian survival of the fittest) operations. However, refinement of the parameters defining each model after each generation allows each model to locate the bottom of the nearest local minimum after each generation. Once a model has been refined, it is placed back into the population where it can contribute genetic information to the rest of the population in the next generation. This evolutionary strategy represents Lamarckian evolution. "In the Lamarckian concept of evolution, characteristics that are acquired by an individual in the course of its lifetime can be passed on to its offspring. In Darwinian evolution on the other hand, the genetic characteristics passed on by a parent to its offspring are those that the parent itself possessed when it was born". [22] Thus a model does not necessarily locate the global minimum by evolutionary processes alone. As the refinement of a model allows it to reach the bottom of the nearest local minimum, a model evolved by the GA that locates the top of the global minimum can reach the bottom in one Lamarckian step rather than relying on evolutionary processes to move the model to the bottom of the global minimum (which is likely to take multiple generations). This combined structure solution and structure refinement strategy can potentially determine a crystal structure in fewer generations than a GA using conventional crossover, mutation and Darwinian survival of the fittest operations.

The ability of Rietveld refinement to determine an accurate crystal structure relies on the accuracy of the variables used to fit the powder profile and the model located during structure solution. In general terms, as long as half the scattering matter in the structural model located by structure solution is within 1Å of the position of the atoms in the real structure, Rietveld refinement is expected to be successful. [23] However, if the model is a poor representation of the real crystal structure, the refinement can become trapped in a local minimum during the least squares refinement or 'explode' as structural variables are refined to incorrect values. As discussed above, geometric constraints are often used to prevent the model from adopting implausible conformations and to ensure that atomic intramolecular distances are realistic. This reduces the probability that the least squares minimisation is trapped in a local minimum or destroys the model and increases the probability of successful refinement.

As protons are weak X-ray scatterers, the location of hydrogen atoms in the crystal structure has limited influence on the profile of the diffraction pattern. During structure solution, the small difference between the simulated and experimental diffraction patterns caused by the incorrect location of hydrogen atoms is obscured by the considerable difference between these patterns caused by the incorrect location of the non-hydrogen atoms within the structural model. Thus if the cost function used to evaluate model structures during structure solution is based only on profile fitting such as R_{wp}, it is difficult to determine the position of hydrogen atoms in the crystal structure. As a result, structure solution is often performed using incomplete models from which many of the hydrogen atoms have been removed. However, if the structure solution locates a model which is a good representation of the real crystal structure, the non-hydrogen atoms will be in roughly the correct location, thus the difference between simulated and experimental patterns caused by the incorrect location of hydrogen atoms becomes more obvious. During structure refinement, the hydrogen atoms that have been excluded from the structure solution can be returned to the model and their correct position in the crystal structure determined.

2.2 Differential Evolution

Differential Evolution is a vector based global optimisation algorithm that self regulates the area of landscape searched during each generation according to the extent of convergence of the population. Unlike GAs (see section 1.5) in which parameter values (chromosomes) of selected individuals are combined to create child structures, DE uses the "differences" between chromosomes possessed by individuals to create a child. [24] Therefore, as the population converges on the global minimum and the genetic differences between individuals decrease, the area of landscape in which children are created decreases concomitantly. Furthermore, the DE algorithm does not use probability functions to control evolutionary operations such as mutation and parent selection, instead the population of individuals is evolved using a small number of arithmetic operations. This makes it easier to find the optimal combination of parameters used to control the operation of DE [25] and so DE is more easily adapted to the optimisation of a variety of problems than GAs which are usually more complex to implement and control. In DE, ‘child’ solutions are created from parent ‘solutions’ using two main arithmetic operations. These operations are; a) recombination (which is analogous to the crossover operation used by GAs) and b) mutation. These two arithmetic operations are each controlled using a separate control parameter, recombination K and mutation F. Since K and F are each independently assigned a value by the user between zero and one the user can adapt the optimisation to suit different kinds of problems by using different combinations of K and F.

Differential evolution has been used in a wide variety of applications including optimisation of the design of digital filters, [26] industrial heat exchangers, [27] chemical reaction conditions [28,29] and in the determination of the crystal structure of proteins [30] and the crystal structure of disordered crystals. [31]

2.2.1 Defining the Structural Model

In this implementation of DE [11] (used to solve the structure of molecular crystals from powder X-ray diffraction data), the first procedure is the generation of a population of model crystal structures. Each model is defined by a set of structure parameters each of which is assigned an appropriate but random value. Parameters that correspond to crystallographic fractional coordinates used to define the position of the model inside the unit cell are each assigned a random value between zero and one. Parameters that define the overall orientation of the model inside the unit cell are each assigned a random value between zero and 360°. If the molecular structure possesses intramolecular flexibility, torsion parameters are used to define the relative orientation between the rigid molecular fragments. If steric factors do not hinder the rotation of these fragments, or prior knowledge of potential conformation is not applied, each torsion parameter is assigned a random value between zero and 360°. If the rotation of the fragments is sterically hindered or prior knowledge is available, this can be incorporated into the structural model by limiting the range of values of the torsion parameters.

2.2.2 Controlling Population Size

The performance and efficiency of DE is significantly affected by the dimensionality of the landscape. [25, 32] When DE is used to solve a crystal structure, the dimensionality of the problem is determined by the number of parameters needed to define a model. A rigid model (such as that representing the structure of benzene) possessing three translation and three orientation parameters would generate a landscape with six dimensions. The addition of a rigid functional group (such as the double ring system of baicalein) would increase the total number of structure parameters and landscape dimensions to seven (as shown in Figure 2.2). For reasons that are demonstrated in the next chapter (chapter 3) it is sometimes necessary to use larger population sizes to successfully optimise complex problems which require description by a greater number of structure parameters. As the complexity of a problem is largely dependent on the number of structure parameters to be optimised, it is logical to regulate the size of a population by the number of structure parameters. In this implementation of DE the size of a population (*NP*) is defined as a simple multiple of the number of structure parameters. Results discussed in chapter 3 show that for most searches to have a high probability of solving a crystal structure a search should use a population containing at least 10 times as many models as parameters defining the model. Thus a crystal structure solution that requires a model defined by seven parameters should be performed by a search using a population size of 7 x 10 = 70 models.

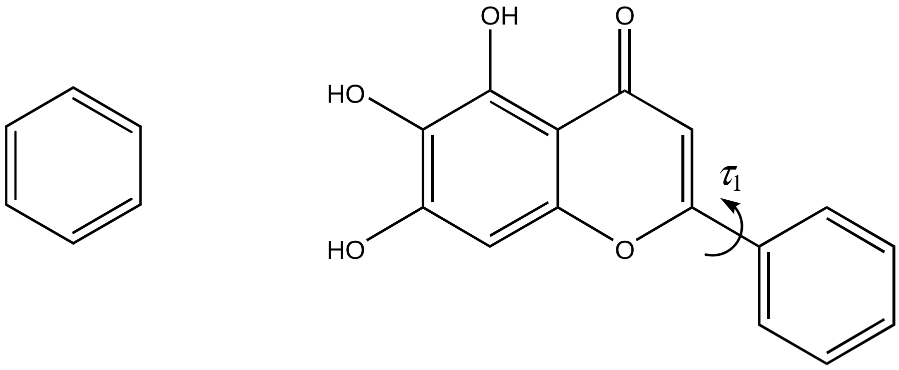


Figure 2.2, Structural models of benzene (left) and baicalein (right).

2.2.3 Recombination and mutation parameter

The recombination parameter *K* and the mutation parameter *F* are assigned values between zero and one. (It is not necessary for *K* and *F* to have the same value). Previous work [11] has shown that a *K* of one causes a search to converge rapidly but this can increase the probability that the search converges prematurely in a local minimum. However, assigning K a value of 0.99 reduces the probability of premature convergence whilst maintaining a fast rate of convergence. The mutation constant *F* is usually assigned a value in the range 0.4-0.7 for this application of structure solution. Increasing the value of *F* reduces the probability that a search converges prematurely in a local minimum but increases the number of generations required by a search to converge.

2.2.4 The search

The search proceeds in a series of generations. During each generation, each model is systematically selected to act as a parent *P*, and each parent produces one child, *C*, per generation. To reduce the rate at which the genetic diversity of a population is lost, three new random individuals, *R1*, *R2* and *R3*, are selected in each case to assist each new parent in the creation of a child.

A child (or trial) is created by the sum of a pair of vectors joining the parent and the three randomly selected individuals (as represented by figure 2.3). The first vector of the pair represents "recombination" of genetic material from two individuals and is calculated from the parent *P* to the first randomly selected individual *R1*. The second vector represents "mutation" of that genetic material to ensure that the child is sufficiently different from its parent to avoid premature convergence. The mutation vector is calculated from the second randomly selected individual *R2* to the third randomly selected individual *R3*. The child is then defined using the following expression.

$$\text{trial} = \text{parent} + K(\text{random}_1 - \text{parent}) + F(\text{random}_2 - \text{random}_3) \qquad (2.2).$$

In this way, a recombination vector (first term) and a mutation vector (second term) are calculated. The direction of the recombination vector is determined by the difference between *P* and *R1*, and its length determined or scaled by the value of the recombination parameter *K*. The direction of the mutation vector is determined by the difference between *R2* and *R3*, and its length scaled by the value of the mutation parameter *F*. The child is created in a multi-dimensional space by simultaneous variation of all the structural parameters used to define the

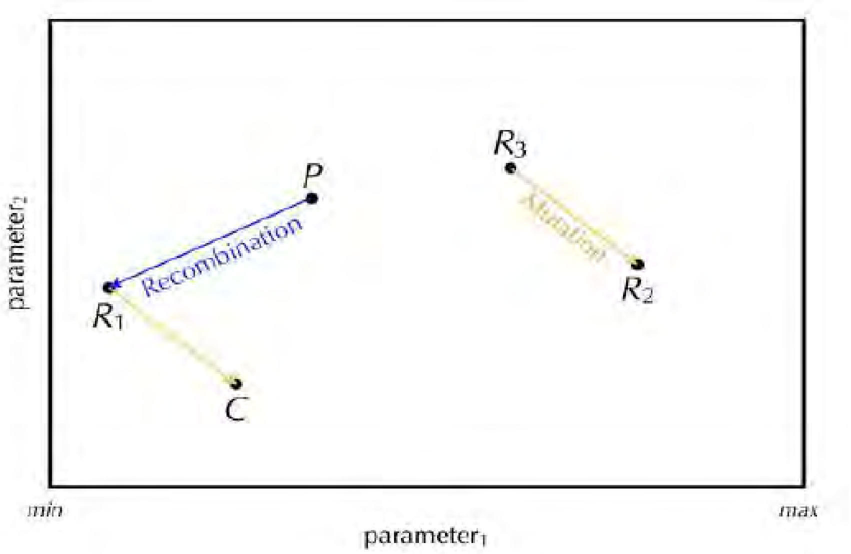


Figure 2.3, Representation of a parent *P* and three random individuals *R1*, *R2* and *R3* in a two-dimensional landscape creating a child *C*. The figure shows that the parent and random 1 individual are joined by a recombination vector and the remaining random 2 and 3 individuals are joined by a mutation vector. (Figure taken from [3]).

parent and three random models through addition of these two vectors (scaled using *K* and *F* respectively) from the parent position.

The fitness of the child is assessed by the cost function (in this work R_{wp}), and if R_{wp} child - R_{wp} parent <= zero, the child immediately replaces the parent as an active (breeding) member of the population. Hence, as soon as a "better" solution to a problem is found, it can immediately contribute genetic information to the rest of the population. [3] This is another way in which DE differs from GAs in which a population is updated with better solutions only at the end of each generation. If R_{wp} child - R_{wp} parent > zero the child is discarded and the parent retained.

The length of the recombination and mutation vectors are determined by the distance between the models selected as *P*, *R1*, *R2* and *R3*. As models cluster together the distance between four models selected as *P*, *R1*, *R2* and *R3* decreases. Thus, although the values of *K* and *F* remain constant throughout a search, the length of the recombination and mutation vectors decrease automatically as a search progresses. During the initial generations of a search there is a high probability that models selected as *P*, *R1*, *R2* and *R3* will be spread over a considerable area of the landscape and that the recombination and mutation vectors will be long. As a result a child is likely to be created at a considerable distance from the parent. If the child is fitter than the parent the search can rapidly move to a distant part of the landscape by accepting the child. This increases the probability that a search initially explores a significant area of the landscape and locates the global minimum. During latter generations of a search (when many models are clustered around the global minimum), there is a high probability that models selected as *P*, *R1*, *R2* and *R3* will be spread over a much smaller area of the landscape. Consequentially the recombination and mutation vectors joining four models selected as *P*, *R1*, *R2* and *R3* will be significantly shorter. This increases the probability that a child is created much nearer the parent. If the child is fitter than the parent it indicates that the child is likely to be nearer the global minimum than the parent and the search can move slightly nearer the global minimum by accepting the child. Thus without changing the values of *K* and *F*, a DE search can initially rapidly explore a significant area of the landscape and once models cluster around the global minimum the search can conduct a detailed exploration of the global minimum and locate the optimal solution.

2.2.5 Terminating a search

A search continues until a termination criterion is reached. Ideally a search is terminated automatically when all the model structures converge on the global minimum. The model that is located at the global minimum is the model structure that produces the simulated diffraction pattern most like the pattern recorded for the real crystal structure. Thus the model located at the global minimum is the best possible representation of the real crystal structure that can be found by structure solution. The successful convergence of the population on a single point is indicated by all models being assigned an *R* factor of the same value. However if a search fails to locate the global minimum (solve the crystal structure) in a convenient number of generations, the search can be aborted by using the control parameter *Gmax*. *Gmax* can be used to specify the maximum number of generations allowed for a search. If a number of individuals become irrevocably trapped in different local minima it is likely that a search will not converge and hence will fail to solve the crystal structure. If this happens a search will be aborted after *Gmax* number of generations.

If a search is conducted using an insufficient rate of mutation it is likely that the genetic diversity of a population will decrease rapidly and cause the search to converge prematurely in a local minimum. Even though all the models are assigned an *R* factor of the same value it does not necessarily indicate that the optimal solution has been located. In order to prove that a search has converged in the global minimum and that a model that has been assigned an *R* factor with a particular value is the optimal solution it is necessary to perform multiple DE runs to solve one crystal structure. As the models in the initial population are generated at random, initiating multiple DE runs decreases the probability that a search converges in the same minima. Increasing the value of *F* decreases the rate at which genetic diversity of a population is lost and thus decreases the probability that a search converges prematurely in a local minimum. If multiple DE runs using different rates of mutation all converge and locate models that are assigned an *R* factor with the same value it suggests that the models are in fact the optimal solution.

2.2.6 Landscape Boundaries

Since DE is a vector based search, the child structure can be generated in an area of the landscape that is some distance from the four individuals used to create it, or in a portion of landscape that corresponds to parameter values outside the defined limits of the problem. [3,11,27,32] In direct space structure determination, these parameter values may correspond to space outside the unit cell eg less than zero or more than one, rotational space less than zero or more than 360° or, due to steric hindrance, physically impossible conformations of the molecular structure. Although in this crystallographic application, areas outside the original unit cell and the rotational space can be considered as equivalent these sensible boundaries to the landscape maintain control over the parameterisation and enhance efficiency. To maintain the child structures within the landscape, DE invokes "boundary" conditions. This ensures that stray children are placed back into the area of interest i.e the landscape corresponding to the unit cell with the added advantage that sensible limits can be placed on internal torsion parameters so that prior information can be included in the search, without the disruption of pathways between parent and child.

The simplest "reset" approach is to place a child half way between the parameter value of the parent and the boundary value that has been exceeded. [3] This median-point reset function (as represented in figure 2.4) has been found to be more successful than a more complex, scaled reset, in which the distance a child is placed back inside the boundary is proportional to the distance a child exceeds the boundary. [3]

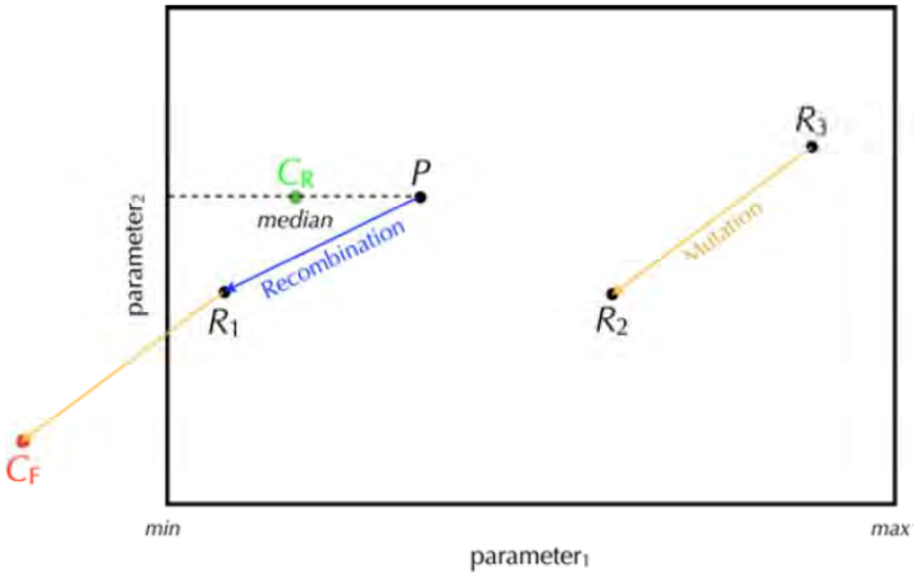


Figure 2.4, A child that has exceeded a certain boundary is replaced half way between the parent and the boundary. (Figure taken from [3]).

The use of fixed boundaries and the inherent rescaling of the recombination and mutation vectors as a search converges leads to passive confinement of a population. However, active confinement, in which the boundaries are driven inwards upon the converging population, can accelerate the rate of convergence. For example, by the reduction of the amount of intramolecular rotational freedom so that children are prevented from inheriting unfavourable structural configurations that have previously been evaluated and rejected. [33] Such active confinement requires the provision of additional boundary information by either previous studies or information provided by the optimisation itself, [33-35] i.e provided by previous generations to tell the DE where in the landscape the model structures are clustering and which boundaries can be most effectively moved to enhance efficiency.

2.3 Cultural Differential Evolution

The notion of guiding the evolution of the individuals in the current generation of a search towards the global minimum by using knowledge acquired by individuals during previous generations has been likened to human culture, [3,26,33-35] where the behaviour of individuals is influenced by social trends, that are set by influential members of the population.

Cultural evolution algorithms maintain two distinct search spaces: population space and belief space. Population space stores the parameters of the individuals used to optimise the problem while belief space stores the behavioural traits of previous generations. An effective cultural algorithm requires a method of transmitting information between the population and belief spaces. This function must specify the effect of an individual's experience on the belief space, and the influence of belief space on the evolution of the population. Thus, both belief space and population space are constantly adapting due to the influence of the other.

In our implementation of cultural differential evolution, [3,33] population space stores the parameters used to define the model structures and belief space stores information about where in the landscape these model structures are located. The information stored in the belief space is used to control the position of the boundaries that define the population space. When the belief space detects clustering of individuals in particular areas of the landscape the position of the boundaries are adjusted to confine the search so that individuals can only explore the areas of the landscape with a high population density. This implementation of CDE is relatively simple compared to the implementation developed by Becerra *et al*. [36] where the belief space is divided into four distinct knowledge sources that influence the population space in different ways and additional functions control which knowledge source(s) have the greatest influence on the population.

In the Becerra *et al*. implementation of cultural differential evolution, [36] belief space is divided into situational, normative, topographical and history knowledge sources. Acceptance and influence functions control how information is transmitted between the population and belief spaces respectively.

The acceptance function is used to control how many individuals in a single generation can transmit information from the population space to the belief space. At the start of a search, a default number of accepted individuals can transmit information to the belief space. As the search progresses, the number of accepted individuals is decreased, however if the best individual does not change for a defined number of generations the number of accepted individuals is increased to the default. At the start of a search the influence function has an equal probability of selecting each of the four knowledge sources to affect the evolution of the population. As the search progresses the influence function selects a knowledge source with greater probability if the knowledge source causes a greater number of parents to be replaced by fitter children relative to the other knowledge sources. In this way, the influence function learns which knowledge sources, if selected, are more likely to cause a search to converge rapidly.

The situational knowledge source is continuously updated with the best individual that has been located by the search. If situational knowledge is used to affect the evolution of a population, the recombination vector (that is used to create a child) is calculated from a parent to the current best individual that is stored in the situational knowledge: referring to equation 2.2 in section 2.2.4 this would mean that *R1* is replaced with the current best individual. In effect, this pushes a child towards the best point on the landscape.

The normative knowledge source uses the landscape boundaries to store information about areas of the landscape where relatively fit individuals are clustering. This knowledge source also controls a scale factor that can be used to scale the mutation function and influence how far a child is created from the parent. At the start of a search, each pair of dimension-specific landscape boundaries are placed in their respective default positions. After each generation, the position of the children accepted to transmit information to the belief space in each dimension in the landscape is collated. If all accepted children are located inside the boundary pair, these boundaries are adjusted inwards, whereas if some children are outside the boundary pair the boundaries are adjusted outwards. The distance between a pair of boundaries is also used to calculate a dimension-specific scale factor used to adjust the mutation function. Thus as a search converges, children cluster on one small area of landscape, the boundaries are adjusted to lie close to either side of the children and therefore a small dimension scale factor will be calculated. During the next generation the mutation vector will be short and hence a child will be created close to the parent and have a high probability of being created inside these boundaries.

The topographical knowledge is used to generate a map of significantly fit individuals in the landscape. Each dimension of the landscape is divided into 'cells' and the position of the best model in each cell is recorded. If the topographical knowledge is selected by the influence function to affect the evolution of the population, children that are generated inside a particular cell are encouraged to move towards the best individual.

The history knowledge stores information about the position of an individual if it remains the best individual for a specified number of generations. If a search fails to locate a better solution within this number of generations, it suggests that the current best individual occupies a minimum. However, without a systematic search of the whole landscape it is not possible to state that the best individual is located in the global minimum. If an individual remains the ’best’ individual for a specified number of generations, the influence function is then used to decrease the probability that children are created near the best individual, thus encouraging the search to explore the landscape further and locate a better solution.

Our implementation of cultural differential evolution (CDE) [3,33] combines the ideas of population and belief space using a less complex approach based loosely on the idea behind the normative knowledge source and the movement of boundaries within the search. In the CDE method used here, areas of the landscape which have a high population density (where models are clustering) are determined by collating the position of the children in the multi-dimensional landscape. Areas of the landscape with a low child population density determined by the ‘population under threshold’, parameter *NUT*, are then removed and the search is prevented from exploring these areas. During each generation, the position of each child on the landscape is recorded regardless of whether the child has a higher fitness value than the parent and accepted, or a lower fitness value than the parent and rejected. At the end of each generation, the CDE determines where in the landscape the children created during the present generation are clustering. The positions of all the children are determined by sorting and analyzing the structure parameters defining each child structure. The position of a child in each dimension of the landscape is defined by a dimension-specific structure parameter. Thus the position of the children in an N-dimensional landscape is stored in an N-dimensional record. After each generation, the values of the structure parameters in each dimension of the record are placed in sequence of increasing value and the sorted values are then placed into histogram bins. [3,33] Figure 2.5 shows the distribution of parameter values used to define the x fractional coordinate of 70 models. The 70 parameter values have been sorted into a total of 22 histogram bins. The maximum and minimum values of the histogram bins at each end of the distribution of parameter values are defined by the current maximum and minimum values of the dimension specific landscape boundaries: thus each bin represents a defined area of one dimension of the landscape. The number of bins is constant throughout a search thus the number of parameters in one bin can be used to determine the child

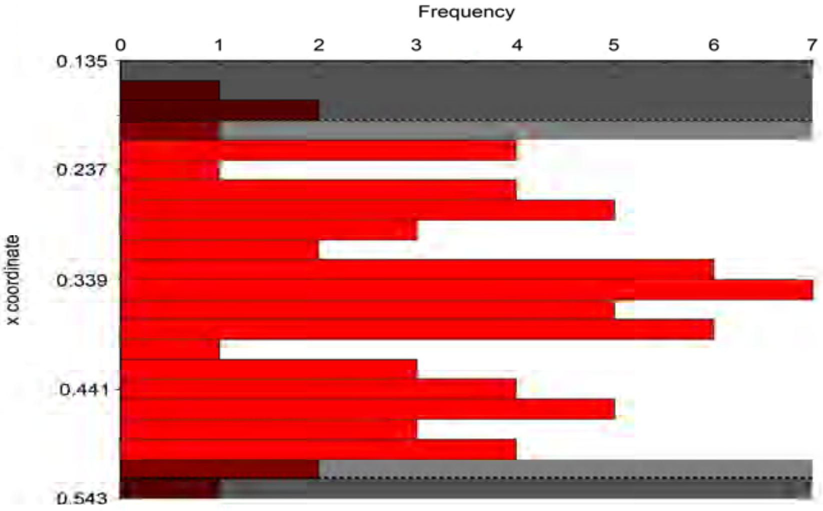


Figure 2.5, the distribution of parameter values representing the x fractional coordinate of 70 model structures across 22 histogram bins. Bins are removed from each end of the distribution until four individuals have been pruned from each end of the distribution. One bin is then reinstated at each end. (Figure taken from [33]).

population density in each area of each dimension. This approach in which the boundaries of each parameter are treated independently is significantly different from the rest of the DE process in which the changes in all the parameters are treated simultaneously.

Bins are removed from each end of the distribution until the number of children that have been removed from each end of the distribution is equal to the value of the user-defined cultural pruning parameter *NUT*. [3,33] One bin is then restored to each end of the distribution to prevent aggressive pruning. The maximum and minimum parameter values of the pruned distribution specify the new maximum and minimum values of the dimension specific landscape boundaries. In this way children generated in the next generation are prevented from exploring areas of the landscape in which (during the current generation), few children clustered. Figure 2.5 shows a distribution of 70 parameter values across 22 histogram bins where the 'under population threshold' *NUT* is equal to 4. Therefore, in this example bins at each end of the distribution are removed until four individuals have been pruned from each end of the distribution. When a population is created, all boundaries are placed in their respective default positions, thus parameters that correspond to crystallographic fractional coordinates used to define the position of the model inside the unit cell are each assigned a random value in the range zero and one. In this example, after only 80 generations, the population space is pruned so that the value of the parameter that corresponds to the crystallographic *x* fractional coordinate can only be assigned

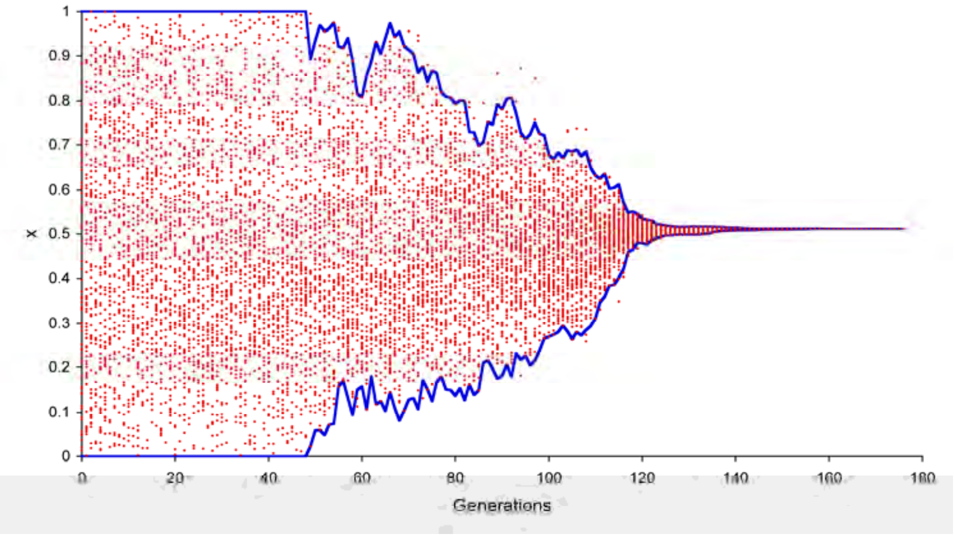


Figure 2.6, The maximum and minimum parameter values of the pruned distribution specify the new maximum and minimum values of the dimension specific landscape boundaries as shown in the blue line. The distribution of x coordinate values is denoted by a red dot for each member of the population. (Figure taken from [33]).

values in the range 0.2-0.7 (figure 2.6).

The record of the positions of the children is deleted after each generation so that each generation is only influenced by the position of the children created during the immediately preceding generation. As a search begins to converge, the parameter values (representing cultural trend setters) cluster together into distinct groups and occupy a small number of histogram bins, leaving social "outliers" to sparsely populate the remaining bins covering the rest of the landscape. Thus, during a search, progressively more bins are removed and a search is increasingly confined to only explore the area of the landscape with a high population density. Increasing the value of *NUT* increases the probability that a greater number of bins are removed at any one time. Thus increasing the value of *NUT* increases the rate at which a search is confined. Confining a search to a smaller area of landscape increases the probability that a search converges after a smaller number of generations.

In this thesis, traditional differential evolution (without cultural guidance) and cultural differential evolution searches have been used to solve previously determined crystal structures. Chapter 3 examines the effect of using different combinations of *F* (mutation), *NP* (population size) and *NUT* (cultural pruning) control parameters on DE and CDE searches used to solve different crystal structures.

2.4 Accelerating Evolution by Eugenic Population Pruning

The results presented in chapter 3 demonstrate that searches using larger population sizes solve the crystal structure (locate the global minimum) in fewer generations than searches using smaller population sizes. This is because larger population sizes create higher population densities in the landscape. For a particular optimisation problem the size of the landscape is determined by the number and range of the parameters defining a structural model. When larger populations are initialised in a particular landscape, the landscape will have a higher population density than when a smaller population is initialised. As these models are generated at random, an increase in the population density results in an increase in the probability that an initial model is generated close to the global minimum. The reduced distance between the initial best model and the global minimum then reduces the number of generations required for the evolutionary processes to optimise this initial best model. However, searches using larger population sizes are significantly more computationally intensive than those using a smaller population size. As every model in the DE population is systematically selected in each generation to produce a child structure, the number of child structure fitness evaluations required increases with the size of the population. As the Rietveld fitness calculation is the most computationally demanding procedure (in this work it is ‘the rate determining step’ of direct space crystal structure solution), a search using a larger population takes significantly longer in real time to solve the crystal structure than a search using a smaller population. In summary, optimisation using a larger population can solve the crystal structure in fewer generations but smaller populations solve the crystal structure in less real time.

In order to address these conflicting requirements for efficiency of the search algorithm, this thesis presents the development and application of the ‘Eugenic’ DE method. The Eugenic DE is a search technique that exploits the greater searching capacity of large populations and the ability of small populations to solve a crystal structure with significantly less computational effort.

Eugenics, noun. ’The science of improving stock, whether human or animal’.
Webster Unabridged Dictionary. (1913).

Eugenics, noun. ’Selective breeding as proposed human improvement’.
Encarta pocket dictionary. Microsoft Corporation. (1999).

In nature, due to predation or disease, the probability that an offspring individual reaches maturity and breeds is low. Thus many more offspring are created than is actually needed for the survival of the species. By initially generating a large population that contains many more models than can be practically evolved to convergence and discarding over 90% of the models that are assigned the lowest fitness, the Eugenic DE mimics natural evolution and survival of the fittest more closely than other search techniques that evolve a population of constant size.

The Eugenic DE technique itself is very simple. A search using a large (primary) population is initially used to explore the landscape and through the production of models with high fitness values, identify deep minima. Once a sufficient number of models with high fitness have been located, the majority of the most unfit models are pruned out of the primary population. This leaves a smaller (secondary) population that contains a high proportion of highly fit individuals that are likely to be close to the global minimum. The secondary population is used to explore the landscape using the traditional DE approach and hence solve the crystal structure. However, the secondary population is now biased towards the area of landscape that contains the fittest individuals, resulting in what should be a more efficient search. A similar approach has been employed, [37] to significantly reduce the number of fitness evaluations calculated during searches based on particle swarm optimisation.

Particle swarm optimisation (PSO) developed in the 1990's [26,38] uses a search technique inspired by the collaborative 'swarm' behaviour of biological populations such as flocks of birds or schools of fish. In PSO, a population of individual 'particles' is generated in a landscape and each particle is 'flown' through the landscape in a series of generations to locate the optimal solution. In each generation, the direction and speed of flight of each individual is calculated using information about the current position of the individual, its position in previous generations and the current position of the best individual. Hence unlike DE and GA techniques, PSO intrinsically uses historical knowledge to influence a search. The position of each individual changes in each generation, hence for a swarm of *NP* individuals that is allowed to evolve for *G* generations, *NP x G* fitness evaluations are calculated during a search for the optimal solution.

Variation of the number of individuals in a population is found to decrease the total number of fitness evaluations calculated by a PSO search. [37] One search technique initially explores a landscape with a small population and increases the number of individuals in the population as the search converges on the optimal solution and the average fitness of the individuals increases.

The second technique initially explores the landscape with a large population and periodically prunes individuals with the lowest fitness from the population. Both search techniques reduce the total number of fitness evaluations during a search by approximately 60% compared to a PSO using a population of constant size. However, the search that increases the number of individuals in a population is significantly less likely to locate the optimal solution than a PSO using a population of constant or decreasing size. [37] Initial exploration of a landscape with a small population increases the probability that a significant proportion of the individuals cluster in a local minimum and are assigned a relatively high fitness value. If the size of the population is then increased as the search progresses by the generation of new individuals in random positions, the new individuals are more likely to be assigned lower fitness values than the individuals clustered in the local minimum. As a result, the new individuals are more likely to travel towards the local minimum rather than explore the landscape and locate the global minimum. The second approach in which the number of individuals in the population is decreased is more likely to locate the optimal solution because the large initial population increases the probability that one or more individuals rapidly locate the global minimum and are assigned a relatively high fitness value. These individuals then attract the rest of the population towards the global minimum. Periodic pruning of the individuals that are assigned the lowest fitness values reduces the number of individuals in the population and hence the total number of fitness evaluations calculated during the remainder of the search.

However, the PSO in which the population size is reduced differs from our eugenic DE in two significant ways: the eugenic DE reduces the size of a primary population in one 'massive pruning event' rather than multiple smaller ones as used by the PSO approach, [37] and the eugenic DE prunes the primary population once a certain portion of the individuals in a primary population are assigned a relatively high fitness value, rather than after an arbitrary number of generations.

The results from this new eugenic DE approach are discussed in chapter 4 and demonstrate that the DE using a eugenic adaptation is able to solve a crystal structure with high probability, whilst calculating the fitness of significantly fewer children than a traditional DE search using a population of constant size. Thus the eugenic DE can solve a crystal structure in significantly less real time than a traditional DE search.

2.5 Simultaneous SOLUTION of Multiple Crystal Structures and Quantitative Phase Analysis from MultiPhasic Diffraction Data.

When a new crystalline material is synthesised, it may not be possible to prepare a pure crystalline sample without any trace of impurity or additional phase. This is often the case in molecular cocrystal engineering. The technique of cocrystal engineering can be used to improve the biopharmaceutical properties of an active pharmaceutical ingredient (API), by cocrystallising the API with other biologically inert crystalline materials. If the individual starting materials are not combined in perfect stoichiometric quantities, either by experimental error or intentionally as a method of forcing the formation of a particular product with a certain stoichiometry, the desired product may crystallise simultaneously with quantities of unreacted starting material or new crystalline bi-products. Lamotrigine (6-(2,3- dichlorophenyl)-1,2,4-triazine-3,5-diamine), is one example of an API with poor physical properties. It is used primarily as an anticonvulsant drug for the treatment of epilepsy, [39] however in the isolated form it displays low solubility which reduces its efficacy as a useful API. In order to increase the solubility of Lamotrigine, attempts were made [39] to synthesis cocrystals of Lamotrigine with other pharmaceutically approved biologically inert compounds. Production of the form [I] of the 1:1 Lamotrigine methylparaben cocrystal by dissolving the two solid components in tetrahydrofuran and leaving the solution to evaporate slowly caused crystallisation of the desired Lamotrigine methylparaben cocrystal concomitantly with methylparaben and a Lamotrigine THF solvate.

When the solventless dry grinding technique [40] is employed to grind together multiple crystalline starting materials; traces of unreacted starting material may disperse throughout the desired cocrystal product. This problem also arises if multiple products in terms of composition, stoichiometry or multiple polymorphic forms recrystallise from the solution. [40-44] The powder diffraction pattern collected from a sample containing multiple crystalline phases necessarily contains multiple sets of diffraction peaks that each result from diffraction by different crystal structures and are superimposed to produce a single ‘mixed’ multiphasic pattern. In some cases the different crystalline materials produce observable diffraction peaks with different and distinctive shapes. [44] In these circumstances it is possible to visually separate these diffraction peaks into sets that each result from diffraction by one of the materials. It is then possible to determine the different crystal structures by straightforward peak exclusion and subsequent use of existing crystal structure determination techniques, although this is often only done when the ‘other’ peaks arise from relatively small amounts of impurities. However it is more common that the different sets of diffraction peaks that each result from diffraction by the different crystal structures will not have distinctive shapes, and that these different sets of diffraction peaks overlap. In such cases, it is not trivial to separate the observable diffraction peaks into distinct phase sets. Since the observed peak spacing will not be compatible with any single unit cell, attempts to index a multiphasic pattern using a single unit cell will fail unless it is possible to identify a discrete set of peaks that correspond to diffraction by one crystalline phase.

Pattern decomposition methods [8,45,46] (discussed further in chapter 5) can be used to sort the observable diffraction peaks into distinct sets that each result from diffraction by one crystal structure. Although these pattern decomposition methods are not always capable of sorting a sufficient number of peaks corresponding to one crystal structure to allow the crystal structure to be determined directly from the pattern, the pattern decomposition methods can identify a sufficient number of peaks to allow the lattice parameters of each of the crystals to be determined.

Despite the peak overlap intrinsic to powder X-ray diffraction, providing the lattice parameters of a crystal structure are known, direct space methods can be used to solve a crystal structure from monophasic powder diffraction data. If a cost function based on R_{wp} is used to evaluate model structures generated in the direct space method, the overlap of peaks in a diffraction pattern can be taken into account and the need to fit individual peak positions and intensities is negated by matching the whole profile shape of simulated and real powder diffraction patterns. Since direct space methods are capable of solving crystal structures from monophasic powder patterns despite the overlap of peaks, direct space methods are potentially capable of solving crystal structures from multiphasic powder diffraction patterns. Chapter 5 of this thesis explores the possibilities of solving one (or two crystal structures simultaneously) from a biphasic powder diffraction pattern. Additionally quantitative phase analysis by Rietveld refinement [47-51] is used to improve the fit between simulated and experimental biphasic powder diffraction patterns.

In a diffraction pattern recorded for a multiphasic sample the intensity of a peak that results from diffraction by one crystal phase relative to the intensity of other peaks that result from diffraction by different crystal phases is proportional to the relative abundance of that crystal phase [47,48] This relationship between the intensity of a peak and the abundance of a crystal phase forms the basis of quantitative phase analysis by Rietveld refinement. As discussed in section 2.1.3 of this thesis, the Rietveld method involves simulating a diffraction pattern for a computer generated model of a crystal structure and quantitatively comparing the simulated pattern with a pattern recorded for a real sample of the crystal. During each cycle of refinement a scale factor that is one variable defining the simulated profile is refined so that simulated diffraction peaks have the same magnitude of intensity as equivalent peaks in the real pattern. If the scale factor was not refined it is unlikely that a diffraction pattern simulated for a model structure that was a good representation of the real crystal structure would match the real diffraction pattern. This would result in a model that was a good representation of the real crystal structure being assigned an *R* factor with a high value, therefore it would be unlikely that the refinement process would determine an accurate crystal structure.

Rietveld refinement can be used to simulate diffraction patterns for multiphasic crystalline materials and by refining a scale factor specific to each crystal phase (so that simulated peaks have the same intensity as equivalent peaks in the real multiphasic pattern), determine the relative abundance of a crystalline phase that is present in a multiphasic material. In order to calculate a phase-specific scale factor it is necessary to know the crystal structure. [51] In the work discussed in chapter 5 of this thesis, model structures generated by the direct space method are used to supply the structural information used in the quantitative phase analysis by Rietveld refinement.

2.6 Coevolution

"Coevolution refers to the simultaneous evolution of multiple populations with coupled fitness", [52] and as such can be considered as a strategy for increasing the efficiency of population-based algorithms. Interactions between individuals of different populations can be either competitive [52, 53] or cooperative. [54-56] In competitive coevolution, an individual in one population competes with other similar individuals in the same population by evolving strategies that allow that individual to exploit characteristics displayed by (competing) individuals in different populations. The individual assigned the highest fitness value in one population will be the most "ruthless exploiter" of the greatest number of competing individuals. In cooperative coevolution, an individual in one population competes with other similar individuals in the same population by evolving strategies that allow the individual to interact synergistically with (cooperating) individuals in different populations. The individual that is assigned the highest fitness value will therefore be the individual that is able to interact most sympathetically with the greatest number of cooperating individuals. The "natural" predator-prey relationship of competitive coevolution has been applied to evolutionary algorithms to investigate game playing strategies. [52-54,57] In these examples, each individual in a particular population represents a particular style of game playing strategy. The fitness value assigned to an individual is determined by the rate of success of an individual at beating individuals in other populations that represent opposing strategies. Cooperative coevolution has moved away from its roots in natural symbiosis and has been applied to global optimisation techniques [54-56,58] where it is used to increase the efficiency of the algorithms.

Although competitive coevolution is not obviously applicable to crystal structure solution, cooperative coevolution has the potential to increase the efficiency of differential evolution applied to crystal structure solution from powder diffraction data by direct space methods. Appendix (C) discusses some of the operations specific to cooperative coevolutionary global optimisation algorithms (CCGOAs) including problem decomposition, collaboration and fitness assignment. The discussion concludes by identifying some evolutionary operators that appear most appropriate for the creation of a hypothetical cooperative coevolutionary differential evolution that could be used to solve crystal structures from powder diffraction data by the direct space method.

References

[1] K. D. M. Harris and E. Y. Cheung. How to Determine Structures When Single Crystals cannot be Grown: Opportunities for Structure Determination of Molecular Materials using Powder Diffraction Data. *Chem. Soc. Rev.* (2004). **33**. 526.

[2] L. B. McCusker, R. B. Von Dreele, D. E. Cox, D. Louer and P. Scardi. Rietveld Refinement Guidelines. *J. Appl. Cryst.* (1999). **32**. 36.

[3] S. Y. Chong and M. Tremayne. Development of Novel Evolutionary Algorithms for Crystal Structure Determination from Powder Diffraction Data. School of Chemistry. University of Birmingham UK. (2006).

[4] J. W. Visser. A Fully Automatic Program for Finding the Unit Cell from Powder Data. *J. Appl. Cryst.* (1969). **2**. 89.

[5] P. E. Werner, L. Eriksson and M. Westdahl. TREOR, a Semi-exhaustive Trial-and-Error Powder Indexing Program for all Symmetries. *J. Appl. Cryst.* (1985). **18**. 367.

[6] A. Boultif and D. Louer. Indexing of Powder Diffraction Patterns for Oow-Symmetry Lattices by the Successive Dichotomy Method. *J. Appl. Cryst.* (1991). **24**. 987.

[7] R. A. Shirley. CRYSFIRE. Suite of Programs for Indexing Powder Diffraction Patterns. University of Surrey.

[8] B. M. Kariuki, S. A. Belmonte, M. I. McMahon, R. L. Johnston, K. D. M. Harris and R. J. Nelmes. A New Approach for Indexing Powder Diffraction Data Based on Whole-profile Fitting and Global Optimization using a Genetic Algorithm. *J. Synchrotron Rad*. (1999). **6**. 87.

[9] A. Le Bail, H. Duroy and J. L. Fourquet. Ab-initio Structure Determination of LiSbWO6 by X-ray Powder Diffraction. *Mater. Res. Bull*. (1988). **23**. 447.

[10] G. S. Pawley. Unit-cell Refinement from Powder Diffraction Scans. *J. Appl. Cryst*. (1981). **14**. 357.

[11] C. C. Seaton and M. Tremayne, POSSUM. Programs for Direct-Space Structure Solution from Powder Diffraction Data. School of Chemistry. University of Birmingham UK. (2002).

[12] H. M. Rietveld. A Profile Refinement Method for Nuclear and Magnetic Structures. *J. Appl. Cryst*. (1969). **2**. 65.

[13] A. C. Larson and R. B. Von Dreele. GSAS: Generalized Structure Analysis System. Manual LAUR 86-748. Los Alamos National Laboratory. Los Alamos. USA. (1986).

[14] ChemOffice Pro 2010. CambridgeSoft, 1 Signet Court, Swanns Road, Cambridge, CB5 8LA.

[15] R. A. Young and D. B. Wiles. Profile Shape Functions in Rietveld Refinements. *J. Appl. Cryst*. (1982). **15**. 430.

[16] F. Izumi, H. Asano, H. Murata and N. Watanabe. Rietveld Analysis of Powder Patterns Obtained by TOF Neutron Diffraction using Cold Neutron Sources. *J. Appl. Cryst*. (1987). **20**. 411.

[17] J. K. Cockcroft. PROFIL, Version 5 17, Department of Crystallography, Birkbeck College, UK, (1994).

[18] J. Rodnguez-Carvajal. in Collected Abstracts of Powder Diffraction Meeting, Toulouse, France, (1990). 127.

[19] D. B. Wiles and R. A. Young. A New Computer Program for Rietveld Analysis of X-ray Powder Diffraction Patterns. *J. Appl. Cryst*. (1981). **14**. 149.

[20] A. Coehlo. (2007). TOPAS-Academic. Version4.1. Coehlo Software. Brisbane. Australia. http://www.topasacademic.net.

[21] C. Baerlocher. “Restraints and Constraints in Rietveld Refinement” in The Rietveld Method. Eds. R.A.Young. *IUCr Monograph on Crystallography*. OUP. (1993). 186-196.

[22] 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. Lett*. (2000). **321**. 183.

[23] K. D. M. Harris and M. Tremayne. Crystal Structure Determination from Powder Diffraction Data. *Chem. Mater*.

(1996). **8**. 2554.

[24] J. Velazquez-Reyes and C. A. Coello. A Comparative Study of Differential Evolution Variants for Global Optimization. Gecco. (2006). 485.

[25] Q. Yang, L. Cai, S. X. Yang and Y. Xue. Differential Evolution using Historical Knowledge. IEEE Xplore. Gran. Comp. (2008). 730.

[26] A. P. Englelbrecht. Computational Intelligence: An Introduction. John Wiley and Sons Ltd. Chester. (2002).

[27] B. V. Babu and S. A. Munawar. Differential Evolution Strategies for Optimal Design of Shell-and-Tube Heat Exchangers. *Chem. Eng. Sci*. (2007). **62**. 3720.

[28] W. Yanling, L. Jiangang and S. Youxian. An Improved Differential Evolution for Optimization of Chemical Process. *Chinese J. Chem. Eng*. (2008). 16. 228.

[29] M. H. Lee, C. Han and K. S. Chang. Dynamic Optimization of a Continuous Polymer Reactor using a Modified Differential Evolution Algorithm. Bid. *Eng. Chem. Res*. (1999). **38**. 4825.

[30] D. E. McRee. Differential Evolution for Protein Crystallographic Optimizations. *Acta. Cryst. D*. (2004). **60**. 2276.

[31] T. Weber and H. B. Biirgi. Determination and Refinement of Disordered Crystal Structures using Evolutionary Algorithms in Combination with Monte Carlo Methods. *Acta. Cryst. A*. (2002). **58**. 526.

[32] Z. Yang, H. Tang and X. Yao. Differential Evolution for High-Dimensional Function Optimization. Proc. 2007 IEEE Cong. Evol. Com. (2007). 3523.

[33] S. Y. Chong and M. Tremayne. Combined Optimisation using Cultural and Differential Evolution Application to Crystal Structure Solution from Powder Diffraction Data. *Chem. Comm*. (2006). 4078.

[34] R. Storn and K. V. Price. Differential Evolution, A Fast and Efficient Heuristic for Global Optimisation Over Continuous Spaces. *J. Glob. Opt*. (1997). **11**. 341.

[35] K. V. Price. New Ideas in Optimization. McGraw-Hill. London UK. (1999). 77-158.

[36] C. A. C. Coello and R. L. Becerra. Cultured Differential Evolution for Constrained Optimization. *Eng. Opt*. (2004). **36**. 219.

[37] B. Soudan and M. Saad,. An Evolutionary Dynamic Population Size PSO Implementation. 3rd Int. Conf. Info. Comm. Tech. from Theo. to Appl. (2008). 620.

[38] J. Kennedy and R. C. Eberhart. Particle Swarm Optimization. IEEE Int. Conf. Neural Net. (1995). 1942.

[39] M. L. Cheney, N. Shan, E. R. Healey, M. Hanna, L. Wojtas, M. J. Zaworotko, V. Sava, S. Song and J. R. Sanchez-Ramos. Effects of Crystal Form on Solubility and Pharmacokinetics, A Crystal Engineering Case Study of Lamotrigine. *Cryst. Growth. Des*. (2010). **10**. 394.

[40] A. V. Trask, J. van de Streek, W. D. S. Motherwell and W. Jones. Achieving Polymorphic and Stoichiometric Diversity in Cocrystal Formation, Importance of Solid-state Grinding, Powder X-ray Structure Determination and Seeding. *Cryst. Growth. Des*. (2005). **5**. 2233.

[41] R. E. Dinnebier, F. Olbrich, S. van Smaalen and P. W. Stephens. Ab Initio Structure Determination of Two Polymorphs of Cyclopentadienylrubidium in a Single Powder Pattern. *Acta. Cryst. B*. (1997) .**53**. 153.

[42] Z. Hugonin, M. Johnsson and S. Lidin. Two for the Price of One, Resolvable Polymorphism in a 'Single Crystal' of a- and $3Sb_3O_4I$. *Sol. Stat. Sci*. (2009). **11**. 24.

[43] T. R. Shattock, P. Vishweshwar, Z. Wang and M. J. Zaworotko. 18-Fold Interpenetration and Concomitant Polymorphism in the 2,3 Co-Crystal of Trimesic Acid and 1,2-Bis(4-pyridyl)ethane. *Cryst. Growth. Des*. (2005). **5**. 2046.

[44] K. F. Bowes, G. Ferguson, A. J. Lough and C. Glidewell. The 1,1 Adduct of triphenylsilanol and 4,4-bipyridyl and Three Pairwise-concomitant Triclinic Polymorphs of the 4,1 Adduct Having Z' = 0.5, 1 and 4. *Acta. Cryst. B*. (2003). **59**. 277.

[45] E. Maccaroni, G. B. Giovenzana, G. Palmisano, D. Botta, P. Volante and N. Masciocchi. Structures from Powders, Diflorasone diacetate. *Steroids*. (2009). 74. 102.

[46] M. Brunelli, J. P. Wright, G. B. M. Vaughan, A. J. Nora and A. N. Fitch. Solving Larger Molecular Crystal Structures from Powder Diffraction Data by Exploiting Anisotropic Thermal Expansion. Angew. *Chem. Int. Ed*. (2003). **42**. 2029.

[47] R. J. Hill and C. J. Howard. Quantitative Phase Analysis from Neutron Powder Diffraction Data using the Rietveld Method. *J. Appl. Cryst*. (1987). **20**. 467.

[48] D. L. Bish and S. A. Howard. Quantitative Phase Analysis using the Rietveld Method. *J. Appl. Cryst*. (1988). **21**. 86.

[49] N. A. S. Webster, I. C. Madsen, M. J. Loan, R. B. Knott, F. Naim, K. S. Wallwork and J. A. Kimpton. An Investigation of Goethite-Seeded Al[OH]3 Precipitation using In Situ X-ray Diffraction and Rietveld-Based Quantitative Phase Analysis. *J. Appl. Cryst*. (2010). **43**. 46.

[50] v. Esteve, L. E. Ochando, M. M. Reventos, G. Peris and X. M. Amigo. Quantitative Phase Analysis of Mixtures of Three Components Using Rietveld and Rius Standardless Methods: Comparative Results. *Cryst. Res. Tech*. (2000). **35**. 1183.

[51] N. V. Y. Scarlett, I. C. Madsen, C. Manias and D. Retallack. On-Line X-ray Diffraction for Quantitative Phase Analysis: Application in the Portland Cement Industry. *Powder Diff.* (2001). **16**. 71.

[52] C. D. Rosin and R. K. Belew. New Methods for Competitive Coevolution. *J. Evol. Comp*. (1997). **5**. 1.

[53] J. Sardanyés. Matching Allele Dynamics and Coevolution in a Minimal Predator Prey Replicator Model. *Phys. Lett. A.* (2008). **372**. 341.

[54] R. P. Wiegand, W. C. Liles and K. A. De Jong. An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms. Proc. Gen. Evol. Conf. Morgan Kaufmann Publishers.

[55] Z. Yang, H. Tang and X. Yao. Large Scale Evolutionary Optimization using Cooperative Coevolution. *Info. Sci*. (2008). **178**. 2985.

[56] L. M. Simao, D. M. Dias and M. A. l. C. Pacheco. Refinery Scheduling Optimization using Genetic Algorithms and Cooperative Coevolution. IEEE Symp. Comp. Intel. Sched. (2007). 151.

[57] A. Bucci and J. B. Pollack. On Identifying Global Optima in Cooperative Coevolution. Gecco. *Genetic and Evolutionary Computation Conference*. (2005). **1-2**. 539.

[58] M. A. Potter and K. A. De Jong. A Cooperative Coevolutionary Approach to Function Optimization. Third Parallel Problem Solving From Nature. Jerusalem Israel. (1994). 249.

Chapter 3. Optimisation of differential and cultural differential evolution algorithms applied to structure solution of molecular crystals.

Although global optimisation algorithms can and have been successfully applied to direct space methods and used to solve crystal structures, these algorithms suffer from intrinsic limitations. One disadvantage is the large number of model structures generated during a single search progression and the significant computational effort required to evaluate all these models before a sufficiently accurate model is located. Unlike the exhaustive grid search method, global optimisation algorithms do not evaluate every possible crystal structure and can converge prematurely in local minima (locating an incorrect model) if governed by control parameters with non-optimal values.

In this chapter, DE and CDE algorithms using different combinations of control parameters are applied to the direct space method and used to solve three different crystal structures that have been previously solved by our DE technique. Since the crystal structures have already been solved by our DE technique it is possible to evaluate the success and efficiency of each structure solution calculation using different combinations of control parameters.

The three known crystal structures used for these tests are; baicalein (section 3.2), adipamide (section 3.3) and acetarsone (section 3.4). An attempt is made to determine if there is a 'universal' optimal combination of DE control parameters that increases the probability that a search is successful (locates the correct crystal structure), or if certain 'test' structures are best solved by a combination of DE parameters particular to that structure. This prior knowledge could be used to reduce the number of model structures evaluated during a search (thus the time needed to solve a structure), and increase the probability that a search converges successfully, therefore reducing the need for computationally demanding multiple searches.

3.1 The Differential Evolution algorithm

The differential evolution algorithm used in this thesis and in previous work, to solve crystal structures, [1-3] has four control parameters (discussed in sections 2.2.2-2.2.5) that can be assigned values by the user. These four control parameters are: (*K*) the recombination rate, (*F*) the mutation rate, (*NP*) the number of structural models in a population and (*Gmax*) a user defined stopping criteria. Although each of these parameters is used to control specific operations of the differential evolution algorithm, their effect on the efficiency and reliability of the search is coupled. Hence it is feasible to systematically test different combinations of these control parameters and analyse what effect these different combinations have on the search.

3.1.1 The *Gmax* control parameter

From experience it is possible to make an educated guess regarding how many generations will be required to allow a search of defined complexity to converge on a solution (although this may not always be the global optimum). If a search fails to converge within a certain number of generations it often indicates that the search has become trapped in a local minimum. To avoid continuing the calculation of trapped searches, the parameter *Gmax* is used to terminate searches that have failed to converge after a convenient number of generations.

3.1.2 *K*, the recombination control parameter

The recombination parameter *K* (discussed in section 2.2.3) can be assigned any value between zero and one. It has been demonstrated [1,2] that assigning *K* a value of one causes a search to converge rapidly but can increase the probability that the search converges prematurely. Reducing the value of *K* to 0.99 maintains a fast rate of convergence whilst also reducing the probability that a search will converge prematurely in a local minimum, failing to locate the optimal solution (correct crystal structure). [3] In the work presented here *K* is always assigned a value of 0.99. Having established that the values of *Gmax* and *K* can be pre-defined at optimal values, in the work discussed in this chapter, the combination of only two control parameters, *F* and *NP*, are investigated.

3.1.3 *F*, the mutation control parameter

The value of the mutation parameter *F* (discussed in section 2.2.3) controls the rate at which new genetic material is introduced into the population. *F* can be assigned any value between zero and one. However, in this application of DE, *F* is usually assigned a value in the range 0.3-0.8. [1-3] High mutation rates (achieved by assigning *F* a large value) ensure that the level of genetic diversity in the population remains high for a greater number of generations. This high level of genetic diversity encourages a thorough exploration of the landscape, increasing the probability that a search locates the global minimum. However, if the value of *F* is too large, the convergence rate of the population will be inconveniently slow, requiring more generations to converge and maybe terminated by *Gmax*. Smaller values of *F* cause the population to converge rapidly, but if the mutation rate is too small the genetic diversity of the population is rapidly lost. This increases the probability that a search converges prematurely. Thus a compromise must be found such that searches have a high probability of solving the crystal structure but do this in the least number of generations.

3.1.4 *NP*, the population size

The value of *NP*, the population size, determines how many models or individuals are in a population. This means that the value of *NP* has significant influence on the initial amount of genetic diversity in a population and hence is considered as an algorithmic control parameter. The size of the landscape representing a particular crystal structure solution problem depends on the number and range of values of the structure parameters that define a model (as discussed in section 2.2.1). The size of the landscape is not dependent on the size of a population, thus when a larger population is initialised in a particular landscape, that landscape will have a higher population density than if a smaller population is used. The increased population density increases the probability that an initial model is generated near the global minimum. The reduction in the distance between this initial “best” model and the global minimum reduces the number of generations required for the evolutionary processes to move the best model into the global minimum. This means that searches using larger populations can converge successfully in fewer generations than searches using smaller populations. However, excessive genetic diversity caused by a large population can also slow the convergence rate of a search. The greater the population size, the longer it takes for the evolutionary process to reduce the genetic diversity of the population so that all models become identical and converge. It is also computationally more demanding to evolve a large population as every model breeds one child per generation, each of which must then have a fitness evaluated. Thus again, a compromise between high genetic diversity and low computational demand needs to be made.

The parameter *NP* can either be assigned a value that is independent from the problem or a value that is related to the particular optimization problem under consideration. As discussed in sections 1.3.2 and 2.2.1, the complexity of crystal structure solution by direct space methods is linked to the number of structure parameters needed to define the orientation, conformation and position of a model in the unit cell. Thus as the number of structure parameters needed to define a model increases it is logical to use a larger population size as this increases the initial genetic diversity of a population. In these studies, the size of the population is automatically calculated as a multiple of the number of parameters needed to define a model. For example, the crystal structure solution of baicalein is performed using a model defined by seven parameters (figure 3.1), hence searches were carried out using population sizes of 49 (7x7), 70 (7x10), 105 (7x15) 140 (7x20) and 280 (7x40). The crystal structure solution of adipamide required a structural model defined by eight parameters (figure 3.4), and hence searches with population sizes of 56 (8x7), 80 (8x10), 160 (8x20) and 320 (8x40) were used. Crystal structure solution of acetarsone is performed using a model (figure 3.5) defined by nine parameters. Hence searches using population sizes of 63 (9x7), 90 (9x10), 135 (9x15), 180 (9x20) and 360 (9x40) models were used to solve the crystal structure.

3.1.5 Cultural differential evolution

As discussed in section 2.3, the addition of cultural knowledge to the DE search can increase the rate at which a search converges. It is postulated that before a search begins to converge rapidly, a significant number of models cluster around a single point in parameter space. The identification of a cluster of models can be used to restrict the exploration of a search by actively adjusting the position of the landscape boundaries as the search progresses, resulting in a confined search that is encouraged to only explore the area of landscape near the global minimum. In our implementation of CDE [4,5] the position of the landscape boundaries is controlled by measuring which areas of the landscape have a relatively low population density and excluding these areas from the search. This means that rather than waiting for a cluster of models to locate the global minimum before moving the boundaries (which may take a considerable number of generations), it is possible to begin confining a search as soon as a significant number of models start to cluster. In this way cultural knowledge can be used to control the position of the boundaries after fewer generations, potentially further reducing the number of generations required by a search to converge.

3.1.6 Initiating the cultural pruning

In our implementation of CDE, [4,5] a conventional DE search is performed for the first 50 generations of a search, with boundaries maintained at their default positions and models allowed to explore the whole landscape. It is assumed that during this stage of the search, many parent models are replaced by fitter children and the Darwinian survival of the fittest principle promotes the clustering of models in minima. After the 50th generation, it is assumed that most models will have clustered in small groups and only relatively few remain sparsely spread across areas of the landscape with a relatively low population density. Thus after the 50th generation, cultural knowledge is applied to prevent further exploration of the areas of the landscape with relatively low population density.

3.1.7 Population under-threshold *NUT* parameter

As discussed in section 2.3, *NUT* is used to define how many models will be considered as cultural outliers after each generation. Increasing the value of *NUT* increases the number of models in a population that are treated as cultural outliers therefore causing the landscape boundaries to move more rapidly, and in theory, increase the rate of convergence of the search.

The cultural differential evolution algorithm is controlled by the parameter *NUT* in combination with the parameters *NP*, *F*, *K* and *Gmax*. CDE searches were used to solve the crystal structures of baicalein, adipamide and acetarsone using different combinations of *NP*, *F* and *NUT* in order to determine relationships between these parameters and identify the optimal combination. Compared with an analogous DE search using the same *NP* and *F* combination, the optimal value for *NUT* is the value that results in the fastest convergence rate whilst not reducing the probability that a search locates the optimum crystal structure.

3.2 The crystal structure solution of baicalein

3.2.1 Background

Baicalein (5, 6, 7-trihydroxyflavone) is a flavone that is commonly used in Chinese herbal medicine, and displays a variety of useful biological properties including; anti-inflammatory, anti-viral and anti-cancer activity. [6-11] The crystal structure of baicalein has been determined using data obtained from single crystal diffraction experiments performed at ambient [12] and low temperature [13].

3.2.2 Optimisation of the DE algorithm

The direct space structure solution of baicalein requires no constraints to be placed on the position or orientation of the molecule within the unit cell as the space group symmetry requires one molecule in a general position. The structural model of baicalein is then defined by seven parameters; three parameters to define the position (*x,y,z*: between 0 and 1) and orientation (φ,ψ,θ: between 0 and 360) of the molecule and one torsion parameter (τ1: between 0 and 360) to define the intramolecular rotation between the two rigid units as shown in figure 3.1.

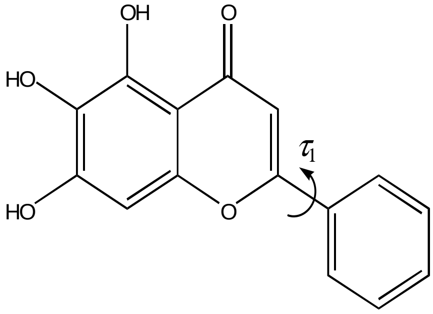
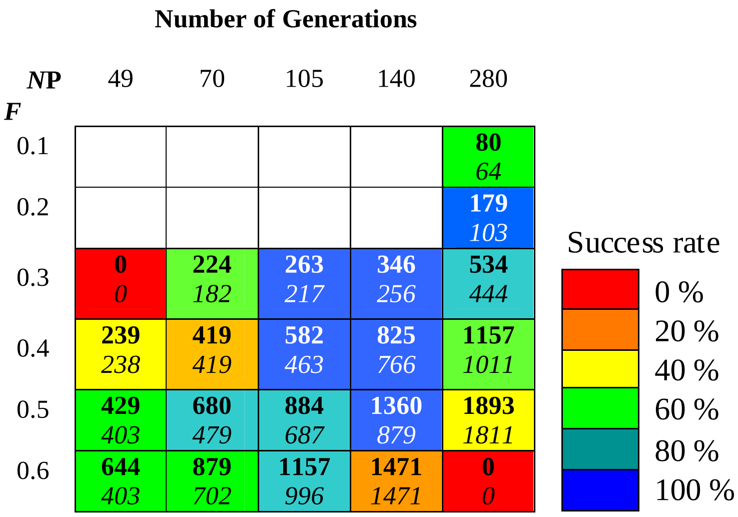


Figure 3.1. The structural model of baicalein used in the DE calculations. Arrows indicate torsional flexibility, figure taken from. [5]

Baicalein is a suitable test case for control parameter optimisation of DE because the searches consistently solve the crystal structure within a convenient number of generations using a variety of combinations of *F* and *NP* control parameters. Thus it is possible to draw meaningful conclusions on how different combinations of the population size and mutation rate affect a search.

In the case of baicalein, for a successful Rietveld refinement of the crystal structure based on a successful structure solution, the model located by the search was required to have an *R* factor (R_{wp}) <= 15.6%. All models assigned an *R* factor <= 15.6% were judged to be close enough for successful refinement. The DE structure solution calculation was run five times for each combination of *NP* and *F* parameters: *NP*=49, 70, 105 and 140 with *F*=0.3-0.6 and *NP*=280 with *F*=0.1-0.6. For each combination of *NP* and *F*, the *R* factor of the optimum solution located by each of the five searches was recorded. If the *R* factor of the final solution was <= 15.6%, the search was judged to be successful. The results of these successful searches are shown in Table 3.1.



Number of Generations

NP ***F***	49	70	105	140	280
0.1					**80** *64*
0.2					**179** *103*
0.3	**0** *0*	**224** *182*	**263** *217*	**346** *256*	**534** *444*
0.4	**239** *238*	**419** *419*	**582** *463*	**825** *766*	**1157** *1011*
0.5	**429** *403*	**680** *479*	**884** *687*	**1360** *879*	**1893** *1811*
0.6	**644** *403*	**879** *702*	**1157** *996*	**1471** *1471*	**0** *0*

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.1: Structure solution of baicalein by DE using different combinations of *NP* (population size) and *F* (mutation rate). The success rate over the five calculations for each *NP* and *F* combination is indicated in the colour chart: blue for 100% success and red for 0% success rate. The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. For searches performed with *NP*=49, 70, 105 and 140 *Gmax*=1500, for searches with *NP*=280 *Gmax*=2000.

3.2.3 Optimised combination of the *NP* and *F* control parameters

Table 3.1 shows that in general, as the population size increases, the success rate of the search also increases. Calculations using the smallest population size (49) are the least successful in solving the crystal structure using the mutation range used in this study. For example, a mutation rate of 0.4, and a population size of 49 has a 40% success rate, whereas populations of 70, 105, 140 and 280 solve the crystal structure with 20%, 100%, 80% and 60% success respectively. Similar trends can be seen for other mutation rates, demonstrating that searches using larger population sizes with greater initial genetic diversity are more likely to converge successfully than searches using smaller population sizes with less initial genetic diversity.

The last column (in table 3.1) shows the results from a population size of 280 in which the amount of genetic diversity initially generated in the population is increased. The high level of genetic diversity means that searches are often more efficient (and successful) using smaller mutation rates, such as *F*=0.1 and 0.2, and hence these additional runs were performed for *NP*=280. The large size of the population reduces the probability that a significant number of models cluster in a single local minima causing a search to converge prematurely. If searches with *NP*=280 are performed with large *F* the convergence rate is reduced and hence the specified (*Gmax*) number of generations was increased for *NP*=280.

The trend in success rate clearly shows a correlation between mutation rate and population size with the only conclusion being that it seems with a higher population size, lower *F* values can be used. The most successful combination is that of median population size (*NP*=105 and 140), with median mutation rate (*F*=0.3-0.5) in which 29/30 runs were successful. There is however a clear trend that an increase in the mutation rate increases the number of generations required by a search to converge on the correct solution (figure 3.2). The four plots shown in this figure have roughly the same shape but clearly show that searches performed with larger *F* take significantly more generations to converge. In all four cases, we can clearly see the convergence of the calculation as the optimum fitness and the mean fitness of the population become the same. It is clear from table 3.1 that for every increase in the mutation rate of 0.1, the average number of generations required for convergence increases significantly. As the size of the population

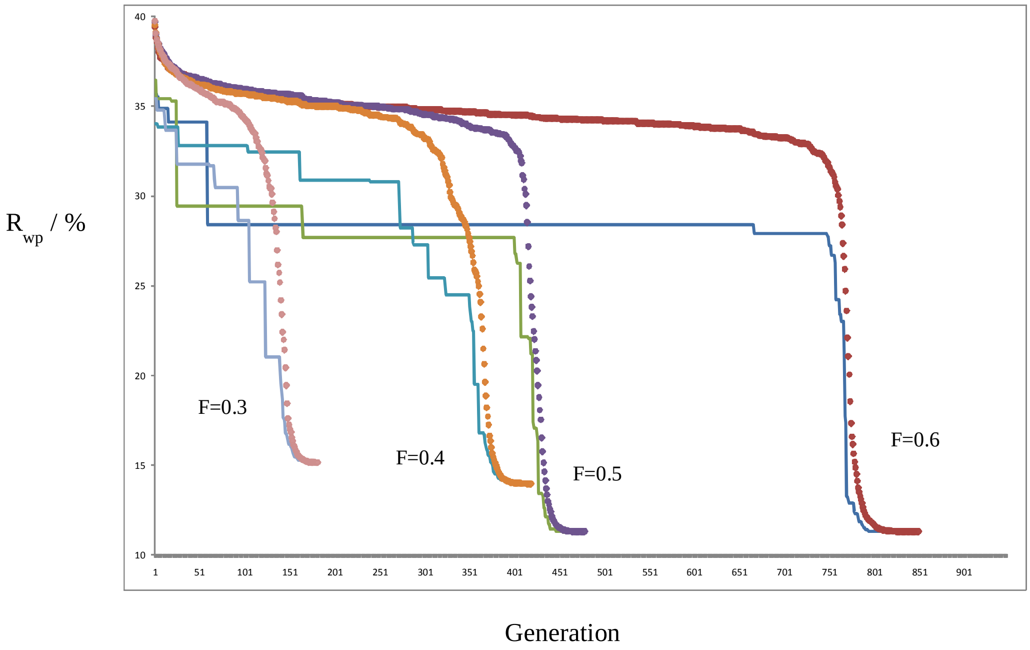


Figure 3.2, The convergence rate of four DE searches with *NP*=70 and *F*=0.3, 0.4, 0.5 and 0.6. Dots indicate the mean R$_{wp}$ of the population, whereas the line shows the evolution of the fittest within each generation, R$_{wp}$ best. For each set of parameters, the most efficient calculation is shown.

increases this slowing in the convergence rate by increasing the mutation rate becomes more significant. For example, in the case of *NP*=140, searches with *F*=0.5 require approximately four times as many generations as those with *F*=0.3. When *F*=0.6, a significant number of the searches fail to converge successfully within the specified maximum number of generations and the effect on success rate is clearly shown (table 3.1). This trend is demonstrated further in the results obtained using a population size of 280, in which some searches using *F*=0.5-0.6 fail to converge within the specified maximum number of generations.

These results also demonstrate the effect of population size on the speed of convergence and clearly show that as the population size increases, the average number of generations required also increases. This can clearly be seen by examination across a row in table 3.1, for example *F*=0.5. As the population size increases, the amount of genetic diversity in the initial population also increases. For a search to converge, all models in a population must have the same genetic information. If two populations of different size evolve using the same rate of mutation and recombination, it will take more generations for the larger population to reduce its larger initial amount of genetic diversity and converge. This is illustrated further in figure 3.3 which shows the convergence rate of four searches with *F*=0.3 and increasing population size.

However, by using very small mutation rates it is possible to use large population sizes to solve the crystal structure in fewer generations than those searches using smaller population sizes. Table 3.1 shows that the DE search with *NP*=280 and *F*=0.2 converges with 100% success on average in 179 generations. We can compare this with the DE search with *NP*=105 and *F*=0.3 which converges with 100% success on average in 263 generations. A population size of 280 initially generates a higher population density in the landscape than a population size of 105, and hence there is a greater probability that members of the initial population are generated closer to the global minimum in the larger population. The small mutation rate then allows the search with *NP*=280 to converge more rapidly than that with *NP*=105. The increased genetic diversity of the larger population reduces the probability that a significant number of models cluster in one local minimum causing the search to converge prematurely.

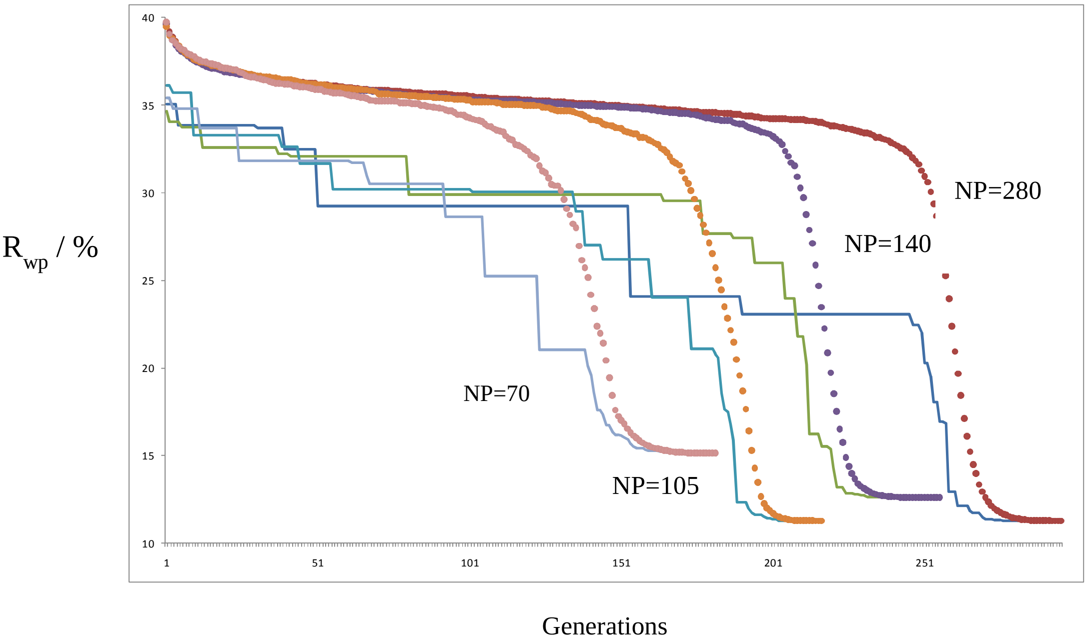


Figure 3.3, The convergence rate of four searches with *F*=0.3 and *NP*=70, 105, 140 and 280. Dots indicate the mean R_{wp} of the population, whereas the line shows the evolution of the fittest within each generation, R_{wp} best. For each set of parameters, the most efficient calculation is shown.

However, when comparing different population sizes, searches that converge in the least generations are not necessarily the quickest to converge in real time as the rate determining step of direct space crystal structure solution is the child fitness evaluation step. DE systematically selects every model in the population to act as parent in each generation, so the R_{wp} fitness evaluation must be calculated *NP* times in each generation. Hence, with *NP*=280 and *F*=0.2, the DE search solves the crystal structure on average in 179 generations or 179x280=50120 child fitness evaluations. In comparison the search with *NP*=105 and *F*=0.3 converges on average in 263 generations or 263x105=27615 child fitness evaluations. Hence care must be taken in interpretation of these results and although the number of generations for convergence is useful for comparison within a given *NP* value, the number of fitness evaluations will be used later in this thesis to assess efficiency.

Number of Fitness Evaluations

NP ***F***	49	70	105	140	280
0.1					🟩 **22400** *17920*
0.2					🟦 **50120** *28840*
0.3	🟥 **0** *0*	🟩 **15680** *12740*	🟦 **27615** *22785*	🟦 **48440** *35840*	▨ **149520** *124320*
0.4	🟨 **11711** *11662*	🟧 **29330** *29330*	🟦 **61110** *48615*	🟦 **115500** *107240*	🟩 **323960** *283080*
0.5	🟩 **21021** *19747*	▨ **47600** *33530*	▨ **92820** *72135*	🟦 **190400** *123060*	🟨 **530040** *507080*
0.6	🟩 **31556** *19747*	🟩 **61530** *49140*	▨ **121485** *104580*	🟧 **205940** *205940*	🟥 **0** *0*

Success rate	
🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.2, The average number of child fitness evaluations calculated by searches using different combinations of *NP* and *F*.

Table 3.2 shows that in terms of the most reliable and most efficient calculation in terms of fitness evaluations, the combination of median *NP*=105 and *F*=0.3 is optimum, whereas the quickest single calculation was that, as expected, with the smallest *NP*=49 and *F*=0.3.

3.2.4 Optimised combination of *NP*, *F* and *NUT* control parameters

The CDE was run for baicalein, using the same model, data and criteria as the DE calculation, five times for each combination of *NUT*, *NP* and *F*. The results of these searches are shown in Tables 3.3a-e.

Mean G_{con}

(a)

N_{ut} ***F***	static	1	2	3	4	5	6
0.3	🟥 **0** *0*	🟥 **0** **0**	🟧 **121** *121*	🟥 **0** *0*	🟧 **147** *147*	🟥 **0** *0*	🟥 **0** *0*
0.4	🟨 **239** *238*	▨ **279** *175*	▨ **331** *266*	🟦 **281** *245*	🟧 **246** *246*	🟥 **0** *0*	🟧 **219** *219*
0.5	🟩 **429** *403*	🟧 **550** *550*	▨ **369** *306*	🟩 **328** *222*	🟧 **207** *207*	🟧 **1201** *1201*	🟥 **0** *0*
0.6	🟩 **644** *403*	🟩 **412** *314*	🟦 **722** *415*	🟧 **598** *598*	🟨 **289** *203*	🟨 **251** *229*	🟥 **0** *0*

Success rate	
🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.3: Structure solution of baicalein by CDE using different combinations of F (mutation rate) and *NUT* (cultural pruning parameter) for populations of a)49, b)70, c)105, d)140 and e)280. The static column indicates a conventional DE search with no cultural pruning. The success rate over the five calculations for each *NP*, *F* and *NUT* combination is indicated in the colour chart: blue for 100% success and red for 0% success rate. The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. For searches performed with *NP*=49 *Gmax*=1500, for all other *NP* values *Gmax*=2000.

(b)

Mean G_{con}

N_{ut} F	static	1	2	3	4	5	6	7	8
0.3	🟩 **224** *182*	🟨 **179** *170*	🟨 **245** *224*	▨ **174** *153*	🟧 **150** *150*	🟨 **181** *173*	🟩 **170** *153*	🟧 **163** *163*	🟥 **0** *0*
0.4	🟧 **419** *419*	▨ **396** *343*	🟦 **325** *246*	🟦 **370** *315*	🟧 **257** *257*	🟨 **269** *258*	🟧 **225** *225*	🟨 **232** *185*	🟥 **0** *0*
0.5	▨ **680** *479*	🟩 **690** *644*	🟦 **656** *430*	▨ **597** *450*	▨ **532** *412*	🟨 **405** *397*	🟨 **297** *288*	🟥 **0** *0*	🟧 **193** *193*
0.6	🟩 **879** *702*	🟩 **1127** *1033*	🟩 **977** *846*	🟦 **845** *637*	▨ **675** *560*	🟩 **491** *368*	🟩 **359** *325*	🟨 **274** *233*	🟧 **247** *247*

Success rate	
🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

(c)

Mean G_{con}

N_{ut} F	static	3	4	5	6	7	8	9	10	11	12
0.3	🟦 **263** *217*	▨ **287** *252*	▨ **254** *236*	🟨 **242** *235*	▨ **296** *275*	🟨 **194** *181*	🟩 **193** *177*	🟨 **257** *191*	🟧 **183** *183*	🟥 **0** *0*	🟨 **166** *120*
0.4	🟦 **582** *463*	🟦 **540** *260*	🟦 **417** *345*	▨ **399** *276*	▨ **398** *310*	▨ **406** *291*	▨ **318** *251*	🟩 **282** *234*	▨ **267** *248*	🟧 **195** *195*	🟧 **261** *261*
0.5	▨ **884** *687*	🟦 **949** *851*	🟦 **818** *561*	▨ **721** *630*	🟦 **588** *498*	▨ **447** *391*	🟩 **380** *372*	🟦 **349** *299*	🟨 **333** *274*	🟩 **293** *241*	🟥 **0** *0*
0.6	▨ **1157** *996*	▨ **1644** *1502*	🟦 **1435** *926*	▨ **1175** *1001*	🟦 **926** *763*	🟦 **602** *539*	▨ **498** *461*	🟨 **347** *294*	▨ **390** *307*	▨ **337** *304*	🟨 **283** *274*

(d)

Mean G_{con}

N_{ut} F	static	3	4	5	8	9	10	11	12	13	14	15
0.3	🟦 **346** *256*	🟦 **402** *174*	▨ **293** *240*	🟦 **284** *237*	🟦 **249** *224*	▨ **230** *185*	▨ **247** *231*	▨ **219** *161*	🟩 **225** *196*	🟨 **214** *193*	🟨 **188** *179*	🟧 **249** *249*
0.4	▨ **825** *766*	🟦 **747** *587*	🟦 **630** *522*	▨ **690** *608*	🟦 **389** *274*	🟦 **342** *258*	🟦 **355** *308*	🟦 **302** *272*	🟩 **235** *213*	▨ **294** *235*	🟨 **260** *226*	🟧 **406** *406*
0.5	🟦 **1360** *879*	🟦 **1256** *840*	🟦 **1291** *1172*	🟦 **989** *773*	🟦 **593** *446*	🟦 **527** *461*	🟦 **429** *360*	▨ **441** *400*	🟦 **364** *321*	▨ **352** *328*	🟨 **311** *279*	🟩 **268** *257*
0.6	🟧 **1471** *1471*	🟩 **1460** *1293*	🟦 **1658** *1285*	🟨 **1613** *1470*	🟦 **951** *747*	🟦 **747** *587*	🟦 **630** *522*	🟦 **513** *439*	▨ **470** *377*	▨ **418** *346*	🟩 **424** *395*	▨ **354** *307*

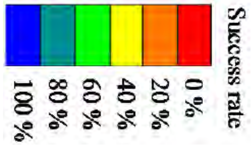
Table 3.3. Continued

(e)

Mean G_{con}

N_{ut} F	static	5	8	10	13	14	15	16	17	18	19	20	21	22	26	28	30	32	34
0.1	**80** *64*	**68** *68*	**85** *74*	**81** *81*	**83** *71*	**70** *70*	**78** *78*	**72** *66*	**82** *82*	**91** *89*	**79** *67*	**0** *0*	**74** *62*	**83** *75*	**56** *56*	**268** *268*	**82** *77*	**74** *74*	**90** *90*
0.2	**179** *103*	**198** *115*	**192** *156*	**172** *123*	**191** *156*	**169** *126*	**169** *143*	**155** *105*	**167** *148*	**156** *114*	**167** *143*	**151** *130*	**150** *146*	**148** *109*	**171** *154*	**124** *107*	**169** *153*	**139** *128*	**159** *138*
0.3	**534** *444*	**519** *312*	**619** *440*	**415** *326*	**347** *291*	**428** *348*	**393** *378*	**316** *253*	**296** *247*	**280** *231*	**304** *284*	**287** *253*	**253** *239*	**244** *209*	**212** *180*	**223** *212*	**199** *193*	**209** *209*	**180** *180*
0.4	**1157** *1011*	**1182** *1182*	**1004** *723*	**997** *798*	**824** *707*	**595** *494*	**702** *495*	**600** *494*	**431** *404*	**476** *393*	**435** *387*	**411** *372*	**402** *336*	**398** *336*	**289** *260*	**289** *256*	**262** *250*	**239** *218*	**252** *244*
0.5	**1893** *1811*	**1758** *1723*	**1862** *1696*	**1732** *1561*	**1487** *1297*	**1042** *999*	**1042** *971*	**960** *754*	**740** *69*	**676** *569*	**646** *581*	**608** *511*	**562** *502*	**556** *494*	**416** *365*	**354** *280*	**314** *279*	**283** *258*	**265** *259*
0.6	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**1779** *1668*	**1758** *1486*	**1428** *1218*	**1241** *1046*	**1085** *964*	**838** *758*	**819** *663*	**786** *666*	**790** *525*	**494** *445*	**458** *441*	**400** *354*	**365** *303*	**398** *352*

Table 3.3. Continued



Tables 3.3a-e demonstrate that the addition of cultural evolution to a traditional DE search can reduce the number of generations required for convergence. The results also show that as the value of *NUT* is increased, searches progressively require fewer generations to converge. For example, a static DE with *NP*=70 and *F*=0.5 (Table 3.3b) converges on average in 680 generations, while the analogous CDE searches with *NUT*=2-5 converge on average in 656, 597, 532 and 405 generations respectively. This increase in efficiency is more marked with greater population size such as *NP*=280, in which the static DE search with *F*=0.4 converges on average in 1157 generations and the analogous CDE searches with *NUT* = 20-26 converge on average in 411, 402, 398 and 289 generations respectively.

These results also demonstrate that as the size of the population is increased the value of *NUT* can also be significantly increased without reducing the probability that a CDE search is successful. This is immediately clear for example from table 3.3b (*NP* = 70) in which pruning with *NUT* > 5 results in less success than the static DE search. However, for *NP* = 280 (table 3.3e), only a value of *NUT* > 30 results in CDE searches with a lower success rate than the static DE. As the value of *NUT* is increased, more members of the population are treated as outliers. The area of the landscape occupied by outliers is removed from the search, and thus a higher value of *NUT* confines the search to a smaller area of landscape. As a search becomes more confined, the genetic diversity of the population decreases and the probability that a search will become trapped in local minima increases.

Larger populations initially have higher levels of genetic diversity than smaller populations and hence can withstand greater amounts of cultural pruning. This is illustrated by the greater success rate and efficiency following cultural pruning with *NP* = 280 but the detrimental effect the CDE approach has on the success and speed of the small population *NP* = 49 calculations (table 3.3a). It is also clear from this set of results that there may be an optimum relationship between the *NP* and *NUT* ratios.

There is also a small effect of mutation rate on success aligned with an increase in *F* with *NUT*, but this trend is not as marked as other combinations discussed. Since mutation introduces new genetic material into a population thus slowing the convergence rate and the cultural function reduces the genetic diversity of a population thus increasing the convergence rate it is logical to observe this relationship between the *NUT* and *F* control parameters.

The CDE tables demonstrate that CDE searches with sufficiently large *NP* and *F* can converge in significantly fewer generations than that required by static DE searches. For example, in table 3.3e, the DE search with $F = 0.5$ converges with 40% success on average in 1893 generations, whereas the analogous CDE searches with $NUT = 28$ converge with 100% success on average in 354 generations, representing an increase in efficiency of 81%.

3.2.5 Summary

These results demonstrate that the size of a population, mutation rate and amount of cultural pruning all affect the success and efficiency of a search. Searches using larger populations are significantly more likely to be successful than those using smaller populations because of more initial genetic diversity. Searches using larger populations can also converge in fewer generations but a better indicator of real calculation time is that of the number of structure evaluations calculated which means that searches conducted using large populations tend to be slower. In general, searches conducted using larger mutation rates often take longer to converge, and smaller values of *F* can be used with larger *NP*. Cultural pruning can be used to dramatically reduce the number of generations required for a successful solution, but aggressive pruning can significantly reduce this success rate. An increase in the mutation rate can increase the success rate but the contrasting effect on the genetic diversity of *NP*, *F* and *NUT* means that it is not trivial to find the optimal combination of these three parameters.

3.3 The crystal structure solution of adipamide

3.3.1 Optimisation of the DE algorithm

The crystal structure of the triclinic form of adipamide (1,6-hexanediamide, $C_6H_{12}N_2O_2$) has been previously solved by the direct space method, [2] and is known to adopt the P-1 space group through an internal centre of symmetry. However, here the structure solution is attempted in the P1 space group. The position of the molecule in the unit cell is hence arbitrary and the structure solution requires no constraints on the orientation or flexibility of the molecule. Unlike baicalein, adipamide has considerable intramolecular flexibility, with the model defined by eight structure parameters; three parameters define the orientation (φ,ψ,θ: between 0 and 360) of the model and

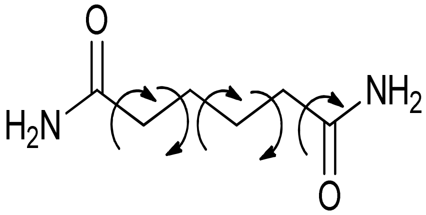


Figure 3.4. The structural model of adipamide used in the DE calculations. Arrows indicate torsional flexibility.

five torsion parameters (τ1-t5: between 0 and 360) define the intramolecular flexibility as shown in figure 3.4. Adipamide was chosen as a suitable test case because the searches consistently solve the crystal structure within a convenient number of generations and it demonstrates that the exclusion of prior knowledge of symmetry from a search does not prevent location of a structure that is compatible with the known crystal structure. In addition, it will test the ability of DE to solve the crystal structure of a molecule with more flexibility.

For adipamide, a successful structure solution was deemed to be a model located by the search with an *R* factor (R_{wp}) <= 16.0%. Hence, all models assigned an *R* factor <= 16.0% were judged to be close enough for successful Rietveld refinement. The DE structure solution calculation was run five times for each combination of *NP* and *F*: *NP* = 56 with *F* = 0.5-0.8, *NP* = 80 with *F* = 0.3-0.8, *NP* = 160 and 320 with *F* = 0.1-0.8. For each combination of *NP* and *F*, the *R* factor of the optimum solution located by each of the five searches was recorded. The results of these searches are shown in Table 3.4.

3.3.2 Optimised combination of *NP* and *F* control parameters

The results presented in table 3.4 demonstrate that the mutation rate has a significant effect on the success of the search. The smallest *F* values generally have the least success (*NP* = 80 and 160), whereas those with *F* = 0.4-0.8 converge in most cases, although there is no clear trend within these results. Unlike the structure solution of baicalein where searches with *NP* = 49 and 70 have relatively low success rates compared to searches with larger *NP*, for solution of adipamide there is not such a significant decrease in success rate with decreasing population size. For *F*=0.5-0.8 searches with *NP* = 56,160 and 320 solve the structure of adipamide with the same overall success rate. The fact that searches with smaller *NP* still have relatively high success rates suggests that the structure solution of adipamide is different in character to that of baicalein.

Number of Generations

F \ *NP*	56	80	160	320
0.1			🟥 **0** *0*	🟨 **52** *49*
0.2			🟧 **86** *86*	▨ **124** *90*
0.3		🟥 **0** *0*	▨ **171** *146*	🟦 **251** *216*
0.4		🟩 **224** *182*	🟦 **358** *256*	🟩 **371** *357*
0.5	▨ **217** *159*	▨ **369** *245*	🟩 **516** *462*	🟦 **699** *543*
0.6	🟦 **436** *329*	▨ **497** *413*	▨ **802** *676*	▨ **880** *628*
0.7	🟩 **616** *550*	🟦 **689** *575*	🟦 **1107** *769*	🟦 **1265** *1028*
0.8	🟦 **752** *571*	🟦 **1113** *886*	▨ **1322** *1203*	🟩 **1543** *1362*

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.4: Structure solution of adipamide by DE using different combinations of *NP* (population size) and *F* (mutation rate). The success rate over the five calculations for each *NP* and *F* combination is indicated in the colour chart (as in Table 3.1). The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. For *NP*=56 searches were performed with *Gmax*=1500, whereas for *NP*=80,160 and 320, *Gmax*=2000.

The R_{wp} landscape representing the crystal structure solution of adipamide has eight dimensions whereas that of baicalein is seven. However, the shape of these two landscapes will have a greater influence on the character of a search than the landscape dimensionality. The intramolecular flexibility of adipamide means that structural models can adopt many different conformations. Although there is still only one correct solution, some of these incorrect conformations will be more plausible than others. This results in an R_{wp} landscape that contains numerous local minima, so there is a high probability that the adipamide search explores many local minima before locating the global minimum. Additionally, the high density of local minima potentially means that at least one minimum is significantly near the global minimum. Thus a search converging in a local minimum that is very near the global minimum can be considered successful i.e. close enough for successful Rietveld refinement of the non-global solution. In contrast, the structural model of baicalein comprises two rigid units linked with one torsional degree of freedom. This means that few different conformations can be adopted so the landscape contains few local minima. This reduces the probability that the baicalein search explores many local minima before locating the global minimum. It also reduces the probability that a search that converges in a local minimum is considered successful.

This is similar to the conclusion from a previous study [14] in which the efficacy of the direct space method using an R_{wp}-based cost function was investigated in the solution of the crystal structure of $Ph_2PO(CH_2)_7POPh_2$. The flexibility of the aliphatic chain means that many different conformations of the model are possible, resulting in an R_{wp} landscape containing many local minima. Since a cost function based only on R_{wp} evaluates the model as a whole, a search cannot easily identify perturbations to parameters defining the aliphatic chain that produce significantly better models. As a result, a search is likely to visit many local minima before locating the global minimum. During this study, [14] a SA search indeed evaluated many incorrect models before locating the global minimum.

In cases where the landscape contains numerous local minima it is likely that a DE search spends a significant number of generations exploring and escaping local minima. Higher mutation rates increase the probability that the search escapes local minima and converges in the global minimum. An increase in population size increases the genetic diversity and density of the population in the landscape, but if it is difficult to identify better models, increasing the size of a population merely creates more models that do not assist the search to rapidly locate the global minimum. Thus the ability of a search to converge successfully is less dependent on the size of a population and more dependent on the mutation rate as illustrated in table 3.4. However, table 3.4 demonstrates that increasing the population size decreases the convergence rate. Searches with equivalent F values e.g: for F = 0.6 and NP = 56 and 160 converge on average in 436 and 802 generations respectively. It is possible that searches with NP = 320 and F = 0.8 would have been more successful if *Gmax* had been assigned a larger value. Although higher mutation rates are generally more successful, as expected, this also increases the number of generations required by a search to converge, e.g: for NP = 80 and F = 0.5-0.6 searches converge on average in 369 and 497 generations respectively.

Again, although table 3.4 demonstrates that searches using larger population sizes and small mutation rates successfully converge in the fewest generations, the calculation time is still longer because significantly more child fitness evaluations are calculated by a search with NP = 320 than a search with NP = 56 (table 3.5). The quickest single calculation is that with the smallest NP = 56 and F = 0.5 carried out in this study.

Number of Fitness Evaluations

NP ***F***	56	80	160	320
0.1			🟥 **0** *0*	🟨 **16640** *15680*
0.2			🟧 **13760** *13760*	▨ **39680** *28800*
0.3		🟥 **0** *0*	▨ **27360** *23360*	🟦 **80320** *69120*
0.4		🟩 **17920** *14560*	🟦 **57280** *40960*	🟩 **118720** *114240*
0.5	▨ **12152** *8904*	▨ **29520** *19600*	🟩 **82560** *73920*	🟦 **223680** *173760*
0.6	🟦 **24416** *18424*	▨ **39760** *33040*	🟦 **128320** *108160*	▨ **281600** *200960*
0.7	🟩 **34496** *30800*	🟦 **55120** *46000*	🟦 **177120** *123040*	🟦 **404800** *328960*
0.8	🟦 **42112** *31976*	🟦 **89040** *70880*	▨ **211520** *192480*	🟩 **493760** *435840*

Success rate	
🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.5, The average number of child fitness evaluations calculated by searches using different combinations of *NP* and *F*.

Comparison of the results for baicalein (table 3.1) and adipamide (table 3.4) with equivalent values of *F* (0.1-0.6), demonstrates that the structure of adipamide is solved successfully on average in significantly fewer generations despite searches using a smaller population size. However, a more accurate measure of search efficiency can be achieved by comparison between tables 3.2 and 3.5 respectively showing the results of searches with *NP* = 105 for baicalein and *NP* = 56 for adipamide. The number of models in each population is calculated by multiplying the number of parameters required for model definition by fifteen for baicalein and seven for adipamide.

Table 3.2 shows that the fastest searches used to solve the structure of baicalein with *NP* = 105 and *F* = 0.3 converge with 100% success calculating on average 27615 child fitness evaluations. Table 3.5 shows that the fastest searches used to solve the structure of adipamide with *NP* = 56 and *F* = 0.6 converge with 100% success calculating on average 24416 child fitness evaluations. Thus comparing tables 3.2 and 3.5 shows that the searches for adipamide are not particularly more efficient.

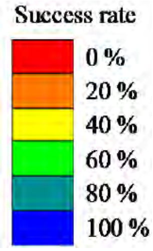
3.3.3 Optimised combination of *NP*, *F* and *NUT* control parameters

The CDE was also run for adipamide, using the same model, data and criteria as the DE calculation, five times for each combination of *NUT*, *NP* and *F*. The results of these searches are shown in Tables 3.6a-d.

Mean G_{con}

(a)

N_{ut} / F	static	1	2	3	4	5	6
0.5	**217** *159*	**246** *164*	**286** *286*	**195** *144*	**0** *0*	**0** *0*	**0** *0*
0.6	**436** *329*	**352** *319*	**356** *334*	**264** *224*	**0** *0*	**0** *0*	**0** *0*
0.7	**616** *550*	**484** *430*	**511** *429*	**332** *332*	**0** *0*	**356** *356*	**0** *0*
0.8	**752** *571*	**732** *610*	**604** *401*	**493** *431*	**322** *307*	**0** *0*	**0** *0*



(b)

Mean G_{con}

N_{ut} / F	static	3	4	5	6	7
0.3	**0** *0*	**103** *103*	**0** *0*	*0* *0*	*0* *0*	*0* *0*
0.4	**224** *182*	**201** *201*	**194** *194*	**186** *186*	**128** *128*	**0** *0*
0.5	**369** *245*	**280** *200*	**0** *0*	**0** *0*	**0** *0*	**260** *260*
0.6	**497** *413*	**407** *412*	**320** *290*	**335** *316*	**0** *0*	**356** *356*
0.7	**689** *575*	**514** *427*	**474** *447*	**444** *340*	**332** *221*	**298** *280*
0.8	**1113** *886*	**857** *722*	**625** *484*	**546** *469*	**475** *454*	**0** *0*

(c)

Mean G_{con}

N N_{ut} / ***F***	static	3	4	5	8	9	10	11	12	13
0.1	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.2	**86** **86**	**116** **70**	**0** *0*	**108** **108**	**82** **82**	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.3	**171** *146*	**189** *158*	**200** *160*	**210** *186*	**160** *139*	**176** *176*	**0** *0*	**0** *0*	**126** *126*	**170** *170*
0.4	**358** *256*	**311** *239*	**335** *272*	**294** *270*	**230** *208*	**253** *216*	**359** *249*	**785** *202*	**0** *0*	**190** *173*
0.5	**516** *462*	**502** *381*	**451** *395*	**485** *391*	**338** *325*	**366** *278*	**232** *232*	**310** *310*	**361** *361*	**287** *287*
0.6	**802** *676*	**841** *640*	**792** *526*	**653** *572*	**518** *491*	**440** *372*	**497** *444*	**312** *299*	**368** *298*	**359** *359*
0.7	**1107** *769*	**1074** *818*	**1207** *837*	**844** *584*	**734** *605*	**521** *468*	**547** *430*	**601** *446*	**846** *338*	**596** *522*
0.8	**1322** *1203*	**1441** *944*	**1294** *959*	**1365** *1082*	**1021** *767*	**982** *798*	**767** *615*	**792** *602*	**590** *558*	**633** *480*

(d)

Mean G_{con}

N_{ut} / ***F***	static	10	12	14	16	18	20	22	24	26	28	30	32
0.1	**52** *49*	**0** *0*	**46** *46*	**68** *68*	**0** *0*	**46** *46*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**40** *40*
0.2	**124** *90*	**107** *99*	**124** *114*	**118** *117*	**116** *88*	**126** *116*	**120** *120*	**101** *101*	**86** *86*	**112** *112*	**0** *0*	**113** *97*	**0** *0*
0.3	**251** *216*	**235** *196*	**191** *140*	**194** *188*	**231** *217*	**219** *219*	**194** *174*	**0** *0*	**219** *219*	**0** *0*	**156** *140*	**183** *183*	**0** *0*
0.4	**371** *357*	**370** *282*	**329** *237*	**275** *236*	**269** *229*	**238** *218*	**281** *239*	**257** *211*	**224** *206*	**212** *212*	**207** *196*	**201** *190*	**0** *0*
0.5	**699** *543*	**628** *391*	**493** *334*	**476** *396*	**455** *406*	**353** *309*	**386** *344*	**328** *270*	**341** *282*	**360** *360*	**335** *335*	**276** *276*	**232** *232*
0.6	**880** *628*	**911** *819*	**844** *560*	**810** *601*	**735** *541*	**527** *408*	**485** *445*	**417** *388*	**386** *336*	**345** *306*	**329** *300*	**323** *282*	**371** *371*
0.7	**1265** *1028*	**1144** *830*	**1107** *888*	**890** *724*	**1042** *754*	**843** *735*	**595** *485*	**548** *456*	**539** *472*	**499** *434*	**455** *410*	**390** *390*	**447** *370*
0.8	**1543** *1362*	**1645** *1305*	**1449** *1309*	**1317** *852*	**1234** *1149*	**844** *703*	**898** *696*	**864** *638*	**687** *534*	**649** *588*	**597** *523*	**465** *442*	**416** *416*

Table 3.6: Structure solution of adipamide by CDE using different combinations of *F* (mutation rate) and *NUT* (cultural pruning parameter) for populations of a)56, b)80, c)160, and d)320. The static column indicates a conventional DE search with no cultural pruning. The success rate over the five calculations for each *NP*, *F* and *NUT* combination is indicated as in table 3.1. The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. For searches performed with *NP*=56 *Gmax*=1500, for all other *NP* values *Gmax*=2000.

Tables 3.6a-d show that in general, cultural searches converge in fewer generations than analogous DE searches, and that unlike static DE searches (table 3.4), cultural searches are significantly more successful if performed with large *NP* and *F*. Table 3.6d shows that DE searches with *F* = 0.6 and 0.8 converge with 80% and 60% success on average in 880 and 1543 generations respectively. CDE searches with F = 0.6 and *NUT* = 26 converge with 100% success on average in 345 generations. CDE searches with *F* = 0.8 and *NUT* = 28 converge with 100% success on average in 597 generations. In these two cases the CDE searches are 61% more efficient than the analogous DE searches.

Since applying cultural pruning to a search decreases the level of genetic diversity in the population, tables 3.6a-d show that as expected, the success rate of cultural searches increases with the size of the mutation rate. Table 3.6b shows that for CDE searches with *NP* = 80, *NUT* = 3 and *F* = 0.3-0.8 searches converge with 20%, 20%, 60%, 80%, 80% and 100% success respectively. Increasing the initial amount of genetic diversity by using larger population sizes also increases the probability that cultural searches converge successfully. CDE searches with *F* = 0.6, *NUT* = 4 and *NP* = 56, 80 and 160 converge with 0%, 40% and 100% success respectively.

Similar to the results presented in tables 3.3a-e, tables 3.6a-d show that the convergence rate of searches generally increases as the value of *NUT* is increased. Table 3.6c shows that CDE searches with *NP* = 160, *F* = 0.5 and *NUT* = 3-13 converge on average in 502, 451, 485, 338, 366, 232, 310, 361 and 287 generations respectively. However, tables 3.6a-d show that the success rate of CDE searches decreases as the value of *NUT* increases. Table 3.6c shows that CDE searches with *NP* = 160, *F* = 0.5 and *NUT* = 3-13 converge with 100%, 100%, 100%, 40%, 100%, 20%, 20%, 20% and 20% success respectively. Comparing tables 3.6c and 3.6d shows that searches using larger populations can be performed with larger *NUT* without compromising success. Table 3.6d shows that CDE searches with *NP* = 320, *F* = 0.5 converge with 100% success *NUT* = 14-18 but only 20% success for *NUT* >=26.

Tables 3.3e and 3.6d respectively show the results of CDE searches used to solve the crystal structures of baicalein and adipamide. The size of each population is calculated by multiplying the number of parameters defining each structural model by 40, giving *NP* = 280 for baicalein and *NP* = 320 for adipamide. Comparing table 3.3e with table 3.6d shows that although the CDE searches use a smaller population to solve the structure of baicalein than adipamide, cultural searches for baicalein can be performed with larger *NUT* (more aggressive pruning) than searches with equivalent *F* for adipamide without compromising success. Table 3.3e shows that CDE searches with *F* = 0.3 converge with 100% success *NUT* = 10, 13, 14, 16, 17, 19, 20, 21, 22 and 26. CDE searches with *F* = 0.6 converge with 100% success *NUT* = 15, 16, 18, 19, 20, 21, 22, 26, 30 and 32.

Table 3.6d shows that CDE searches with *F* = 0.3 converge with 80% success *NUT* = 10, 12, 14 and 16, 20% *NUT* = 18 and 60% success *NUT* = 20. Searches with *F*=0.6 converge with 100% success *NUT* = 10, 12, 14, 16, 18, 20 and 26 and 40% success *NUT* = 28 and 30. This suggests that reducing levels of genetic diversity by encouraging models to cluster increases the probability that CDE searches converge prematurely in local minima.

The actual shape of the landscape representing the crystal structure of adipamide potentially decreases the probability that cultural searches are successful. Table 3.4 demonstrates that the mutation rate of a DE search has a significant influence on the success rate. The use of large *F* decreases the probability that models remain in minima for many generations and forces models to explore the landscape. Since the landscape representing adipamide contains numerous local minima it is likely that during cultural searches performed using small *F*, a significant number of models cluster in local minima. The cultural pruning encourages a search to explore areas of the landscape with a high population density. Thus in this case, pruning discourages models clustered in local minima from leaving and locating the global minimum. This suggests that cultural searches with small *F* are less likely to be more efficient than an analogous static DE search for solving crystal structures represented by R_{wp} landscapes containing numerous local minima.

3.4 The crystal structure solution of acetarsone

3.4.1 Background

Despite its toxicity, acetarsone ((3-acetylamino-4-hydroxyphenyl)arsonic acid) ($C_8H_{10}AsNO_5$) is used to treat parasitic infections, particularly protozoal infections of the intestine and genito-urinary tract. [15] The molecule contains multiple H-bond donors and acceptors, facilitating the possible formation of different H-bonded networks in the crystal structure.

3.4.2 Optimisation of the DE algorithm

The crystal structure of acetarsone was previously determined by the direct space method, [5] and is known to adopt the space group $P2_1$, so the model will hence require nine parameters for definition within the unit cell·

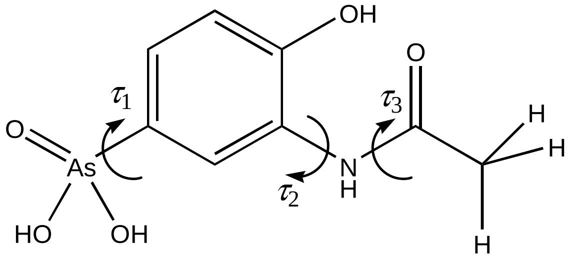


Figure 3.5. The structural model of acetarsone used in the DE calculations. Arrows indicate torsional flexibility, (figure taken from [5]).

In the actual molecule, arsenic forms a pi bond with one oxygen and sigma bonds with the other two oxygen atoms. However, as shown in figure 3.5 (and discussed in section 2.1.2 standard bond lengths are used) and in our model all three As-O bonds are equivalent. This structure solution requires no constraints on the position or orientation of the molecule within the unit cell as the space group symmetry requires one molecule in a general position. The structural model of acetarsone is defined by three parameters to define the position (x,y,z: between 0 and 1), orientation (φ,ψ,θ: between 0 and 360) and torsion parameters (τ1-t3: between 0 and 360) to define the intramolecular rotation between the arsenic acid group and phenyl ring and the flexibility of the acetylamino side chain as shown in figure 3.5. Unlike the previous cases, baicalein and adipamide, acetarsone contains one ‘heavy’ arsenic atom which is a strong X-ray scatterer compared to the rest of the structure. This should mean that structures in which the arsenic atom is in approximately the correct position, are likely to be assigned relatively low *R* factors even if the rest of the organic structure remains non-optimal.

For acetarsone, a successful crystal structure solution has been attained when the model located by the search has an *R* factor R_{wp} <= 14.5%. All models obtained with an *R* factor of this value were judged to be close enough for successful Rietveld refinement. The DE structure solution calculation was run five times for each combination of *NP* and *F* parameters: *NP* = 63, 90, 135 and 180 with *F* = 0.3-0.8 and *NP* = 360 with *F* = 0.1-0.8. For each combination of *NP* and *F*, the *R* factor of the optimum solution located by each of the five searches was recorded. The results of these searches are shown in Table 3.7.

Number of Generations

NP ***F***	63	90	135	180	360
0.1					🟥 **0** *0*
0.2					🟥 **0** *0*
0.3	🟥 **0** *0*	🟥 **0** *0*	🟧 **141** *141*	🟥 **0** *0*	🟦 **258** *213*
0.4	🟥 **0** *0*	🟩 **230** *217*	🟦 **325** *255*	▨ **350** *271*	🟦 **406** *319*
0.5	🟩 **349** *297*	🟦 **420** *376*	🟦 **542** *356*	🟦 **635** *290*	🟦 **963** *557*
0.6	🟦 **420** *373*	▨ **575** *489*	▨ **851** *721*	▨ **722** *622*	🟦 **1373** *942*
0.7	▨ **640** *399*	▨ **956** *702*	▨ **1205** *932*	▨ **1278** *1169*	🟦 **1787** *1264*
0.8	🟦 **1130** *888*	🟦 **1192** *923*	🟩 **1307** *1060*	🟥 **0** *0*	🟩 **1703** *1473*

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.7: Structure solution of acetarsone by DE using different combinations of *NP* (population size) and F (mutation rate). The success rate over the five calculations for each *NP* and *F* combination is indicated in the colour chart as shown in table 3.1. The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. Searches performed with *NP*=63,90 are assigned *Gmax*=1500, *NP*=135,180 are assigned *Gmax*=2000, and *NP*=360 are assigned *Gmax*=3000.

3.4.3 Optimised combination of *NP* and *F* control parameters

Table 3.7 demonstrates that searches with *NP* = 360 are the most successful, searches with *NP* = 360 and *F* = 0.3-0.7 converge with 100% success. However, searches with *NP* = 360 and *F* = 0.1, 0.2 fail to converge, demonstrating that the genetic diversity injected into the population as it evolves by the mutation rate is necessary to prevent premature convergence. This is different to the trend seen with baicalein (table 3.1) in which searches with large *NP* and small *F* have relatively high success rates. In the largest population sizes, 40 times the number of parameters required for model definition, searches with *F* = 0.1 and 0.2 solve the structure of baicalein with 60 and 100% success. This suggests that the R_{wp} landscape representing the crystal structure of acetarsone contains sufficient local minima to prevent searches with small mutation rates from converging in the global minimum.

Examination across a row in table 3.7 shows that increasing population size generally increases the success rate of searches. However, searches with *NP* = 63 are relatively successful once *F* >= 0.5, searches with *NP* = 63 and *F* = 0.6 and 0.8 converge with 100% success. This demonstrates that if a sufficiently large mutation rate is used a large population is not essential. Most searches with the mutation rate >=0.4 are successful, although with higher *NP* and *F* searches are slower to converge so insufficient *Gmax* can significantly reduce the success rate. A high initial level of genetic diversity generated by a large population that is maintained by a large rate of mutation can prevent a search converging within a convenient number of generations and hence *Gmax* was increased to 3000 for the largest *NP* value

As expected, Table 3.7 shows that increasing the mutation rate also increases the number of generations required for convergence. For example, searches with *NP* = 90 and *F* = 0.5-0.7 converge on average in 420, 575 and 956 generations respectively while an increase in population size increases the number of generations required for convergence to 635, 722 and 1278 generations for the same *F* range and *NP* = 180.

Table 3.8 presents the results of these searches in child fitness evaluations rather than in number of generations. This demonstrates that although searches with a larger population and a small *F* converge in fewer generations (e.g: *NP* = 360, *F* = 0.3, 258 generations, 92880 evaluations), searches with a smaller population and larger *F* (*NP* = 63, *F* = 0.6, 420 generations, 26460 evaluations) converge more efficiently, requiring in this case the calculation of 72% fewer child fitness evaluations.

Number of Generations

F \ ***NP***	63	90	135	180	360
0.1					**0** *0*
0.2					**0** *0*
0.3	**0** *0*	**0** *0*	**19035** *19035*	**0** *0*	**92880** *76680*
0.4	**0** *0*	**20700** *19530*	**43875** *22950*	**63000** *48780*	**146160** *114840*
0.5	**21987** *18711*	**37800** *33840*	**48780** *32040*	**114300** *52200*	**346680** *200520*
0.6	**26460** *23499*	**51750** *44010*	**114885** *97335*	**129960** *111960*	**494280** *339120*
0.7	**40320** *25137*	**86040** *63180*	**162675** *125820*	**230040** *210420*	**643320** *455040*
0.8	**71190** *55944*	**107280** *83070*	**176445** *143100*	**0** *0*	**613080** *530280*

Success rate

🟥 0 %

🟧 20 %

🟨 40 %

🟩 60 %

80 %

🟦 100 %

Table 3.8, The average number of child fitness evaluations calculated by searches using different combinations of *NP* and *F*.

3.4.4 Optimised combination of *NP*, *F* and *NUT* control parameters

The CDE was run for acetarsone, using the same model, data and criteria as the DE calculation, five times for each combination of *NUT*, *NP* and *F*. The results of these searches are shown in Tables 3.9a-e.

Mean G_{con}

(a)

N*~ut~** ***F	static	3	5	7	9
0.3	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.4	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.5	**349** *297*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.6	**420** *373*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.7	**640** *399*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.8	**1130** *888*	**0** *0*	**0** *0*	**0** *0*	**0** *0*

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Mean G_{con}

(b)

N*~ut~** ***F	static	1	2	3	4	5	7	9
0.3	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.4	**230** *217*	**242** *241*	**259** *228*	**233** *233*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.5	**420** *376*	**325** *278*	**318** *250*	**317** *218*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.6	**575** *489*	**626** *411*	**624** *296*	**535** *440*	**536** *442*	**0** *0*	**0** *0*	**0** *0*
0.7	**956** *702*	**916** *862*	**935** *674*	**744** *530*	**520** *507*	**0** *0*	**0** *0*	**0** *0*
0.8	**1192** *923*	**1326** *1036*	**1099** *961*	**1056** *890*	**819** *658*	**0** *0*	**0** *0*	**0** *0*

Table 3.9: Structure solution of acetarsone by CDE using different combinations of F (mutation rate) and *NUT* (cultural pruning parameter) for populations of a) 63, b) 90, c) 135, d) 180 and e) 360. The static column indicates a conventional DE search with no cultural pruning. The success rate over the five calculations for each *NP*, *F* and *NUT* combination is indicated in the colour chart (as in Table 3.1). The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. For searches performed with *NP* = 63, 90, 135 *Gma x*= 1500, *NP* = 180 *Gmax* = 2000 and *NP* = 360 *Gmax* = 3000.

(c)

Mean G_{con}

N_{ut} F	static	5	7	9	11	13
0.3	**141** *141*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.4	**325** *255*	**252** *252*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.5	**542** *356*	**412** *328*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.6	**851** *721*	**759** *548*	**487** *487*	**0** *0*	**0** *0*	**0** *0*
0.7	**1205** *932*	**983** *573*	**1297** *1169*	**0** *0*	**0** *0*	**0** *0*
0.8	**1307** *1060*	**1436** *1433*	**1297** *1169*	**0** *0*	**0** *0*	**0** *0*

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

(d)

Mean G_{con}

N_{ut} F	static	4	5	6	7	9	11	13
0.3	**0** *0*	**0** *0*	**162** *162*	**0** *0*	**0** *0*	**0** *0*	**0** *0*	**0** *0*
0.4	**350** *271*	**299** *239*	**296** *219*	**271** *253*	**230** *230*	**0** *0*	**0** *0*	**0** *0*
0.5	**635** *290*	**537** *466*	**483** *321*	**514** *514*	**474** *474*	**0** *0*	**0** *0*	**0** *0*
0.6	**722** *622*	**897** *700*	**765** *526*	**737** *444*	**681** *519*	**738** *721*	**0** *0*	**0** *0*
0.7	**1278** *1169*	**1193** *936*	**1266** *1042*	**1160** *803*	**953** *809*	**1055** *1055*	**0** *0*	**0** *0*
0.8	*0* *0*	**1724** *1520*	**1580** *1296*	**1461** *1258*	**1550** *1313*	**1055** *969*	**610** *610*	**0** *0*

(e)

Mean G_{con}

N_{ut} F	static	5
0.1	**0** *0*	**0** *0*
0.2	**0** *0*	**0** *0*
0.3	**258** *213*	**248** *224*
0.4	**406** *319*	**435** *346*
0.5	**963** *557*	**851** *539*
0.6	**1373** *942*	**1177** *879*
0.7	**1787** *1264*	
0.8	**1703** *1473*	

Success rate

🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

Table 3.9. continued

It is clear that compared to the CDE searches used to solve baicalein and adipamide, the CDE searches solve acetarsone with relatively low success rates and that CDE searches with larger *NP* and *F* are generally more successful. However, table 3.9d shows that if large *F* is combined with large *NP*, CDE searches are frequently terminated by *Gmax*.

Table 3.9b shows that CDE searches with *NUT* = 2 and *F* = 0.4-0.8 converge with 40%, 60%, 100%, 100% and 100% success respectively. Tables 3.9a-e show that CDE searches with *F* = 0.4, *NUT* = 5 and *NP* = 63-360 converge with 0%, 0%, 20%, 60% and 100% success respectively. However, cultural searches with large *F* can calculate more child fitness evaluations than static DE searches using smaller *F*, making the static searches more efficient. Table 3.9c shows that DE searches with *NP* = 135 and *F* = 0.4-0.5 converge with 100% success in 325 and 542 generations (or 43875 and 73170 fitness evaluations). With *NP* = 135, the fastest CDE with 100% success has *F* = 0.7, *NUT* = 5 and converges on average in 983 generations or 132705 fitness evaluations.

Unlike CDE searches for baicalein and adipamide, increasing the value of *NUT* does not necessarily cause the CDE searches to solve the structure of acetarsone in significantly fewer generations. Table 3.9d shows that DE searches with *NP* = 180 and *F* = 0.6 converge with 80% success on average in 722 generations. Analogous CDE searches with *NUT* = 4-9 converge on average in 897, 765, 737, 681 and 738 generations respectively, demonstrating that in this case, only CDE searches with *NUT* = 7 converge on average in fewer generations than the analogous DE search. Comparing tables 3.3d and 3.9d shows that CDE searches with large *NUT* are significantly more efficient and successful at solving the structure of baicalein than acetarsone. In each case the respective crystal structure is solved by CDE searches with *NP* = 140 and 180, 20 times the number of parameters required for model definition.

Table 3.3d shows that CDE searches with *F* = 0.6 generally converge with low success rates until *NUT* >= 8. Due to the large population size and high mutation rate, many CDE searches fail to converge within *Gmax* generations. However, CDE searches with *NUT* = 8-11 converge with 100% success and in fewer generations than the analogous DE search. Table 3.9d shows that CDE searches with *F* = 0.6 generally take longer to converge than the analogous DE search and fail to converge once *NUT* > 9.

Comparing tables 3.3c-e and tables 3.9c-d shows that whereas the success rate of CDE searches for baicalein gradually decrease as *NUT* increases, the success rate of CDE searches for acetarsone seems to abruptly fall once *NUT* increases to a certain value. In general, tables 3.9c-d show that CDE searches with *NUT* >= 7 and *NUT* >= 9 respectively fail to converge.

3.5 Problems with cultural DE

As discussed in section 2.3, the CDE [4,5] determines which areas of the landscape have a high population density by consideration of the position of the children in the landscape. During each generation, the position of each child is recorded regardless of whether the child has a higher fitness value than the parent and is accepted or a lower fitness value and is rejected. At the end of each generation the CDE determines where the children created during the present generation have clustered and those areas with low child population density (determined by *NUT*) are pruned so that the next generation of children cannot be created in these areas.

A comparison of the distribution of children across the landscapes representing the structures of baicalein and adipamide during the CDE searches showed that during the CDE search used to solve the structure of baicalein, the current best model regularly occupies an area of landscape with a high child population density as represented by a histogram bin in the middle of the distribution in Figure 5 in chapter 2. Thus the landscape boundaries lie either side of the area occupied by the best model and cultural pruning encourages the search to explore the landscape occupied by the best model. Conversely, it was found that during the CDE search used to solve the structure of acetarsone, the best model frequently occupies an area of landscape represented by a histogram bin at one end of the distribution with a low child population density. Thus during a CDE search, the pruning frequently removes the bin representing the area of landscape occupied by the best model causing the position of the boundaries to be adjusted to exclude this area. This both increases the number of generations required for convergence compared to a static DE search and also reduces the success rate.

3.6 Alternative CDE implementations

Two alternative implementations of the cultural aspects of CDE were then developed and tested by application to the crystal structure solution of baicalein and acetarsone with a view to establishing an approach in which the cultural pruning could be utilized in a more robust manner.

3.6.1 'ChildBest' CDE

In this first adaption, the original cultural differential evolution [4,5] (now referred to as oCDE) was modified so that the current best model could not be treated as an outlier, i.e: bins representing areas of the landscape occupied by the best model cannot be removed, regardless of their population density. Thus if the bin representing the area of landscape occupied by the current best model has a low child population density and lies at one extreme end of the distribution, the *NUT* pruning criteria is only allowed to remove bins from the opposite end of the distribution, (as a consequence moving the furthest boundary towards the best model.) The landscape boundaries always lie either side of the best model and a search is not prevented from exploring the area of the landscape occupied by the best model. This version of cultural differential evolution is referred to as cbCDE.

3.6.2 'PopulationBest' CDE

The 'original' cultural DE [4,5] controls the position of the landscape boundaries by measuring the position of children created during the previous generation. However, many of these children are immediately discarded because the child has a lower fitness value than the parent. Thus during both oCDE and cbCDE the parameters of rejected children are frequently used to control the position of the landscape boundaries. Hence, an alternative CDE implementation was created in which only accepted children and unbeaten parents are used for cultural pruning.

This version of cultural differential evolution, pbCDE, determines which areas of the landscape have a high population density by consideration of the position of the accepted children and unbeaten parents in the landscape. At the end of each generation, the *N* dimensional record stores the positions of these individuals and the structure parameters in each dimension are placed in sequence of increasing size and sorted into appropriate histogram bins. Bins are then removed as before from each end of the distribution until *NUT* parameters have been pruned from each end of the distribution and the new values define the landscape boundaries. In addition the CDE is forbidden to remove histogram bins that represent areas of the landscape occupied by the best model regardless of the population density of the bin.

3.7 Trial of alternative cultural DE algorithms

3.7.1 The crystal structure solution of baicalein

Searches with *NP* = 140 and *F* = 0.3-0.6 were used to test these alternative CDE approaches and the results are presented in tables 3.10a-c. The relatively large population size was chosen since it increases the initial amount of genetic diversity and decreases the probability of premature convergence. Since the searches were unlikely to converge prematurely through lack of genetic diversity a wide range of NUT values could be trialed with low probability of compromising the success of searches. Searches were run with different *F* values to determine whether the new cultural implementations responded differently to different mutation rates. Each search was run five times for each combination of *F* and *NUT* trialed, and as previously, a solution with an *R* factor <= 15.6% was judged to be a success.

The results shown in tables 3.10a-c demonstrate that the three different variants of CDE converge on average in fewer generations than the analogous static DE. Consideration of the searches with *NP* = 140 and *F* = 0.3 show convergence of the DE with 100% success on average in 346 generations. The oCDE search (*F* = 0.3) with *NUT* = 4 and 8 converges with 80% and 100% success respectively on average in 293 and 249 generations, whereas for the cbCDE searches this is 100% success on average in 319 and 239 generations and the pbCDE searches converge with 100% success on average in 316 and 281 generations respectively. These results demonstrate that the cbCDE and oCDE searches, using the same combination of *NP*, *F* and *NUT*, converge with similar rates of success and in a similar number of generations. These variants of cultural DE use the clustering behaviour of all children to control the movement of the landscape boundaries. As the current best model of baicalein regularly occupies an area of the landscape represented by a histogram bin with a high child population density and towards the centre of the distribution, the cbCDE can regularly remove all the histogram bins to satisfy the *NUT* criteria and prune as in a similar way to the oCDE.

Table 3.10c shows that for pbCDE searches increasing *NUT* does increase the convergence rate but not as significantly as for the other two cultural implementations, implying that pbCDE searches need to be pruned much more vigorously than oCDE and cbCDE searches to gain an equivalent increase in efficiency. However, when cultural searches are performed with *F* = 0.3-0.4 and *NUT* > 10, the oCDE and cbCDE searches frequently converge with less success (in some cases only 20%) than the equivalent pbCDE searches (80 and 100% success). This suggests that using the clustering behavior of the "accepted" children and the unbeaten parents reduces the rate at which the boundaries are pushed inwards on the population and hence the models in the population are not rapidly constrained to only explore a small area of landscape. This reduces the rate at which models cluster and genetic diversity lost.

(a)

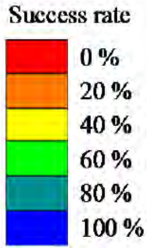
Mean G_{con}

N_{ut} F	static	4	5	6	7	8	10	12	15
0.3	**346** *256*	**293** *240*	**284** *237*	**261** *195*	**245** *224*	**249** *224*	**247** *231*	**225** *196*	**249** *249*
0.4	**825** *766*	**563** *440*	**690** *608*	**545** *371*	**438** *291*	**389** *274*	**355** *308*	**235** *213*	**406** *406*
0.5	**1360** *879*	**1291** *1172*	**989** *773*	**971** *763*	**760** *672*	**593** *446*	**429** *360*	**364** *321*	**268** *257*
0.6	**1471** *1471*	**1658** *1285*	**1613** *1470*	**1487** *1145*	**1225** *905*	**951** *747*	**630** *522*	**470** *377*	**354** *307*

(b)

Mean G_{con}

N_{ut} F	static	4	5	6	7	8	10	12	15
0.3	**346** *256*	**319** *276*	**263** *215*	**289** *198*	**270** *229*	**239** *214*	**249** *223*	**235** *176*	**192** *192*
0.4	**825** *766*	**680** *524*	**589** *465*	**537** *408*	**410** *335*	**383** *304*	**324** *251*	**275** *244*	**233** *233*
0.5	**1360** *879*	**1165** *1014*	**922** *752*	**927** *638*	**699** *612*	**706** *577*	**494** *451*	**351** *332*	**307** *276*
0.6	**1471** *1471*	**1589** *1243*	**1622** *1399*	**1455** *1292*	**1361** *1032*	**989** *762*	**659** *513*	**703** *490*	**419** *374*



(c)

Mean G_{con}

N_{ut} F	static	4	5	6	7	8	10	12	15
0.3	**346** *256*	**316** *171*	**303** *246*	**288** *182*	**294** *209*	**281** *195*	**274** *233*	**242** *170*	**239** *185*
0.4	**825** *766*	**597** *561*	**588** *473*	**624** *447*	**465** *369*	**589** *549*	**558** *434*	**552** *369*	**450** *288*
0.5	**1360** *879*	**1078** *878*	**1109** *947*	**1122** *924*	**1117** *941*	**1105** *874*	**802** *707*	**870** *724*	**768** *663*
0.6	**1471** *1471*	**1353** *1300*	**1654** *1393*	**1561** *1462*	**1324** *886*	**1525** *1341*	**1373** *1089*	**1347** *1032*	**1222** *824*

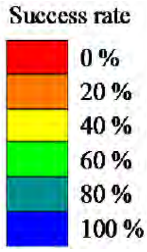


Table 3.10: Crystal structure solution of baicalein using a) ‘Original’, b)‘ChildBest’ and c)‘PopulationBest’ CDE. The success rate over the five calculations for each *F* and *NUT* combination is indicated in the colour chart (as in table 3.1). The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. All searches were performed using *Gmax*=2000.

3.7.2 The crystal structure solution of acetarsone

Structure solution tests on acetarsone were performed by searches with *NP* = 180 and *F* = 0.3-0.8 and the results of these searches presented in tables 3.11a-c. The relatively large population size was chosen since it increases the initial amount of genetic diversity, decreases the probability of premature convergence and allows a wide range of *NUT* values to be trialed without compromising the success of searches. Searches were run with different *F* values to determine whether the new cultural implementations responded differently to different mutation rates. Each search was run five times for each combination of *F* and *NUT* trialed and the search judged to be a success if the *R* factor of the final solution was <= 14.5%.

Table 3.11a shows that oCDE searches are capable of solving the crystal structure of acetarsone but only with high success if *F*>=0.6. However, the use of large *F* slows the rate of convergence, and hence the oCDE searches do not provide greater efficiency than the static DE searches. Table 3.11b demonstrates that cbCDE searches that are prevented from pruning the area of landscape occupied by the best model have higher success rates than oCDE searches. However, many of the results in this table show that the cbCDE searches require more generations to converge than the analogous static DE searches. It is only the cbCDE searches with *F*=0.5 that converge with high success rates and in fewer generations than the static DE search.

The pbCDE searches with small *F* shown in table 3.11c are more likely to solve the crystal structure of acetarsone than oCDE and cbCDE searches. Table 3.11c shows that pbCDE searches with *F* = 0.4 converge with 100% success on average in 337 and 286 generations with *NUT* = 4 and 6 respectively. This suggests that using the clustering behavior of the accepted children and unbeaten parents to control the movement of the landscape boundaries results in a search that is not so aggressively pruned compared to the other two cultural implementations that use the clustering behavior of all children. With less vigorous pruning the pbCDE searches do not converge in significantly fewer generations than analogous DE searches for example, *F* = 0.7 over the range *NUT* = 4-9. Table 3.11c demonstrates that although the pbCDE search converges with greater success than the oCDE and cbCDE cultural implementations, the pbCDE does not frequently converge in significantly fewer generations than analogous DE searches. Thus the pbCDE cultural implementation is not significantly more efficient than static DE searches when used to solve the structure of acetarsone.

(a)

Mean G_{con}

N_{ut} F	static	4	5	6	7	9	11
0.3	🟥 **0** *0*	🟥 **0** *0*	🟧 **162** *162*	🟥 **0** *0*	🟥 **0** *0*	🟥 **0** *0*	🟥 **0** *0*
0.4	▨ **350** *271*	🟩 **299** *239*	🟩 **296** *219*	🟨 **271** *253*	🟧 **230** *230*	🟥 **0** *0*	🟥 **0** *0*
0.5	🟦 **635** *490*	🟦 **537** *466*	🟦 **483** *321*	🟧 **514** *514*	🟧 **474** *474*	🟥 **0** *0*	🟥 **0** *0*
0.6	▨ **722** *622*	🟦 **897** *700*	🟦 **765** *526*	🟦 **737** *444*	▨ **681** *519*	🟩 **738** *721*	🟥 **0** *0*
0.7	▨ **1278** *1169*	▨ **1193** *936*	▨ **1266** *1042*	▨ **1160** *803*	🟦 **953** *809*	🟧 **1055** *1055*	🟥 **0** *0*
0.8	🟥 **0** *0*	🟨 **1724** *1520*	🟨 **1580** *1296*	▨ **1461** *1258*	🟦 **1550** *1313*	▨ **1055** *969*	🟧 **610** *610*

Success rate	
🟥	0 %
🟧	20 %
🟨	40 %
🟩	60 %
▨	80 %
🟦	100 %

(b)

Mean G_{con}

N_{ut} F	static	4	5	6	7	9	11
0.3	🟥 **0** *0*	🟨 **204** *195*	🟥 **0** *0*	🟥 **0** *0*	🟥 **0** *0*	🟥 **0** *0*	🟥 **0** *0*
0.4	▨ **350** *271*	🟩 **315** *249*	🟨 **352** *331*	▨ **399** *313*	🟧 **282** *282*	🟥 **0** *0*	🟥 **0** *0*
0.5	🟦 **635** *490*	🟦 **583** *477*	🟦 **610** *404*	▨ **463** *378*	🟩 **456** *431*	🟧 **366** *366*	🟥 **0** *0*
0.6	▨ **722** *622*	🟦 **802** *500*	🟦 **861** *655*	▨ **835** *480*	▨ **753** *588*	🟧 **550** *550*	🟥 **0** *0*
0.7	▨ **1278** *1169*	🟦 **1105** *946*	🟦 **1294** *1015*	▨ **1277** *977*	🟩 **1159** *912*	▨ **864** *714*	🟥 **0** *0*
0.8	🟥 **0** *0*	🟦 **1626** *1357*	▨ **1527** *1473*	▨ **1589** *1033*	🟦 **1365** *838*	🟦 **1158** *751*	🟥 **0** *0*

(c)

Mean G_{con}

N_{ut} F	static	4	5	6	7	9	11
0.3	🟥 **0** *0*	🟩 **166** *154*	🟧 **210** *210*	🟥 **0** *0*	🟧 **146** *146*	🟥 **0** *0*	🟨 **197** *180*
0.4	▨ **350** *271*	🟦 **337** *248*	▨ **351** *277*	🟦 **286** *260*	🟩 **356** *254*	▨ **267** *236*	🟥 **0** *0*
0.5	🟦 **635** *490*	🟦 **611** *480*	🟦 **448** *344*	▨ **392** *357*	🟦 **574** *509*	🟩 **501** *367*	🟧 **265** *265*
0.6	▨ **722** *622*	🟦 **819** *712*	▨ **869** *630*	🟦 **831** *542*	🟦 **901** *613*	🟦 **632** *573*	🟩 **552** *431*
0.7	▨ **1278** *1169*	🟦 **1236** *1033*	🟦 **1248** *863*	🟦 **1119** *985*	🟦 **1079** *984*	🟦 **1127** *1004*	🟩 **1002** *746*
0.8	🟥 **0** *0*	🟦 **1543** *1211*	🟦 **1585** *1353*	▨ **1585** *1079*	🟦 **1437** *1069*	🟦 **1295** *1083*	🟦 **1307** *1051*

Table 3.11: Crystal structure solution of acetarsone using a)'Original', b)'ChildBest' and c)'PopulationBest' CDE. The success rate over the five calculations for each F and NUT combination is indicated in the colour chart (as in table 3.1). The number in bold is the average number of generations required for convergence over the five runs. The number in italics is the number of generations required for convergence in the quickest run within each group of five. All searches were performed using *Gmax*=2000.

3.8 Auspicious cultural pruning

During the initial generations of a search it is unlikely that many models cluster near the global minimum. If cultural pruning is initiated prematurely it increases the probability that models near the global minimum are treated as outliers and a search is discouraged from exploring near the global minimum. To prevent this, pruning should only be initiated once a cluster of models has formed near the global minimum. However, if pruning is initiated too late, many models are already clustered near the global minimum thus the use of cultural pruning does not significantly reduce the number of generations required by a search to converge.

In this thesis and in the previous work, [4,5] pruning is initiated at generation 50 regardless of the clustering of the models. It was found [4,5] that initiating pruning at this arbitrary number of generations caused the original implementation of cultural searches to reliably converge in fewer generations than static DE searches. However, it is likely that the solution of different crystal structures or the use of different combinations of *NP*, *F* and *NUT* will alter the clustering behavior of models during a search and hence the arbitrary initiation of pruning at generation 50 may not be optimal for all cultural searches.

In order to test this theory, the original cultural implementation described in [4,5] and in section 2.3 was used to solve baicalein and adipamide and pruning initiated after different generations. The generation at which pruning is initiated is defined by the user controlled parameter PruneStart. For each structure solution, pruning was initiated after 1, 25, 50, 100 and 200 generations. During this test, CDE searches used to solve the different structures are not assigned identical combinations of *F*, *NP* and *NUT*; instead each CDE search is assigned a combination of *NP*, *F* and *NUT* that increases the probability that searches solve the particular structure.

3.8.1 The crystal structure solution of baicalein

The structure solution of baicalein was performed with *NP* = 140, *F* = 0.3 and *NUT* = 8 in which average convergence was after 346 generations for the static DE and 249 for oCDE initiating pruning at generation 50 (as shown in table 3.3d). Five structure solution calculations were performed for each value of PruneStart. As previously discussed in section 3.2.2, searches that converge locating a solution that is assigned an *R* factor <= 15.6% are judged successful. The average number of generations required for successful convergence is calculated and presented in table 3.12. Table 3.12 shows that delaying the initiation of pruning increases the average number of generations required by a search to converge. Searches that initiate pruning after 25 generations converge in the smallest number of generations with 100% success. Comparison of the average number of generations required by a search to converge, shows that searches that initiate pruning after 25 generations converge in significantly fewer generations than searches that initiate pruning after 50, 100 and 200 generations. Searches that initiate pruning after 50, 100 and 200 generations require on average approximately the same number of generations to converge. This behaviour suggests that initiating pruning after 50 generations does not guide a significant number of models towards the global minimum. It suggests that after 50 generations a significant number of models in a population with *NP* = 140 and *F* = 0.3 are already clustered near the global minimum. Thus table 3.12 suggests that the optimal generation at which to initiate cultural pruning for structure solution of baicalein using *NP* = 140, *F* = 0.3 and *NUT* = 8 is between generation one and fifty.

	Number of Generations					
Prune Start	Static DE	1	25	50	100	200
	346	**140**	**185**	**249**	**248**	**274**

Table 3.12: Structure solution of baicalein using CDE in which cultural pruning is initiated after different generations. Each column denotes the generation at which pruning was initiated. The success rate over the five calculations is indicated in the colour chart (as in table 3.1) and the number in bold is the average number of generations required for convergence over the five runs.

3.8.2 The crystal structure solution of adipamide

The structure solution of adipamide was performed with *NP* = 160, *F* = 0.5 and *NUT* = 5 in which average convergence was after 516 generations for the static DE and 485 for oCDE initiating pruning at generation 50 (as shown in table 3.6c). Five structure solution calculations were performed for each value of PruneStart. As previously discussed section 3.3.1, searches that converge locating a solution that is assigned an *R* factor <=16.0% are judged successful. The average number of generations required for successful convergence is calculated and presented in table 3.13.

Number of Generations

Prune Start	Static DE	1	25	50	100	200
	516	**490**	**378**	**485**	**443**	**454**

Table 3.13: Structure solution of adipamide using CDE in which cultural pruning is initiated after different generations. Each column denotes the generation at which pruning was initiated. The success rate over the five calculations is indicated in the colour chart (as in table 3.1) and the number in bold is the average number of generations required for convergence over the five runs.

Table 3.13 shows that searches initiating pruning after 25 generations converge in the smallest number of generations with 100% success and that searches initiating pruning after 50, 100 and 200 generations require more generations to converge. However, searches that initiate pruning after one generation require more generations to converge than any other cultural search. This suggests that initiating cultural pruning after one generation frustrates the exploration of the landscape and the identification of the global minimum. Since the R_{wp} landscape representing the crystal structure of adipamide contains numerous local minima it is likely that during the initial generations of a search many models that are assigned relatively low *R* factors occupy local minima. If cultural pruning is initiated after one generation it will detect this and restrict the search to explore these local minima. As a consequence a search is discouraged from exploring the whole landscape and identifying the global minimum. Table 3.13 demonstrates that cultural pruning should not be initiated after the first generation of a search when used to solve the structure of adipamide and that the ideal generation at which to initiate pruning for structure solution of adipamide using *NP* = 160, *F* = 0.5 and *NUT* = 5 is between generation two and fifty.

3.8.3 Summary

In order to maximize the potential of a cultural search, it is necessary to identify the optimal generation at which to initiate pruning. Ideally pruning is initiated once a certain number of models in a population have achieved a relatively low R factor as this indicates that a number of models have located one or more deep minima that have a high probability of being the global minimum.

3.9 Conclusions

Sections 3.2.3, 3.3.2 and 3.4.3 demonstrate that certain combinations of *NP* and *F* do increase the probability that a DE search solves a particular crystal structure and can significantly influence the efficiency of the search. However, the results presented in these sections demonstrate that there is no ‘universal’ optimal combination of *NP* and *F*. Tables 3.1 and 3.2 demonstrate that if a crystal structure is likely to be represented by an R_{wp} landscape with few local minima it is more efficient to solve the structure using a DE search with a moderate population size and relatively small *F* since this increases the probability that a search solves the structure calculating as few child fitness evaluations as possible. However, if a crystal structure is likely to be represented by an R_{wp} landscape containing numerous local minima tables 3.4 and 3.5 for adipamide and 3.7 and 3.8 for acetarsone show that it is more efficient to use a small population and moderate to large *F* since this decreases the probability that many models explore local minima, increases the probability that the search converges in the global rather than a local minimum and calculates fewer child fitness evaluations.

Although the addition of culture to a search can significantly reduce the number of generations required by a search to converge, aggressive over-pruning can reduce the probability that a search converges successfully and it may additionally increase the number of generations required for convergence compared to an analogous static DE search. If no consideration is given to the relationship between the location of the best model in the landscape and where children are clustering, the cultural search can prune the area of landscape occupied by the best model from the search, preventing successful convergence. Cultural searches that identify the area of landscape occupied by the best model and do not prune this area of landscape from the search, are significantly more likely to converge in the global minimum.

No definite conclusions can be drawn concerning whether searches that consider the clustering behaviour of all children to control the position of the boundaries are more efficient than searches that consider the clustering of only accepted children and unbeaten parents. Cultural pruning should not be applied to a search in an arbitrary manner. Table 3.12 demonstrates that if pruning is initiated too late its use does not significantly increase the efficiency of a cultural search compared to an analogous static DE search since many models are already clustered near the global minimum. However, table 3.13 demonstrates that if pruning is initiated prematurely, models may be prevented from reaching the global minimum and encouraged to cluster in local minima.

References

[1] C. C. Seaton and M. Tremayne, POSSUM. Programs for Direct-Space Structure Solution from Powder Diffraction Data. PhD Thesis. School of Chemistry. University of Birmingham UK. (2002).

[2] C. C. Seaton and M. Tremayne. Differential Evolution, Crystal Structure Determination of a Triclinic Polymorph of Adipamide from Powder Diffraction Data. *Chem. Comm*. (2002). 880.

[3] M. Tremayne, C. C. Seaton and C. Glidewell. Structures of Three Substituted Arenesulfonamides From X-ray Powder Diffraction Data Using the Differential Evolution Technique. *Acta Cryst B.* (2002). **58**. 823.

[4] S. Y. Chong and M. Tremayne. Combined Optimisation using Cultural and Differential Evolution Application to Crystal Structure Solution from Powder Diffraction Data. *Chem. Comm*. (2006). 4078.

[5] S. Y. Chong and M. Tremayne. Development of Novel Evolutionary Algorithms for Crystal Structure Determination from Powder Diffraction Data. PhD Thesis. School of Chemistry. University of Birmingham UK. (2006).

[6] Z. H. Shao, C. Q. Li, T. L. Vanden Hoek, L. B. Becker, P. T. Schumacker, J. A. Wu, A. S. Attele and C.-S. Yuan. Extract from Scutellaria Baicalensis Georgi Attenuates Oxidant Stress in Cardiomyocytes. *J. Mol. Cell*. Cardiol. (1999). **31**. 1885.

[7] Z. Gao, K. Huang, X. Yang and H. Xu. Free Radical Scavenging and Antioxidant Activities of Flavonoids Extracted from the Radix of Scutellaria Baicalensis Georgi. *Biochim. Biophys. Acta*. (1999). **1472**. 643.

[8] D. L. Evers, C. F. Chao, X. Wang, Z. Zhang, S. M. Huong and E. S. Huang. Human Cytomegalovirus-inhibitory Flavonoids: Studies on Antiviral Activity and Mechanism of Action. *Antivir. Res*. (2005). **68**. 124.

[9] Y. C. Shen, W. F. Chiou, Y. C. Chou and C. F. Chen. Mechanisms in Mediating the Anti-inflammatory Effects of Baicalin and Baicalein in Human Leukocytes. *Euro. J. Pharmacol*. (2003). **465**. 171.

[10] R. Miocinovic, N. P. McCabe, R. W. Keck, J. Jankun, J. A. Hampton and S. H. Selman. In Vivo and in Vitro Effect of Baicalein on Human Prostate Cancer Cells. *Int. J. Oncol*. (2005). **26**. 241.

[11] M. Bonham, J. Posakony, I. Coleman, B. Montgomery, J. Simon and P. S. Nelson. Characterization of Chemical Constituents in Scutellaria Baicalensis with Antiandrogenic and Growth-Inhibitory Activities toward Prostate Carcinoma. *Clin. Cancer*. Res. (2005). **11**. 3905.

[12] M. Rossi, R. Meyer, P. Constantinou, F. Caruso, D. Castelbuono, M. O'Brien and V. Narasimhan. Molecular Structure and Activity Toward DNA of Baicalein, a Flavone Constituent of the Asian Herbal Medicine "Sho-saiko-to". *J. Nat. Prod*. (2001). **64**. 26.

[13] D. E. Hibbs, J. Overgaard, C. Gatti and T. W. Hambley. The Electron Density in Flavones I. Baicalein. *New. J. Chem*. (2003). **27**. 1392.

[14] G. E. Engel, S. Wilke, O. K. Nig, K. D. M. Harris and F. J. Leusen. Powdersolve, a Complete Package for Crystal Structure Solution from Powder Diffraction Patterns. *J. Appl. Cryst*. (1999). **32**. 1169.

[15] Merriam Webster medical dictionary. www.merriam-webster.com/medical/acetarsones

Chapter 4. Eugenic differential evolution.

Eugenics, noun.	*'The science of improving stock, whether human or animal.' Webster Unabridged Dictionary. (1913).*
Eugenics, noun.	*'Selective breeding as proposed human improvement.' Encarta pocket dictionary. Microsoft corporation. (1999).*

4.1 Stages in the evolution of a population

4.1.1 Initial population.

As previously discussed in sections 1.3.2, 2.2.1 and 2.4 of this thesis, the size and dimensionality of the landscape representing a particular direct space structure solution problem is determined by the number of parameters used to define the position, orientation and conformation of a model in the unit cell, not the size of the population. When a large population is initialised in a particular landscape, the landscape has a higher population density than when a smaller population is used. When a population is initialised, models are generated at random across the landscape and hence as the population size is increased, the probability that an initial model is randomly generated near the global minimum increases. A model that is generated near the global minimum has a higher probability of accessing the global minimum in fewer generations than a model that is generated further away.

4.1.2 Exploration of a landscape

4.1.2.1 Local minima act as traps

As models explore the landscape, fitter models (with similar gene values) cluster in minima. If a parent and three random models are selected from one cluster to create a child, the child will likely have similar gene values and be created in or near the same cluster and hence have a high probability of being created in the same minimum. A child that is fitter than the parent is likely to occupy a deeper part of the minimum and the search will proceed deeper in this direction, whereas a child nearer the lip of the minimum is likely to be less fit than the parent and be rejected. This interbreeding between models clustered in a minimum causes the genetic diversity to decrease and drives the models to the bottom of the minimum reducing the probability that the search converges in the global minimum.

4.1.2.2 Escape

A model trapped in a local minimum can escape by breeding a child that is both fitter and does not occupy the same local minimum. This can be done in two ways:

(a) If the mutation vector is sufficiently long, the child has a low probability of being created near the parent. If the child is fitter than the parent, the parent is replaced by the child and relocates to a distant part of the landscape. Therefore a large *F* value increases the probability that the child structures can escape local minima.

(b) A larger population size initially increases the number of genetically diverse models in the landscape and the probability that the parent and randomly selected models used to create a child occupy different areas of the landscape. If two models used to calculate the mutation vector for a child occupy distant parts of the landscape, the mutation vector will be long regardless of the *F* value, increasing the probability that the child is created at a considerable distance from the parent and increasing the probability that models can escape local minima. In addition, a larger population size reduces the probability that a significant number of models cluster in one local minimum and hence reduces the probability that a child is created using models that occupy one local minimum. Thus for larger population sizes the minimum value of *F* required to prevent premature convergence decreases. Table 3.1 clearly demonstrates that DE searches with large *NP* and small *F* converge with greater success than searches with small *NP* and large *F*.

4.1.3 Final solution

As the number of models in a population with relatively high fitness (near the global minimum) increases, the probability that models near the global minimum interbreed also increases. This means that children can be created close by and rapidly increase the population density around the global minimum. Models with relatively low fitness (at a considerable distance from the global minimum) that breed with models near the global minimum are likely to create children nearer the global minimum and the proportion of the population near the global minimum increases. This process is illustrated in figures 3.2 and 3.3, by the rapid increase in the convergence rate of the search as it nears completion.

As discussed in section 2.2.4, DE is a vector-based global optimisation algorithm that autonomously regulates the area of landscape searched in each generation according to the extent of population convergence. Thus whilst using the same values of *K* and *F*, a DE search can initially rapidly explore a significant area of the landscape but once models cluster near the global minimum concentrate around the global minimum. If a parent and three random models clustered near the global minimum are selected to create a child, the recombination and mutation vectors will be relatively short and the child created near the parent and global minimum. However, the rate at which the search self-scales is influenced by the values of *K* and *F*, the child will be created much nearer the parent if *K* and *F* are small. As discussed in section 3.1.2, *K* = 0.99 is optimal for structure solution calculations. This means that once a significant proportion of models in the population are clustered near the global minimum the value of *F* significantly affects the rate at which the population density increases around the global minimum. As a result, once the global minimum has been located by a proportion of the population, searches with smaller F converge faster than searches with larger *F*.

4.1.4 The effect of increasing population size and decreasing mutation rate

The following two figures show the evolutionary progress of two searches using different *NP* and *F* values. Figure 4.1 shows the progress of a DE search with *NP* = 280 and *F* = 0.1, whereas figure 4.2 illustrates a search with *NP* = 70 and *F* = 0.5.

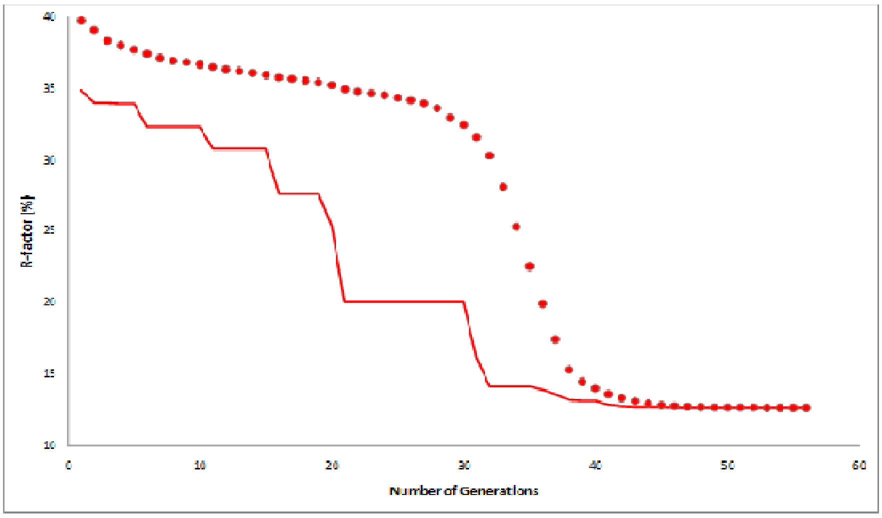


Figure 4.1, Progress of a DE search used to solve the crystal structure of baicalein with *NP* = 280 and *F* = 0.1.

The initial 280 models generated at random have a mean *R* factor of 40.62% with the best initial model at 36.04%. After the search has progressed for 20 generations with *F* = 0.1, the mean *R* factor of the population has decreased to 35.22% while the *R* factor of the best model has decreased to 25.34%. After 40 generations the progress plot shows that the search has completed the stage of rapid convergence and the mean *R* factor is 13.99% with the *R* factor of the best model at 13.11%. After 40 generations the search is close to converging on the global minimum.

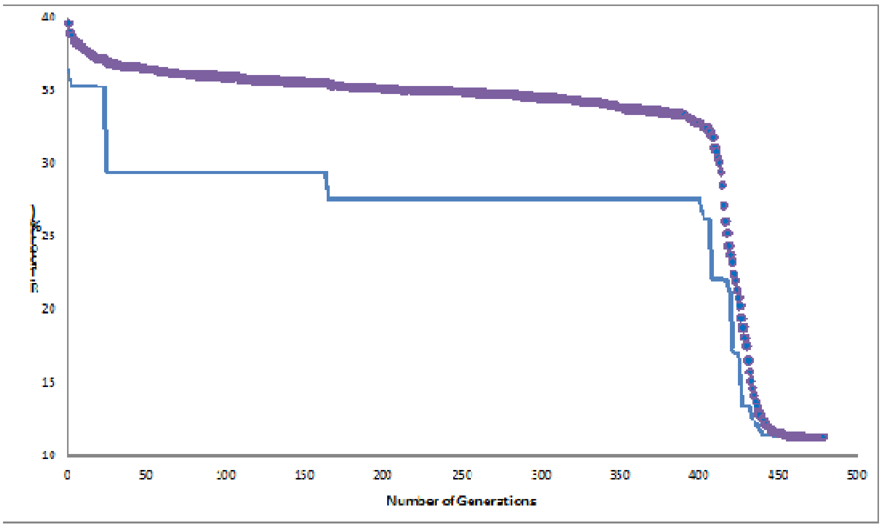


Figure 4.2, Progress of a DE search used to solve the crystal structure of baicalein with *NP* = 70 and *F* = 0.5.

Figure 4.2 shows that with the smaller population, the initial 70 models have a mean *R* factor of 40.71%. The best initial model has an *R* factor of 37.06% and with *F* = 0.5 this decreases to 35.29% after 20 generations with the mean *R* factor of the population at 37.22%. After 40 generations the mean *R* factor has only decreased to 36.65% and the best model decreased to 29.46%; not even near the global minimum. It is only at about 400 generations that rapid convergence of the population begins.

In this case the search with *NP* = 280 not only converges in fewer generations than the search with *NP* = 70, it also converges in less time, requiring the calculation of 15680 child fitness evaluations, whereas the search with *NP* = 70 calculates 33530 evaluations.

4.1.5 Summary

As the population size increases the probability that a significant proportion of models in the population cluster in one local minimum decreases. Thus searches with larger *NP* are less likely to converge prematurely, even if *F* is assigned a small value. For a particular problem, the larger the population size the higher the population density in the landscape and the greater the probability that an initial model is generated near the global minimum and accesses the global minimum in relatively few generations. Once the best model accesses the global minimum it can breed with other models and cause children to be generated near the global minimum, rapidly increasing the local population density. Thus as the population size increases the probability that a significant proportion of models are located near the global minimum after relatively few generations increases. Once a certain proportion of models are clustered near the global minimum the search is prevented from converging in a local minimum, and the size of the mutation rate only affects the rate of convergence on the global minimum. If *F* is assigned a smaller value, parents near the global minimum can breed children that are more likely to be nearer the global minimum than if *F* is assigned a larger value. Since searches with larger *NP* are less likely to converge prematurely it becomes advantageous to assign searches with large *NP* small *F* as this increases the convergence rate. However, as demonstrated in chapter 3, due to the number of child fitness evaluations calculated by a search, searches with larger population sizes and small mutation rates are generally slower to converge in real time than searches with smaller population sizes and larger mutation rates.

4.2 Eugenic DE

4.2.1 Principles of eugenic DE

It is this ability of a search with large *NP* and small *F* to locate the global minimum in few generations by ‘saturating’ a landscape with models, combined with the ability of a search with small *NP* and large *F* to converge on the global minimum requiring the calculation of significantly fewer child fitness evaluations that is exploited to create the eugenic differential evolution approach.

The eugenic DE has two basic principles:

(a) The search is initiated using a large primary population and assigned a small mutation rate. This creates a high population density in the landscape, increasing the probability that an initial model is generated near the global minimum. The small mutation rate causes the DE to act as an R factor minimisation algorithm so that the models ‘fall’ into the nearest minima.

(b) Once a certain proportion of models in the primary population have a relatively low *R* factor, a smaller secondary population is initiated and populated with the fittest models selected from the primary population. A proportion of the models in the primary population with the lowest *R* factor are transferred into the secondary population; the remaining models in the primary population are discarded. The smaller secondary population is assigned a larger mutation rate to reduce premature convergence. This encourages the search to explore the landscape further. However, as a significant proportion of the secondary population are already located in deep minima the search is biased to explore these favoured areas of the landscape and there is a high probability that it will converge successfully in fewer generations. Combining these evolutionary features in this way, a eugenic DE search can converge requiring significantly fewer child fitness evaluations (hence in less real time) than a traditional DE search with fixed *NP* and *F*.

4.2.2 Auspicious population pruning

If the primary population is pruned too early, the secondary population will be created containing few models clustered near the global minimum. As a result, it is unlikely that the search will be sufficiently biased to only explore the landscape around the global minimum and the secondary search will spend additional generations needlessly exploring local minima, reducing the efficiency of a eugenic search. However, if the primary population is pruned too late, many models will have already located the global minimum by the time the secondary population is created, but with a large primary population, a large number of child fitness evaluations will have been calculated, again reducing the efficiency of a eugenic search. The eugenic DE should prune a primary population as soon as a sufficient number of models are clustered near the global minimum, but this is dependent on the complexity of the particular problem, which in turn is affected by the number of parameters defining a model and the number and shape of local minima in the landscape. A search is likely to require a greater number of generations to optimise a model defined by many parameters and if the landscape contains numerous local minima. Hence the optimal moment to prune must be established by following the evolutionary process and must be computer controlled.

4.2.3 Determining the complexity of a crystal structure solution problem

A characteristic that can be used to indicate the complexity of a particular structure solution problem is the difference between the *R* factor assigned to the initial best model and the mean *R* factor assigned to the whole initial population. If the structure solution is relatively complex, it is unlikely that many of the randomly generated initial models will have optimal combinations of parameters and hence the mean *R* factor of the initial population is likely to be significantly greater than that of the initial best model. Conversely if the structure solution problem is relatively simple, it is likely that the mean *R* factor and that of the initial best model have more similar values. As the number of initial models with better combinations of parameters increases, the number of generations required for sufficient clustering of the primary population decreases. The *R* factor difference will be investigated as a potential indicator as to when a primary population is pruned.

4.2.4 The pruning criteria

It is important to establish not only when but by how much the population should be pruned. In this eugenic DE, the *R* factor assigned to the initial best model in the primary population is set as the 'target *R* factor'. As the primary population evolves, more models are assigned an *R* factor <= the 'target *R* factor'. Once a certain proportion of the population (defined by the user) is assigned an *R* factor <= the 'target *R* factor', the primary population is pruned and the secondary population generated. As the complexity of a crystal structure solution increases, the number of generations required for this proportion to be reached also increases and thus the primary population evolves for a greater number of generations before it is used to bias the secondary search towards the global minimum.

In this implementation, the number of models in the primary population required to have an *R* factor <= the 'target *R* factor' before pruning, is automatically calculated from the value of the user defined parameter 'RequireFrac.' A 'RequireFrac' value of 0.25 means that the primary population is pruned when 0.25 of the population transferred into the secondary population have an *R* factor <= the 'target *R* factor'. A 'RequireFrac' value of 0.5 means that half of the models transferred into the secondary population have an *R* factor <= the 'target *R* factor' and hence as 'RequireFrac' is increased, so is the bias of the secondary search.

4.2.5 Determining the size of the primary and secondary populations

The results from the DE searches in chapter 3 (table 3.1 for baicalein and table 3.4 for adipamide) both show the most successful combinations of *NP* and *F* parameters. For larger *NP* (baicalein *NP* = 280, adipamide *NP* = 320) and smaller *F* (0.1-0.3) searches can rapidly locate the global minimum of a landscape and therefore a suitable *NP* for the primary population will be calculated by multiplying the number of parameters required for structural model definition by 40.

For smaller *NP* (baicalein *NP* = 70, adipamide *NP* = 80) successful convergence was obtained in general with *F* >= 0.5, and hence with these control parameters the secondary population should have a relatively high probability of converging successfully. Therefore the secondary *NP* will be calculated by multiplying the number of parameters required for structural model definition by 10.

4.3 Crystal structure solution by eugenic DE

Appendix B shows the eugenic DE subroutine, written in the Perl language.

4.3.1 Initial test

The eugenic DE was first tested and used to solve the crystal structure of adipamide. A detailed discussion of the events occurring during one test structure solution calculation using the eugenic DE is presented here and illustrated by figure 4.3. Figure 4.3 shows the convergence of the population in a similar way to figures 4.1 and 4.2.

As in section 3.3.1, eight parameters are used to define the structural model of adipamide, hence primary *NP* = 320 and primary *F* = 0.1, secondary *NP* = 80 and secondary *F* = 0.6, with 'RequireFrac' = 0.25. Hence in this calculation, the primary population is pruned at the end of a generation once a minimum of (80x0.25) = 20 models are assigned an *R* factor <= the 'target *R* factor'. Figure 4.3a shows that the initial best model in the primary population is assigned an *R* factor = 36.93%. Therefore for this structure solution calculation, the 'target *R* factor' is set to 36.93%. The inset histogram in figure 4.3a shows that after five generations, three models have been assigned an *R* factor<=the 'target *R* factor' whereas after six generations this number has risen to seven models. After eight generations, 23 models have been assigned an *R* factor <= 36.93% and pruning is initiated. The best 80 members of the population (those with the lowest *R* factors, including the 23 models with *R* factors <= the target factor) are transferred to the secondary population; the 240 models with the highest *R* factors are discarded and hence the remainder of the eugenic calculation progresses with a population of 80. The secondary search then converges successfully (figure 4.3b) with the entire eugenic process taking a total of 192 generations. The primary population evolves for eight generations whilst the secondary population evolves for 184 generations. Thus during the whole search, the eugenic DE calculates a total of (320x8) + (80x184) = 17280 child fitness evaluations.

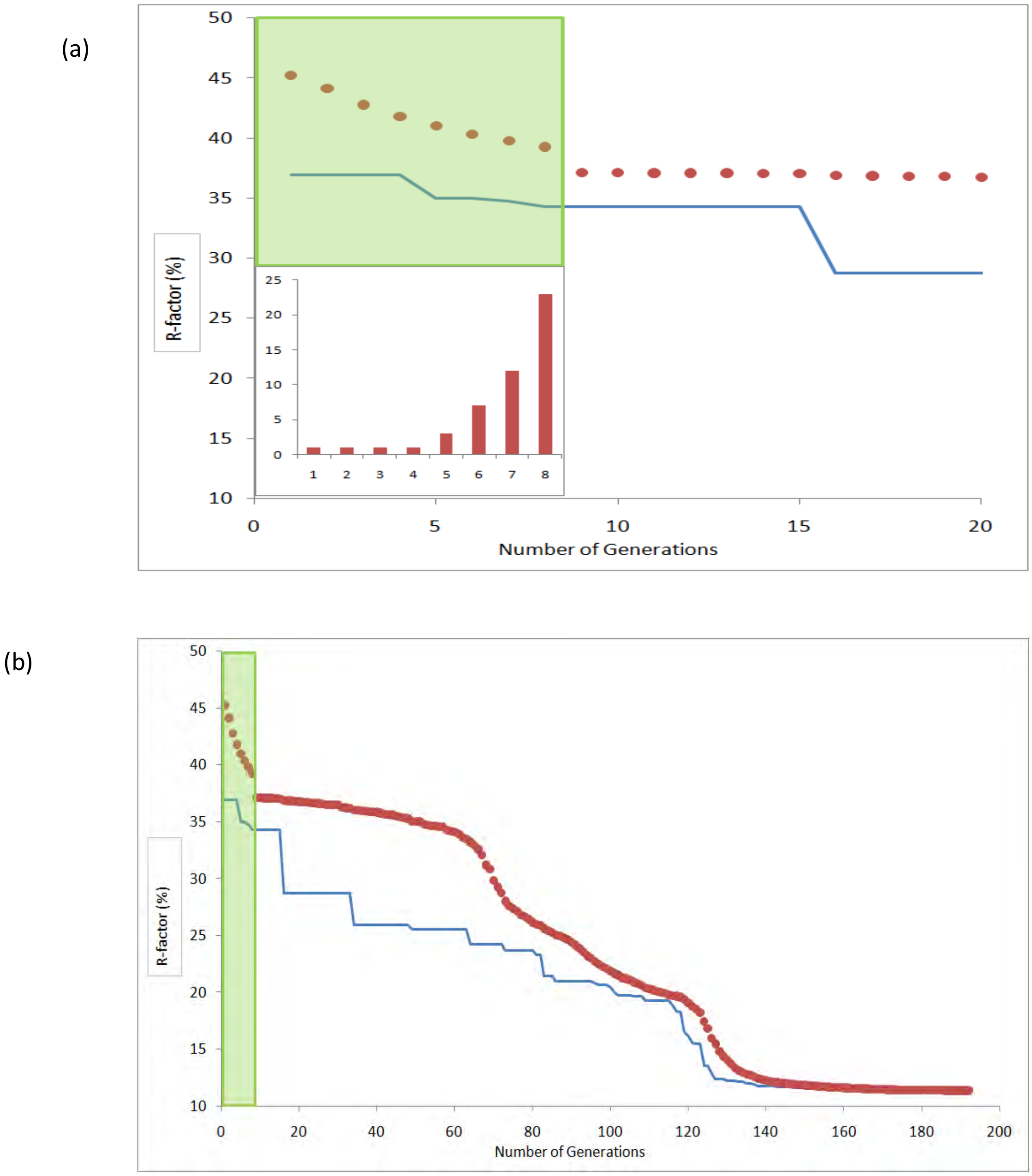


Figure 4.3. Progress of the eugenic DE search used to solve the structure of adipamide for (a) the first 20 generations and for (b) the complete evolutionary process. The red circles denote the mean R_{wp} of the population and the blue line the best individual at that generation. The shaded green area highlights when the primary population is used and the inset histogram the number of individuals with R_{wp} <= the 'target R-factor'.

4.3.2 The crystal structure solution of baicalein by eugenic DE

The structure solution of baicalein is performed using a model defined by seven parameters and for the search to be considered successful, the final solution is required to have an *R* factor <= 15.6%. The average number of child fitness evaluations calculated by successful searches using the same combination of control parameters is used as an indicator of efficiency. In the traditional DE searches, a certain combination of *NP* and *F* is used for each set of five calculations. Due to the increased efficiency of the eugenic DE, each combination of primary *NP*, secondary *NP*, secondary *F* and RequireFrac is used in a set of ten searches. The total number of child fitness evaluations needed by the traditional DE searches are shown in table 4.1 (table 3.2 from chapter 3 presented again here for ease of comparison) and the number needed by the eugenic searches shown in table 4.2.

Comparison of tables 4.1 and 4.2 demonstrates that the eugenic DE searches solve the crystal structure of baicalein in fewer child fitness evaluations than many of the traditional DE searches. The fastest eugenic search with RequireFrac = 0.5 converges with 100% success, calculating on average, 13538 child fitness evaluations. This can be compared with the fastest traditional DE search with *NP* = 105 and *F* = 0.3 that converges with 100% success in an average of 27615 child fitness evaluations. The results in table 4.2 also showed significant increase in overall performance when compared to the traditional DE in which the secondary *NP* = 70 was used throughout the calculation.

Table 4.2 also demonstrates that as the value of RequireFrac increases, the total number of child fitness evaluations calculated decreases. Thus as more models that are clustered near the global minimum are located and transferred into a secondary population, a search using the secondary population is increasingly biased towards the global minimum and converges more rapidly. However, it is not possible to conclude from this limited set of results what effect the value of RequireFrac has on the success rate of a search.

Number of Fitness Evaluations

F \ *NP*	49	70	105	140	280
0.1					**22400** *17920*
0.2					**50120** *28840*
0.3	**0** *0*	**15680** *12740*	**27615** *22785*	**48440** *35840*	**149520** *124320*
0.4	**11711** *11662*	**29330** *29330*	**61110** *48615*	**115500** *107240*	**323960** *283080*
0.5	**21021** *19747*	**47600** *33530*	**92820** *72135*	**190400** *123060*	**530040** *507080*
0.6	**31556** *19747*	**61530** *49140*	**121485** *104580*	**205940** *205940*	**0** *0*

Success rate

Success rate
0 %
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 4.1: Structure solution of baicalein by traditional DE using different combinations of NP (population size) and F (mutation rate). The success rate over the five calculations for each *NP* and *F* combination is indicated in the colour chart: blue for 100% success and red for 0% success rate. The number in bold is the average number of child fitness evaluations required for convergence over the five runs. The number in italics is the optimum calculation from each set of runs in terms of speed of convergence

Number of Fitness Evaluations

RequireFrac ***F(prim/sec)***	**0.25**	**0.333**	**0.5**
0.1/0.5	**23940** ***17710***	**20183** ***15120***	**13538** ***9310***

Table 4.2, Structure solution of baicalein by eugenic DE. As shown in table 4.1, except that data is presented over ten calculations rather than five. Each column represents a RequireFrac proportion. Calculations were carried out with Primary *NP* = 280 and *F* = 0.1; Secondary *NP* = 70 and *F* = 0.5; *K* = 0.99.

4.3.3 The crystal structure solution of adipamide by eugenic DE

The structure solution of adipamide is performed using a model defined by eight parameters. As above, each combination of *NP* and *F* is used five times, whereas each eugenic DE search using a certain combination of primary *NP*, secondary *NP*, secondary *F* and RequireFrac is run ten times. The number of child fitness evaluations calculated by the traditional DE searches are shown in table 4.3 (table 3.5 from chapter 3 presented again here for ease of comparison), and those by the eugenic DE searches are shown in table 4.4.

Number of Fitness Evaluations

F \ ***NP***	56	80	160	320
0.1			**0** *0*	**16640** *15680*
0.2			**13760** *13760*	**39680** *28800*
0.3		**0** *0*	**27360** *23360*	**80320** *69120*
0.4		**17920** *14560*	**57280** *40960*	**118720** *114240*
0.5	**12152** *8904*	**29520** *19600*	**82560** *73920*	**223680** *173760*
0.6	**24416** *18424*	**39760** *33040*	**128320** *108160*	**281600** *200960*
0.7	**34496** *30800*	**55120** *46000*	**177120** *123040*	**404800** *328960*
0.8	**42112** *31976*	**89040** *70880*	**211520** *192480*	**493760** *435840*

Success rate
0 %
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 4.3, Structure solution of adipamide by traditional DE (as denoted in table 4.1).

Number of Fitness Evaluations

RequireFrac ***F(prim/sec)***	0.25	0.333	0.5
0.1/0.5	**24133** *15600*	**17400** *16160*	**17760** *11600*

Table 4.4, Structure solution of adipamide by eugenic DE. As shown in table 4.2. Calculations were carried out with Primary *NP* = 320 and *F* = 0.1; Secondary *NP* = 80 and *F* = 0.5; *K* = 0.99.

Although table 4.4 shows that the eugenic searches do not converge with 100% success, the searches require the calculation of fewer child fitness evaluations than many of the traditional DE searches (table 4.3). The fastest eugenic search with RequireFrac = 0.333 converges on average in 17400 child fitness evaluations, whereas the fastest traditional DE search with *NP* = 80 converges at best with an average of 17920 child fitness evaluations. Again, it is clear that as the value of RequireFrac increases, the total number of child fitness evaluations calculated by eugenic searches generally decreases.

4.3.4 The crystal structure solution of acetarsone by eugenic DE

In the case of acetarsone, the structure solution is performed using a model defined by nine parameters. Each traditional DE search was carried out using a certain combination of *NP* and *F* five times. Each eugenic DE search using a certain combination of primary *NP*, secondary *NP*, secondary *F* and RequireFrac is performed 10 times. The number of child fitness evaluations calculated by the traditional DE searches used to solve the crystal structure of acetarsone are shown in table 4.5 (table 3.8 from chapter 3 presented again here for ease of comparison), and the number of child fitness evaluations calculated by the eugenic DE searches are shown in table 4.6.

Number of Fitness Evaluations

NP ***F***	63	90	135	180	360
0.1					**0** *0*
0.2					**0** *0*
0.3	**0** *0*	**0** *0*	**19035** *19035*	**0** *0*	**92880** *76680*
0.4	**0** *0*	**20700** *19530*	**43875** *22950*	**63000** *48780*	**146160** *114840*
0.5	**21987** *18711*	**37800** *33840*	**48780** *32040*	**114300** *52200*	**346680** *200520*
0.6	**26460** *23499*	**51750** *44010*	**114885** *97335*	**129960** *111960*	**494280** *339120*
0.7	**40320** *25137*	**86040** *63180*	**162675** *125820*	**230040** *210420*	**643320** *455040*
0.8	**71190** *55944*	**107280** *83070*	**176445** *143100*	**0** *0*	**613080** *530280*

Success rate

🟥	0 %
🟫	10 %
🟧	20 %
🟧	30 %
🟨	40 %
🟩	50 %
🟩	60 %
🟦	70 %
🟦	80 %
🟦	90 %
🟦	100 %

Table 4.5, Structure solution of acetarsone by traditional DE (as denoted in table 4.1).

Number of Fitness Evaluations

RequireFrac ***F(prim/sec)***	0.25	0.333	0.5
0.1/0.5	**23490** *16020*	**22928** *20430*	**31253** *18810*

Table 4.6, Structure solution of acetarsone by eugenic DE. As shown in table 4.2. Calculations were carried out with Primary *NP* = 360 and *F* = 0.1; Secondary *NP* = 90 and *F* = 0.5; *K* = 0.99.

Table 4.6 shows that again the eugenic searches converge requiring the calculation of fewer child fitness evaluations than many of the traditional DE searches (table 4.5) and faster than most of the *NP* = 90 calculations. The fastest eugenic search is with RequireFrac = 0.333 and converges with 80% success calculating on average 22928 child fitness evaluations. However, these results suggest that it is also possible to disrupt a search if the secondary population is created containing an excessive number of models that are assigned an *R* factor <= the target *R* factor. In this example, unlike baicalein, eugenic searches with RequireFrac = 0.5 converge calculating on average significantly more child fitness evaluations than when RequireFrac = 0.25 or 0.333.

With a small primary *F* = 0.1, models in the primary population have a high probability of rapidly moving into minima. If the landscape contains multiple relatively deep minima, models in the primary population can gain a relatively low *R* factor by moving into more than one deep minimum. If this happens, when the primary population is pruned, models with relatively low *R* factors occupying different minima are transferred into the secondary population and 'compete' to bias a search towards their particular minima, rather than guiding the search towards the global minimum. Our pruning criteria based on target-*R*-factor cannot distinguish between this case and that in which one deep minimum has been found. The search will not converge until the models in the wrong minima relocate to the global minimum, and this may take many more generations and child fitness evaluations. One possible explanation for the trend observed in the results presented in table 4.6 is that the R_{wp} landscape representing the crystal structure solution of acetarsone has multiple relatively deep minima into which models in a primary population can move. Increasing the value of RequireFrac then increases the probability that many models occupying these deep minima are transferred into a secondary population and compete during the secondary search. In contrast, if the R_{wp} landscapes representing the crystal structures of adipamide and baicalein only have one relatively deep minimum, models located by the primary population with relatively low *R* factors are more likely to be near this deep minimum. Thus a secondary search is more likely to be biased to explore only the global minimum rather than 'multiple' relatively deep local minima; an increase in the value of RequireFrac for these searches does not increase the number of competing models transferred into a secondary population. However, the only way to map the R_{wp} landscapes representing the crystal structures of baicalein, adipamide and acetarsone and determine the nature of the local and global minimum is to run a grid search for each structure. Section 1.4.2 discusses why grid searches are very computationally demanding and because of these computational limitations it has not been practical to perform these grid searches.

4.4 Optimisation of eugenic DE

The value of primary and secondary *NP*, primary and secondary *F* and RequireFrac all affect the total number of child fitness evaluations required for convergence of a eugenic search. In this section, different combinations of primary *NP*, secondary *NP*, secondary *F* and RequireFrac are evaluated in order to determine the optimal combination; defined as the combination that causes the eugenic search to converge with a high success rate whilst calculating the least number of child fitness evaluations. These combinations are tested on eugenic searches used to solve the crystal structures of baicalein, adipamide and the 1:1 salt formed between isonicotinamide and oxamic acid. This salt structure, defined by an R_{wp} landscape with a total of 14 dimensions, was chosen as a significantly more complex example than other structure solution calculations attempted in this and the previous chapter. The eugenic DE was run 10 times for each different combination of control parameters investigated. In the cases of baicalein and adipamide, the structural models and criteria for successful solution are as presented previously.

4.4.1 The crystal structure solution of baicalein

Table 4.7 demonstrates that in general, increasing the secondary mutation rate increases the probability that a search converges successfully; suggesting that although most searches using a secondary population are biased towards the global minimum, it is still possible for a search with a low mutation rate to converge prematurely in a local minimum.

Increasing the number of models in a primary population whilst maintaining both the number of models in a secondary population and the secondary mutation rate, does not significantly increase the probability of successful convergence. Runs in which the secondary population is constant have the same number of models with *R* factors <= the target *R* factor transferred into the secondary population, and hence the search is equally biased towards the global minimum regardless of the number of models in the primary population. Increasing the number of models in the primary population does increase the number of child fitness evaluations calculated by searches, but not as significantly as seen in traditional DE. The eugenic searches with secondary *NP* = 70, RequireFrac = 0.25, secondary *F* = 0.5 and primary *NP* = 280, 560 and 1120 calculate on average 15062,18002 and 26273 child fitness evaluations respectively. Given that the primary population containing 1120 models is four times that containing 280, the searches with primary *NP* = 1120 on average only calculate 74% more child fitness evaluations than searches with

Number of Fitness Evaluations

NP (prim/sec)	280/70		560/70		1120/70	
RequireFrac ***F(prim/sec)***	0.25	0.5	0.25	0.5	0.25	0.5
0.1/0.3	**11284** *7070*	**7245** *7070*	**11634** *11060*	**0** *0*	**20720** *13650*	**25788** *15680*
0.1/0.4	**9888** *8610*	**14728** *12250*	**16463** *14140*	**16758** *11620*	**22840** *15400*	**24127** *15120*
0.1/0.5	**15062** *8400*	**11818** *7560*	**18002** *14210*	**19981** *15610*	**26273** *16380*	**24936** *19320*
0.1/0.6	**15523** *11200*	**14709** *11340*	**19341** *14980*	**19600** *14280*	**25401** *16520*	**28187** *18200*
0.1/0.7	**21656** *15470*	**21686** *14770*	**25953** *16100*	**22435** *16030*	**31204** *21000*	**38010** *29120*
0.1/0.8	**30716** *19460*	**27020** *18970*	**33364** *26670*	**32760** *25060*	**38057** *30380*	**36540** *25620*
NP	280/140		560/140		1120/140	
0.1/0.3	**17472** *15540*	**17808** *9660*	**23485** *19740*	**22260** *16380*	**37730** *28280*	**33530** *28140*
0.1/0.4	**30100** *10780*	**24588** *13720*	**32452** *19880*	**29162** *14980*	**40623** *32200*	**39158** *27020*
0.1/0.5	**33896** *19880*	**35126** *17500*	**44030** *21980*	**35042** *17780*	**45842** *35140*	**38142** *25760*
0.1/0.6	**57295** *24500*	**38556** *25480*	**43358** *27020*	**37816** *28700*	**52262** *41440*	**52780** *39900*
0.1/0.7	**74130** *35280*	**52640** *30940*	**63000** *43120*	**47418** *35700*	**67452** *47460*	**63224** *45220*
0.1/0.8	**91910** *65240*	**57190** *44520*	**63683** *50820*	**66885** *52360*	**83188** *55440*	**79114** *66080*

Success rate

0 %
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 4.7 Structure solution of baicalein by eugenic DE using different combinations of secondary *F*, primary and secondary *NP* and RequireFrac. The top half of the table denotes calculations with a secondary *NP* = 70; the bottom half denotes calculations with a secondary *NP* = 140. The success rate over the ten calculations for each combination is indicated in the colour chart: blue for 100% success and red for 0% success rate. The number in bold is the average number of child fitness evaluations required for convergence over the ten runs, the number in italics denotes the quickest within each set of runs. Calculations were carried out with *K* = 0.99.

primary *NP* = 280. Increasing the size of the primary population increases the probability that initial models are near the global minimum. As the size of a primary population increases, the number of models with *R* factors <= the target *R* factor required to initiate population pruning are located in fewer generations. The number of models in the secondary population has a greater effect on the total number of child fitness evaluations calculated by a search. Searches with primary *NP* = 280, RequireFrac = 0.25, secondary *F* = 0.5 and with secondary *NP* = 70 and 140 calculate on average 15062 and 33896 child fitness evaluations respectively. Thus the searches with the secondary population containing twice as many models calculate at least twice as many child fitness evaluations. Hence an increase in the size of the secondary population significantly reduces the efficiency of the eugenic searches.

Variation of the number of models transferred into the secondary population, RequireFrac, (assigned *R* factors <= the target *R* factor) does not have a completely predictable effect on either the total number of child fitness evaluations required or the success rate of the search. In some cases RequireFrac = 0.25 and 0.5 converge with the same success rate with the larger RequireFrac value requiring fewer child fitness evaluations. For example, (a) primary *NP* = 280, secondary *NP* = 70, secondary *F* = 0.5, RequireFrac = 0.25 and 0.5 converge with 60% success calculating on average 15062 and 11818 child fitness evaluations respectively, (b) primary *NP* = 280, secondary *NP* = 70, secondary *F* = 0.6, RequireFrac = 0.25 and 0.5 converge with 80% success calculating on average 15523 and 14709 child fitness evaluations respectively. Whereas there are other cases in which an increase in RequireFrac results in more child fitness evaluations. The effect on success rate can also be unpredictable. For example, (a) primary *NP* = 560, secondary *NP* = 70, secondary *F* = 0.5, RequireFrac = 0.25 and 0.5 converge with 60 and 90% success respectively, calculating on average 18002 and 19981 child fitness evaluations, (b) primary *NP* = 560, secondary *NP* = 70, secondary *F* = 0.6, RequireFrac = 0.25 and 0.5 converge with 100% and 90% success respectively, calculating on average 19341 and 19600 child fitness evaluations.

Increasing the value of RequireFrac should increase the bias of the secondary search reducing the number of child fitness evaluations calculated by this part of the search. However, increasing the value of RequireFrac increases the number of relatively fit models that need to be located by the primary population which may then need to evolve for more generations. As the primary population contains significantly more models than the secondary population, evolution of the primary population for an increased number of generations potentially requires the calculation of an overall greater number of child fitness evaluations than if evolution of the secondary population required an increased number of generations. When assigning the value of RequireFrac, a compromise needs to be made between the number of child fitness evaluations calculated by the primary and secondary populations. From the results presented in table 4.7 alone, it is not possible to conclude whether it is more efficient to assign RequireFrac a large or small value to optimise the eugenic evolution.

4.4.2 The crystal structure solution of adipamide

Number of Fitness Evaluations

NP (prim/sec)	320/80		640/80		1280/80	
RequireFrac ***F(prim/sec)***	0.25	0.5	0.25	0.5	0.25	0.5
0.1/0.4			**17760** *15440*	**13280** *9040*	**17344** *13840*	**14080** *14080*
0.1/0.5			**18180** *12960*	**19714** *13840*	**24891** *18480*	**24507** *22400*
0.1/0.6	**23634** *13360*	**20446** *15120*	**22302** *18640*	**20516** *16320*	**28950** *21200*	**28229** *20160*
0.1/0.7	**28790** *16000*	**25646** *18240*	**28127** *20720*	**27460** *20880*	**37076** *23360*	**31392** *27600*
0.1/0.8	**43590** *26880*	**45938** *26560*	**42570** *28000*	**38860** *27520*	**47067** *34640*	**40880** *27440*
NP	320/160		640/160			
0.1/0.3	**19840** *14720*	**17200** *16480*	**18944** *14720*	**23488** *20960*		
0.1/0.4	**32320** *17600*	**21440** *18400*	**24580** *21120*	**31223** *22880*		
0.1/0.5	**38960** *20000*	**27460** *17600*	**35063** *24960*	**33404** *23200*		
0.1/0.6	**51893** *30240*	**44480** *25600*	**46160** *31840*	**36060** *25900*		
0.1/0.7	**55253** *30880*	**52366** *40320*	**69831** *40640*	**62200** *36960*		
0.1/0.8	**111238** *68000*	**76278** *43520*	**105248** *53920*	**81720** *52360*		

Success rate

🟥	0 %
🟫	10 %
🟧	20 %
🟧	30 %
🟨	40 %
🟨	50 %
🟩	60 %
🟩	70 %
🟦	80 %
🟦	90 %
🟦	100 %

Table 4.8, Structure solution of adipamide by eugenic DE using different combinations of secondary *F*, primary and secondary *NP* and RequireFrac. The results are presented as in table 4.7.

Table 4.8 demonstrates that once the secondary *F* >= 0.5, increasing *F* does not significantly increase the success rate of a search. This suggests that many secondary searches are sufficiently biased towards the global minimum and increasing the secondary mutation rate does not assist a search to locate the global minimum. This table also demonstrates that increasing the secondary mutation rate increases the total number of child fitness evaluations calculated by a search and hence there is no advantage of using eugenic searches with large secondary *F*s.

Increasing the number of models in the primary population does not significantly increase the total number of child fitness evaluations calculated by a search or the success rate of the search. For example, secondary *NP* = 80, secondary *F* = 0.6, RequireFrac = 0.25 and primary *NP* = 320, 640 and 1280 calculate on average 23634, 22302 and 28950 child fitness evaluations with 70%, 90% and 80% success respectively. However, increasing the number of models in the secondary population significantly increases the number of child fitness evaluations required. Searches with primary *NP* = 320, secondary *F* = 0.6, RequireFrac = 0.25 and secondary *NP* = 160 calculate on average more than twice as many child fitness evaluations than the equivalent search with secondary *NP* = 80. Increasing the number of models in the secondary population does not predictably increase the success rate, hence there is no clear advantage gained by increasing the size of the secondary population.

Increasing the value of RequireFrac generally decreases the total number of child fitness evaluations calculated by a search, demonstrating that increasing the bias of secondary searches causes searches to converge more rapidly. However, considering the results presented in table 4.8, it is not possible to conclude whether increasing the value of RequireFrac increases the success rate of a search.

4.4.3 The crystal structure solution of the isonicotinamide : oxamate 1:1 salt.

The structure of the 1:1 salt formed between isonicotinamide and oxamic acid has been previously solved by the direct space method using traditional DE, [1,2] thus it is possible to test whether eugenic DE solves the structure with greater efficiency. The direct space structure solution of isonicotinamide : oxamate 1:1 requires no constraints on the orientation, position or flexibility of the two models within the unit cell as the space group symmetry ($P2_1/n$) requires the two molecules in a general position. The original structure solution was performed considering this molecular adduct as a cocrystal so the two independent units will be the neutral isonicotinamide and oxamic acid molecules. The two independent molecules are each defined by seven structure parameters; three parameters to define the position (*x,y,z*: between 0 and 1), three orientation (φ,ψ,θ: between 0 and 360) of the molecule and one torsion parameter (τ1: between 0 and 360) as shown in figure 4.4; thus the overall structure is defined by 14 parameters.

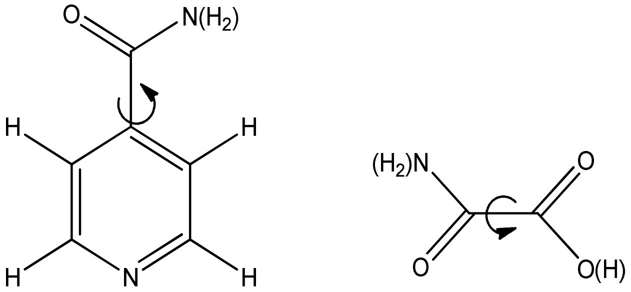


Figure 4.4 The structural model of the isonicotinamide and oxamic acid adduct used in the structure solution calculations. Arrows indicate torsional flexibility. Hydrogen atoms were excluded from the structure solution.

In the case of isonicotinamide : oxamate, a successful structure solution is required to have an *R* factor <= 18.0% at which all models were judged to be close enough for successful Rietveld refinement. For each successful search, the mean number of fitness evaluations calculated by successful searches was calculated. The eugenic DE structure solution calculation was run 10 times for each combination of primary and secondary *NP*, secondary *F* and RequireFrac. The results of the eugenic searches are given in Table 4.9. For comparison, additional calculations were carried out using the traditional DE using different combinations of *NP* and *F*. Due to the extra time needed for traditional DE structure solution calculations, five searches were performed for each combination of *NP* and F: *NP* = 140 and 280 with *F* = 0.1-0.6 and *NP* = 560 with *F* = 0.1-0.4. The results of structure solution by traditional DE are presented in table 4.10.

Number of Fitness Evaluations

NP (prim/sec)	560/140		1120/140	
RequireFrac	0.25	0.5	0.25	0.5
F(prim/sec)				
0.1/0.5	**53340** *39340*	**48944** *41860*	**58450** *48300*	**55650** *32900*
0.1/0.6	**81676** *60620*	**65357** *56140*	**77035** *51660*	**72730** *55440*
0.1/0.7	**120360** *83860*	**128996** *119140*	**151200** *138880*	**110343** *98980*
0.1/0.8	**184007** *155680*	**204480** *136500*	**167230** *132160*	**159600** *142100*

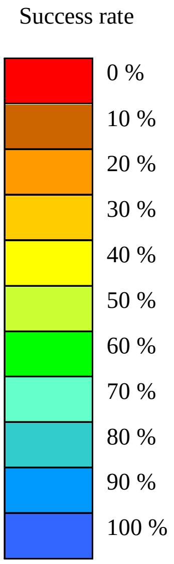


Table 4.9, Structure solution of isonicotinamide : oxamate 1:1 by eugenic DE using different combinations of secondary *F*, primary *NP* and RequireFrac. The results are presented as in table 4.7.

Number of Fitness Evaluations

NP	140	280	560
F			
0.1	**0** *0*	**0** *0*	**0** *0*
0.2	**0** *0*	**0** *0*	**0** *0*
0.3	**0** *0*	**0** *0*	**445200** *413840*
0.4	**0** *0*	**306227** *268520*	**871080** *812000*
0.5	**108990** *107520*	**516880** *513800*	
0.6	**192220** *182980*	**704947** *655480*	

Table 4.10, Structure solution of isonicotinamide : oxamate 1:1 by traditional DE. The results are presented as in previous tables.

Table 4.9 shows that increasing the secondary mutation rate increases the total number of child fitness evaluations calculated. However, searches with smaller secondary *F* can also have relatively high success rates, so again there is no definite advantage of assigning a large value of secondary *F*. In some cases, increasing the number of models in the primary population can decrease the total number of child fitness evaluations calculated by a search for example with secondary *F* = 0.8 in which searches with primary *NP* = 560 and 1120 calculate on average

204480 and 159600 child fitness evaluations respectively. This is because an increase in the number of models in the primary population increases the population density in the landscape and the probability that an initial model is generated near the global minimum. The landscape representing this salt structure is defined by twice as many dimensions as that of an example such as baicalein, hence to create a high population density the population must contain relatively more models. This is in contrast with the majority of the results presented in tables 4.7 and 4.8 which demonstrate that a larger primary population increases the number of child fitness evaluations calculated (although there are results in these tables that also show the opposite trend).

In this example, increasing the value of RequireFrac can increase the success rate of a search. Since the R_{wp} landscape representing the salt structure is defined by twice as many dimensions than that of an example such as baicalein, primary populations created in the landscape representing the salt structure have a comparatively low population density. Increasing the value of RequireFrac significantly increases the number of models clustered near the global minimum that are transferred into the secondary population, increasing the bias of the secondary search and increasing the probability that the search converges in the global minimum. Higher RequireFrac can also reduce the total number of child fitness evaluations required, suggesting that complex crystal structures represented by landscapes with many dimensions should be solved using eugenic searches that transfer relatively many models that are assigned *R* factors <= the target *R* factor into the secondary population.

Comparison of the number of child fitness evaluations calculated by eugenic searches (table 4.9) with those required by traditional DE searches (table 4.10) shows that many of the eugenic searches are significantly more efficient than many of the traditional DE searches. The fastest eugenic search with primary *NP* = 560, secondary *NP* = 140, secondary *F* = 0.5 and RequireFrac = 0.5 converges on average in 48944 child fitness evaluations, while the fastest traditional DE search with *NP* = 280 and *F* = 0.5 converges on average in 108990 child fitness evaluations. In terms of the optimum run in each case, the fastest eugenic DE calculation converged in 39340 child fitness evaluations (primary *NP* = 560, secondary *F* = 0.5, RequireFrac = 0.25) whereas the traditional DE required 107520 (*NP* = 140 and *F* = 0.5).

4.5 Pruning after an arbitrary number of generations

4.5.1 Justification for delayed pruning

Increasing the number of models transferred into the secondary population that are assigned an *R* factor <= the target *R* factor can increase the bias of the secondary search towards the global minimum and reduce the total number of child fitness evaluations required by a search. However, increasing the number of relatively fit models that need to be located by the primary population increases the probability that the primary population evolves for more generations and calculates even more child fitness evaluations. Since a significant number of child fitness evaluations are calculated during the evolution of the primary population, delaying pruning and transferring more fit models into a secondary population may reduce the efficiency of a eugenic search. To determine whether late pruning has a significant detrimental effect on the total number of child fitness evaluations calculated, FixedGenPrune searches were developed in which the pruning is initiated after an arbitrary number of generations defined by the user.

4.5.2 FixedGenPrune DE

A large primary population is created and assigned *F* = 0.1, thus models in the primary population rapidly fall into the nearest minimum. These models have a high probability of being assigned low *R* factors relative to the rest of the population and hence this primary population search can rapidly locate models that are likely to be near the global minimum. After a number of generations (defined by the user), the primary population is pruned and the secondary population created. The models in the primary population are placed in order of increasing *R* factor. A proportion of the models with the highest *R* factors are discarded and a proportion of the models with the lowest *R* factors are transferred into the secondary population, thus the secondary search is biased towards the minima located by the primary population. The mutation rate is increased to 0.6 to reduce the probability that a search converges prematurely.

In the work discussed here, two different sizes of primary population are investigated. The value of primary *NP* = 40 and 80 times the number of parameters defining a structural model in the unit cell. Hence for baicalein the primary *NP* = 280 (7*40) and 560 (7*80), for adipamide the primary *NP* = 320 (8*40) and 640 (8*80) and for the isonicotinamide : oxamate salt, the primary

NP = 560 (14*40) and 1120 (14*80). The number of models transferred into the secondary population is calculated at 10 times the number of parameters defining a model. Hence for baicalein, the secondary *NP* = 70, for adipamide, the secondary *NP* = 80 and for isonicotinamide : oxamate, the secondary *NP* = 140. Each set of FixedGenPrune searches were run 10 times for each combination of primary population size and the generation at which pruning is initiated. Success of solution was judged according to the *R*-factor criteria described in earlier sections.

4.5.3 The crystal structure solution of baicalein

	Number of Fitness Evaluations	
NP (prim/sec) ***GenPrune***	280/70	560/70
5	**22626** *10570*	**21858** *14350*
10	**16840** *12040*	**18370** *15260*
15	**14856** *10220*	**18822** *16520*
20	**14376** *10710*	**20510** *17640*
RequireFrac 0.25	**15523** *11200*	**19341** *14980*
RequireFrac 0.5	**14709** *11340*	**19600** *14280*

Success rate
0 %
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 4.11, Structure solution of baicalein by FixedGenPrune DE using different pruning criteria. GenPrune indicates the generation at which the primary population is pruned and is compared with the RequireFrac results. Calculations were performed with secondary *NP* = 70, primary *F* = 0.1, secondary *F* = 0.6, *K* = 0.99. The results are denoted as in previous tables.

The results presented in table 4.11 for FixedGenPrune searches with primary *NP* = 280 show that delaying pruning decreases the total number of child fitness evaluations calculated by a search, suggesting that delaying pruning increases the number of models clustered near the global minimum that are transferred into the secondary population, increasing the bias of a search towards the global minimum. However, for searches with primary *NP* = 560 delaying pruning decreases and then increases the number of child fitness evaluations calculated. This suggests that if the primary population is sufficiently large, delaying pruning causes a significant number of fitness evaluations to be calculated during evolution of the primary population without simultaneously increasing the bias of the secondary search towards the global minimum. Thus as the number of fitness evaluations calculated during evolution of the primary population increases, the number of fitness evaluations calculated during evolution of the secondary population does not decrease in proportion. These results suggest that as the number of models in the primary population increases, the advantage gained by delaying pruning decreases. There is no predictable effect of delayed pruning on the success rate of the search.

Comparing the number of child fitness evaluations calculated by FixedGenPrune searches and eugenic searches using RequireFrac as the pruning criteria using a secondary *NP* = 70, demonstrates that overall there is no significant improvement with the FixedGenPrune searches within a particular primary population size.

4.5.4 The crystal structure solution of adipamide

Number of Fitness Evaluations

NP (prim/sec)	320/80	640/80
GenPrune		
5	**18930** *12240*	**24423** *18400*
10	**16144** *11280*	**20409** *15680*
15	**15472** *10800*	**21110** *18320*
20	**16350** *13600*	**23467** *19440*
RequireFrac 0.25	**23634** *13360*	**22302** *18640*
RequireFrac 0.5	**20446** *15120*	**20516** *16320*

Success rate

Success rate
0 %
10 %
20 %
30 %
40 %
50 %
60 %
70 %
80 %
90 %
100 %

Table 4.12, Structure solution of adipamide by FixedGenPrune DE using different pruning criteria. GenPrune indicates the generation at which the primary population is pruned and is compared with the RequireFrac results. Calculations were performed with secondary *NP* = 80, primary *F* = 0.1, secondary *F* = 0.6, *K* = 0.99. The results are denoted as in previous tables.

Table 4.12 demonstrates that as pruning is delayed from 5 to 20 generations, the number of child fitness evaluations calculated by a search decreases before it increases again as FixedGenPrune gets larger. This suggests that evolution of the primary population for more generations requires the calculation of more child fitness evaluations without increasing the bias of a secondary search. Again, there is no clear effect on the success rate of the searches.

Comparing the number of child fitness evaluations calculated by FixedGenPrune and eugenic RequireFrac searches, shows that FixedGenPrune searches with *NP* = 320 solve the structure of adipamide with consistently higher efficiency (the fastest on average in 15472 child fitness evaluations for GenPrune = 15), whereas for *NP* = 640, the efficiency of the two methods is similar.

4.5.5 The crystal structure solution of the isonicotinamide : oxamate 1:1 salt

	Number of Fitness Evaluations	
NP (prim/sec)	560/140	1120/140
GenPrune		
5	**57610** *56700*	**68635** *64960*
10	**96133** *48440*	**71645** *65520*
15	**62944** *57540*	**69020** *69020*
20	**51660** *51660*	**76240** *63560*
RequireFrac 0.25	**81676** *60620*	**77035** *51660*
RequireFrac 0.5	**65357** *56140*	**72730** *55440*

Success rate	
🟥	0 %
🟫	10 %
🟧	20 %
🟧	30 %
🟨	40 %
🟩	50 %
🟩	60 %
🟦	70 %
🟦	80 %
🟦	90 %
🟦	100 %

Table 4.13, Structure solution of isonicotinamide : oxamate by FixedGenPrune DE using different pruning criteria. GenPrune indicates the generation at which the primary population is pruned and is compared with the RequireFrac results. Calculations were performed with secondary *NP* = 140, primary *F* = 0.1, secondary *F* = 0.6, *K* = 0.99. The results are denoted as in previous tables.

Table 4.13 demonstrates that delaying pruning has no predictable effect on the number of child fitness evaluations calculated by searches with primary *NP* = 560 especially in the case of FixedGenPrune = 10 where this value seems exceptionally high. For searches with primary *NP* = 1120, delaying pruning does not significantly change the number of fitness evaluations required. Delaying pruning for searches with primary *NP* = 1120 may increase the success rate but results are not conclusive.

These results show that increasing the size of the primary population generally increases the number of child fitness evaluations calculated by a search. If increasing the size of the primary population does not increase the bias of the secondary search it suggests that the primary population does not locate a sufficient number of models clustered near the global minimum that can be transferred into the secondary population. As the size of the primary population increases, the number of child fitness evaluations calculated during evolution of this primary population increases, but the number of fitness evaluations calculated during the evolution of the secondary population does not decrease simultaneously. Using significantly larger primary populations or delaying pruning for longer, is likely to result in primary populations locating more models near the global minimum that can be transferred to the secondary population and successfully bias the secondary search. However, using a larger primary population or further delaying pruning is likely to increase the overall number of fitness evaluations calculated during evolution of an even larger primary population. Since models located in minima are assigned lower *R* factors relative to other models, these models can be transferred into the secondary population as well as models near the global minimum. Although delaying pruning increases the probability that more models near the global minimum are transferred, delaying pruning also increases the probability that a greater proportion of models in the population evolve and locate local minima, thus delaying pruning may increase the probability that more models located in local minima are transferred. As the proportion of models in the secondary population that are initially located in local minima increases, the probability that a search converges prematurely also increases and the probability that a search converges within a convenient number of generations decreases; thus delaying pruning further may not be advantageous. The results presented in table 4.13 suggest that FixedGenPrune searches may be unsuitable for use in the solution of relatively complex crystal structures.

4.6 Convergence rate accelerator

4.6.1 Theory

The results so far in this chapter have shown that the initial implementation of the eugenic DE is likely to be more robust, solving a greater variety of structures with higher rates of success even when governed by a non-optimal combination of control parameters. This is based on the initiation of pruning once a certain proportion of models have acquired a low *R* factor relative to the primary population rather than after an arbitrary number of generations. To increase the probability that a search is successful it is clear that pruning should ideally be initiated by criteria that indicate the clustering of models near the global minimum rather than after an arbitrary number of generations.

An indicator of the clustering of models near the global minimum is to measure the convergence rate of a search. As discussed in section 4.1.3 and demonstrated in figures 3.2, 3.3, 4.1 and 4.2, as the number of models in a population near the global minimum increases, the convergence rate accelerates. The convergence rate can be measured by monitoring the rate of change in the mean *R* factor assigned to a population. The figures demonstrate that the mean *R* factor initially decreases relatively rapidly as parents with significantly low fitness are frequently replaced by fitter children. After a small number of generations, the rate of change of the mean *R* factor decreases as models in the population explore the landscape and parents are replaced by fitter children less frequently. The rate of change in the value of the mean *R* factor does not increase again until the terminal stage of the search when a significant number of models are near the global minimum and children are rapidly replacing the parents as there is rapid final convergence. Thus the value of the mean *R* factor assigned to models in the primary population will be investigated as an indicator of the clustering of models near the global minimum.

4.6.2 The Accelerator DE

The Accelerator DE first initiates a large primary population to increase the probability that an initial model is generated near the global minimum; the primary population is assigned $F = 0.1$ to increase the convergence rate. After the first and tenth generations, the value of the mean *R* factor is recorded and the 'baseline convergence rate' (defined as the average rate of change in the mean *R* factor over the first 10 generations) calculated using equation 4.1. The ‘threshold convergence rate’ is then calculated using equation 4.2. The value of the ‘threshold multiplier’ is defined by the user before the structure solution calculation is initiated.

$$\textit{Baseline convergence rate=(([MeanR generation1]–[MeanR generation10])/10)} \tag{4.1}$$

$$\textit{Threshold convergence rate=([baseline convergence rate]*[threshold multiplier])} \tag{4.2}$$

Once a search has evolved the primary population for 11 generations, the mean *R* factor is recorded after each generation and a ‘generation specific convergence rate’ is calculated using equation 4.3.

$$\textit{Generation specific convergence rate=([MeanR generationX-1]-[MeanR generationX])} \tag{4.3}$$

If the value of the ‘generation specific convergence rate’ is greater than the ‘threshold convergence rate’ the value of a ‘threshold rate exceed counter’ is increased by one. Once a significant proportion of the models in the primary population cluster near the global minimum, the convergence rate accelerates. The value of the ‘generation specific convergence rate’ frequently exceeds the ‘threshold convergence rate’ and the value of the ‘threshold rate exceed counter’ increases. When the value of the ‘threshold rate exceed counter’ exceeds the value of the control parameter ‘Exceed’ (defined by the user), pruning is initiated. The *R* factors of the models in the primary population are placed in order of increasing value and a proportion of the models with the lowest *R* factors are transferred into the secondary population. The models that remain in the primary population are then discarded and the mutation rate is increased to decrease the probability that the secondary search converges prematurely.

The values of the ‘threshold multiplier’ and the parameter ‘Exceed’ control how rapidly the primary population is pruned. Increasing the value of the ‘threshold multiplier’ increases the size of the ‘threshold convergence rate’ relative to the ‘baseline convergence rate’ and decreases the probability that the value of the ‘generation specific convergence rate’ exceeds the ‘threshold convergence rate’. Thus increasing the value of the ‘threshold multiplier’ decreases the probability that pruning is initiated. Increasing the value of ‘Exceed’ increases the number of generations required to have a ‘generation specific convergence rate’ greater than the ‘threshold convergence rate’ before pruning is initiated.

In the work discussed here, two different sizes of primary population are investigated. The value of primary *NP* is calculated at 40 and 80 times the number of parameters required to define a model in the unit cell. Hence for baicalein, the primary *NP* = 280 and 560, for adipamide the primary *NP* = 320 and 640 and for the isonicotinamide : oxamate salt, the primary *NP* = 560 and 1120. The number of models transferred into the secondary population is calculated at 10 times the number of parameters required to define a model: hence for baicalein, the secondary *NP* = 70, for adipamide the secondary *NP* = 80 and for the isonicotinamide : oxamate salt the secondary *NP* = 140. Two values of ‘Exceed’, 2 and 4 are evaluated. Each structure solution calculation was run 10 times using different combinations of primary *NP*, secondary *F* and ‘Exceed’. The value of the ‘threshold multiplier’ was fixed at 1.5 for all calculations.

4.6.3 The crystal structure solution of baicalein

Number of Fitness Evaluations

NP (prim/sec)	280/70		560/70	
Exceed ***F(prim/sec)***	2	4	2	4
0.1/0.6	**16740** *14700*	**15400** *9870*	**31255** *22330*	**29225** *22400*
0.1/0.8	**24458** *20510*	**23144** *18900*	**38430** *30030*	**39235** *28210*
RequireFrac	0.25	0.5	0.25	0.5
0.1/0.6	**15523** *11200*	**14709** *11340*	**19341** *14980*	**19600** *14280*
0.1/0.7	**21656** *15470*	**21686** *14770*	**25953** *16100*	**22435** *16030*
0.1/0.8	**30716** *19460*	**27020** *18970*	**33364** *26670*	**32760** *25060*

Success rate	
🟥	0 %
🟫	10 %
🟧	20 %
🟨	30 %
🟨	40 %
🟩	50 %
🟩	60 %
🟩	70 %
🟦	80 %
🟦	90 %
🟦	100 %

Table 4.14, Structure solution of baicalein by accelerator DE using different values of Exceed. Calculations were performed with secondary *NP* = 70, primary *F* = 0.1, threshold multiplier = 1.5, *K* = 0.99, with different values of secondary *F* and Exceed. For comparison, results from earlier calculations using RequireFrac (table 4.7) are included. The results are denoted as in previous tables.

Table 4.14 demonstrates that increasing the secondary mutation rate increases the success rate of the search. Thus although secondary searches are biased towards the global minimum, searches are still vulnerable to premature convergence and increasing the secondary mutation rate still helps to prevent this. However, increasing the secondary mutation rate increases the number of child fitness evaluations required. Thus when assigning a value to secondary *F*, a compromise needs to be made between speed of convergence and success rate.

Increasing the value of ‘Exceed’ and hence the required number of generations in which the convergence rate is greater than the threshold convergence rate before initiating pruning, generally reduces the number of child fitness evaluations required by a search. This demonstrates that delaying pruning increases the number of models clustered near the global minimum that are transferred into the secondary population. However, increasing the value of Exceed does not necessarily increase the probability that a search converges successfully.

An increase in the size of the primary population also increases the success rate of the searches: in table 4.14, all searches with primary *NP* = 560 converge with 100% success. This suggests that increasing the number of models in a primary population significantly increases the probability that models clustered near the global minimum are transferred into the secondary population.

Overall the accelerator DE and the eugenic DE are similar in terms of efficiency at solving the structure of baicalein. Table 4.14 shows that the fastest accelerator DE search to converge with 100% success was with primary *NP* = 280, Exceed = 2 and secondary *F* = 0.8 and calculates on average 24458 child fitness evaluations. The fastest eugenic search with 100% convergence and primary *NP* = 280, secondary *NP* = 70, secondary *F* = 0.7 and RequireFrac = 0.25 required on average 21686 child fitness evaluations.

4.6.4 The crystal structure solution of adipamide

Table 4.15 demonstrates that increasing the secondary mutation rate again increases the number of child fitness evaluations calculated by a search, although in this case the effect on success rate is not as predictable. If the R_{wp} landscape representing the crystal structure of adipamide contains numerous local minima, there is a relatively high probability that models cluster in local minima as well as the global minimum. In the primary population, models that cluster in local minima are also assigned a relatively low *R* factor and hence when models are transferred into the secondary population there is a high probability that models clustered in local minima as well as the global minimum are transferred. This decreases the probability that a search converges within a convenient number of generations and increases the probability of premature convergence. This is in contrast with traditional differential evolution (table 4.3) where increasing the mutation rate increases the probability that a search converges successfully.

Number of Fitness Evaluations

NP (prim/sec)	320/80		640/80	
Exceed ***F(prim/sec)***	2	4	2	4
0.1/0.6	**16810** *13680*	**11040** *11040*	**25568** *23120*	**33027** *26560*
0.1/0.8	**27093** *21680*	**25440** *25440*	**38260** *24080*	**40520** *33440*
RequireFrac	0.25	0.5	0.25	0.5
0.1/0.6	**23634** *13360*	**20446** *15120*	**22302** *18640*	**20516** *16320*
0.1/0.7	**28790** *16000*	**25646** *18240*	**28127** *20720*	**27460** *20880*
0.1/0.8	**43590** *26880*	**45938** *26560*	**42570** *28000*	**38860** *27520*

Success rate

🟥	0 %
🟫	10 %
🟧	20 %
🟨	30 %
🟨	40 %
🟩	50 %
🟩	60 %
🟩	70 %
🟦	80 %
🟦	90 %
🟦	100 %

Table 4.15, Structure solution of adipamide by accelerator DE using different values of Exceed. Calculations were performed with secondary *NP* = 80, primary *F* = 0.1, threshold multiplier = 1.5, *K* = 0.99, with different values of secondary *F* and Exceed. For comparison, results from earlier calculations using RequireFrac (table 4.8) are included. The results are denoted as in previous tables.

Table 4.15 demonstrates that increasing the size of the primary population increases the number of child fitness evaluations calculated by a search and in general increases the success rate. This suggests that increasing the number of models in a primary population increases the probability that a greater number of models clustered near the global minimum are transferred into the secondary population although this does increase the total number of child fitness evaluations calculated. In this case, delaying pruning by increasing the value of Exceed does not have a predictable effect on the success rate or the total number of child fitness evaluations required. When the searches with primary *NP* = 320, increasing the value of Exceed decreases the number of child fitness evaluations calculated, whereas with primary *NP* = 640, increasing the value of Exceed increases the number of child fitness evaluations.

When the primary population is initiated containing 320 models, the landscape has a lower population density (than primary *NP* = 640) reducing the probability that many models cluster in local minima. Increasing the value of Exceed can increase the number of models near the global minimum transferred into the secondary population. This can increase the bias of a search towards the global minimum and accelerate convergence. However, in this case, increasing the value of Exceed and delaying pruning may increase the probability that a significant number of models clustered in local minima are also transferred. When a search is initiated using primary *NP* = 640, the higher population density increases the probability that models cluster in local minima. Therefore the value of Exceed and the point at which pruning is initiated has less effect on the number of models clustered in local minima transferred and the probability that a search converges prematurely. However, delaying pruning means that many more fitness evaluations are calculated during evolution of the primary population. These results suggest that delaying pruning increases the number of fitness evaluations calculated during evolution of the larger primary population without simultaneously reducing the number of fitness evaluations calculated during evolution of a secondary population.

It is not easy to predict or analyse the effect that a particular combination of control parameters may have on the accelerator DE search applied to the structure solution of adipamide, and it may be that this particular search technique is unsuitable for this particular example.

4.6.5 The crystal structure solution of the isonicotinamide : oxamate 1:1 salt.

Table 4.16 demonstrates that increasing the secondary mutation rate again increases the total number of child fitness evaluations required, but does not predictably increase the success rate of the searches. Increasing the size of the primary population has a similar effect on both success and number of evaluations and hence neither of these combinations of control parameters are advantageous.

Increasing the value of Exceed and delaying pruning, reduces the total number of child fitness

Number of Fitness Evaluations

NP (prim/sec)	560/140		1120/140	
Exceed ***F(prim/sec)***	2	4	2	4
0.1/0.6	**60600** *54600*	**56000** *51240*	**111900** *69160*	**97020** *82880*
0.1/0.8	**138740** *125720*	**133392** *122500*	**199040** *154560*	**176316** *154840*
RequireFrac	0.25	0.5	0.25	0.5
0.1/0.6	**81676** *60620*	**65357** *56140*	**77035** *51660*	**72730** *55440*
0.1/0.7	**120360** *83860*	**128996** *119140*	**151200** *138880*	**110343** *98980*
0.1/0.8	**184007** *155680*	**204480** *136500*	**167230** *132160*	**159600** *142100*

Success rate

■	0 %
■	10 %
■	20 %
■	30 %
■	40 %
■	50 %
■	60 %
■	70 %
■	80 %
■	90 %
■	100 %

Table 4.16, Structure solution of the isonicotinamide:oxamate salt by accelerator DE using different values of Exceed. Calculations were performed with secondary *NP* = 140, primary *F* = 0.1, threshold multiplier = 1.5, *K* = 0.99, with different values of secondary *F* and Exceed. For comparison, results from earlier calculations using RequireFrac (table 4.9) are included. The results are denoted as in previous tables.

evaluations calculated by a search, demonstrating that delaying pruning increases the probability that more models cluster near the global minimum and are transferred into the secondary population, increasing the bias of the secondary search. However, delaying pruning often decreases the success rate of the searches, therefore when deciding when to prune, a compromise needs to be made between speed of convergence and success rate.

Comparing the number of child fitness evaluations and success rate of the accelerator and the eugenic DE searches, the accelerator DE is clearly more efficient. The fastest accelerator search converging with 70% success with primary *NP* = 560, secondary *F* = 0.6 and Exceed = 2 calculates on average 60600 child fitness evaluations, whereas the fastest eugenic search, also with 70% success with primary *NP* = 560, secondary *F* = 0.7 and RequireFrac = 0.25 calculates on average 120360 child fitness evaluations.

4.7 Conclusions

The results obtained by using the three different search techniques discussed in this chapter demonstrate that searches in which the size of a population is decreased as a search progresses are generally more efficient than searches that use a population of constant size. Pruning many of the most unfit models (with the highest *R* factors) from a population once the global minimum has been approximately located, significantly reduces the number of child fitness evaluations required whilst a search converges on the optimal solution.

Increasing the mutation rate once a significant number of models have been removed from a population may increase the probability that the optimal solution is located by a search, but increasing the mutation rate can slow the convergence rate and increase the number of child fitness evaluations calculated.

Although evolution of a large primary population containing many models requires the calculation of a significant number of child fitness evaluations, delaying pruning may reduce the total number of child fitness evaluations by allowing a greater number of models clustered near the global minimum to be located and transferred into the secondary population. As a result, a search is concentrated around the global minimum and rapidly converges on the optimal solution requiring the calculation of fewer child fitness evaluations. However, in some cases delaying pruning may increase the number of child fitness evaluations calculated during evolution of the primary population without simultaneously reducing the number of fitness evaluations calculated during evolution of the secondary population.

Since the eugenic and accelerator searches prune a population when a significant number of models are near the global minimum, these two search techniques are potentially more robust than FixedGenPrune searches that prune a population after an arbitrary number of generations (regardless of how many models are near the global minimum). Thus eugenic and accelerator searches are more likely to successfully solve a greater variety of crystal structures even when a search is governed by a non-optimal combination of control parameters. This increases the utility of eugenic DE as the user is not required to ponder what combination of control parameters might be optimal for a particular calculation.

Additional Note.

Since the work in this thesis has been carried out, the Eugenic DE has been used by others in the Tremayne research group and has successfully solved the structure of a previously unknown cocrystal, 1:1 nicotinamide:succinic acid from powder diffraction data[3]. The structural model was comprised of two individual molecular components requiring definition by a total of 15 structural parameters. The traditional DE solved the structure successfully in 444,300 child fitness evaluations (*NP* = 300, *K* = 0.99, *F* = 0.4, Generations = 1481) whereas the eugenic DE solved the structure in 60,750 child fitness evaluations (primary *NP* = 600, secondary *NP* = 150, *K* = 0.99, primary *F* = 0.1, secondary *F* = 0.5, RequireFrac = 0.25, Generations = 378). A view of the final crystal structure is shown in Figure 4.5.

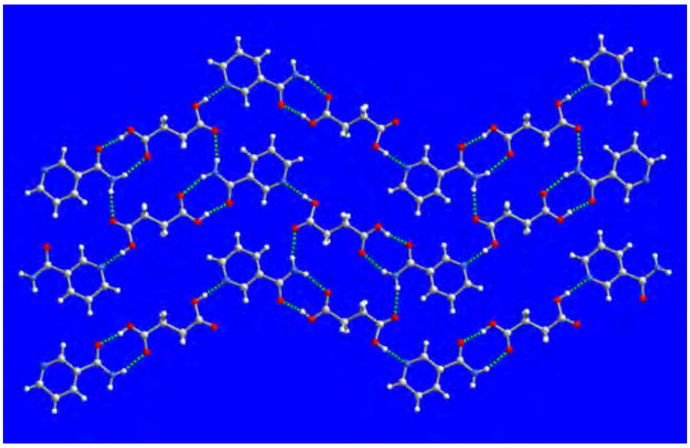


Figure 4.5, A View of the Crystal Structure of the Nicotinamide : Succinic Acid 1:1 Cocrystal

References.

[1] A. Cowell. An Investigation into the Synthesis, Structural Characterisation, Thermal and Polymorphic Behaviour of Organic Crystalline Materials. PhD Thesis. School of Chemistry. University of Birmingham UK. (2011).

[2] A. Cowell, B. M. Kariuki, R. W. Lancaster and M. Tremayne. Oxamate Salts of Isonicotinamide and Nicotinamide: A Powder and Single Crystal Diffraction Study. *Manuscript In Preparation*.

[3] L. J. Thompson, D. Bell, E. J. Shotton and M. Tremayne. Structure Determination of the 1:1 Cocrystal Nicotinamide:Succinic Acid by Powder X-ray Diffraction. *Manuscript In Preparation*.

Chapter 5 Analysing Biphasic Crystalline Materials using X-ray Powder Diffraction.

Section 1.7 of this thesis discusses how Rietveld refinement is used to evaluate multiphasic powder diffraction patterns recorded for mixtures of crystalline phases and determine the abundance of each crystalline phase. However, quantitative phase analysis (QPA) by Rietveld refinement is only possible when each crystal structure is known *a priori*, as this structural information is used to simulate multiphasic powder patterns that are compared with the real multiphasic pattern. Due to the overlap of diffraction peaks that each result from diffraction by different crystal phases, the intensity and position of peaks observed in the real pattern may not correspond to any ‘single’ crystal structure. It may also be difficult to distinguish between overlapped peaks and single peaks that correspond to one phase. Thus it is significantly more difficult to determine a crystal structure from a multiphasic diffraction pattern.

Although most crystal structures are determined from monophasic diffraction patterns, sometimes it is not possible to produce a ‘pure’ monophasic sample of the target material and record a monophasic diffraction pattern. In these cases only a multiphasic pattern can be recorded, hence it becomes necessary to determine the structure of the target material directly from the multiphasic pattern.

5.1 Multiphasic crystalline materials

5.1.1 Simultaneous polymorphism

Although recrystallisation generally removes impurities from the final product it is not unknown for multiple materials to crystallise simultaneously. Section 2.5 of this thesis discusses the simultaneous crystallisation of cocrystals with quantities of unreacted starting material and a solvate. [1] It is also not unknown for multiple polymorphs to crystallise simultaneously. [2-11] Figure 5.1 shows the crystal shape of two polymorphs of the 2:1 4-cyanopyridine : 4,4'-biphenol cocrystal that crystallise simultaneously, whereas Figure 5.2 shows the different packing arrangements of these two polymorphs.

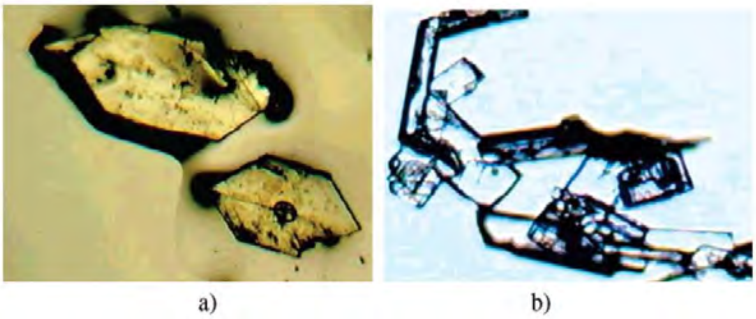


Figure 5.1. Two polymorphs of 2:1 4-cyanopyridine : 4,4'-biphenol cocrystals that crystallise simultaneously from the mother liquor. (a) Form [I] irregular hexagons, (b) form [II] parallelepiped plates. Figure taken from reference 6.

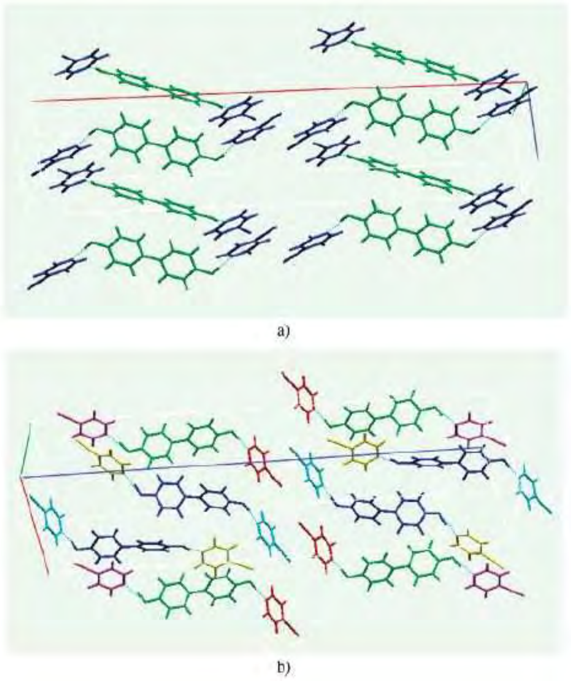


Figure 5.2. Crystal packing of two polymorphs of 2:1 4-cyanopyridine : 4,4'-biphenol cocrystals, (a) form [I] (b) form [II]. Molecules are coloured according to the symmetry equivalence. Figure taken from reference 6.

5.1.2 Determination of crystal structure.

When it is visually obvious that a material is comprised of crystals of different shape or colour it is possible to manually separate the different crystals and determine their respective structures separately. [6,8,9,11] However, it is not always obvious that a material contains multiple crystal phases, for example, two polymorphs of cyclopentadienyl rubidium were simultaneously synthesised as a white powder. [3] When it is not visually obvious that a material is multiphasic, it is not possible to manually separate the phases and determine their respective structures separately. In these circumstances, it is necessary to record a diffraction pattern for the multiphasic material and determine the different crystal structures from the multiphasic pattern. Once the separate crystal structures are known, it may be possible to design synthetic routes to favour production of one form. However, it is not necessarily trivial to determine crystal structure from a multiphasic diffraction pattern. Since the observed peak spacing is not compatible with one unit cell, traditional indexing attempts to index a multiphasic pattern using a single unit cell will fail unless a discrete set of peaks corresponding to diffraction by one crystal phase can be identified.

5.1.2.1 Crystal structure determination of cyclopentadienylrubidium.

In this example, a powder diffraction pattern was recorded for a material containing two polymorphs of cyclopentadienylrubidium, and to the unsuspecting eye it was not obvious that the pattern contained peaks corresponding to different crystal structures. Initially, attempts to index the powder pattern with a single set of lattice parameters failed. However, two sets of peaks with distinct shapes (one much narrower than the other) could be identified (figure 5.3). [3] It was possible to index the different sets of peaks to two different orthorhombic unit cells and use *ab initio* techniques to determine each crystal structure.

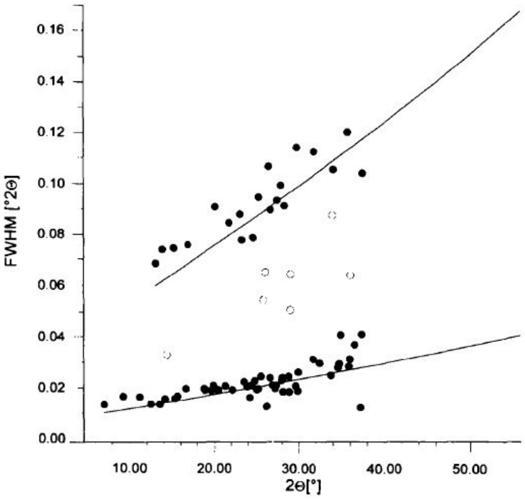


Figure 5.3, A plot showing the FWHM for both phases of cyclopentadienylrubidium extracted from the biphasic powder diffraction pattern. The pattern contains two sets of peaks of distinct shape, one much broader than the other. The open circles mark peaks that cannot be assigned to one phase. Figure taken from reference 3.

5.1.2.2 Crystal structure determination of Sb_3O_4I.

Initially all attempts to index the single crystal diffraction pattern recorded for crystals of Sb_3O_4I failed. [2] The pattern displays perfect non-crystallographic extinction conditions that are often associated with twinning, however no twinning operation could be found to reconcile the observed diffraction data with a single unit cell. It was discovered that the diffraction pattern was compatible with the single crystals being comprised of two intergrown polymorphs. [2] Identification of systematic absences was used to postulate a unit cell and the crystal structure of one polymorph was found to be compatible with this first unit cell. A second unit cell compatible with the structure of the second polymorph was generated by using the first unit cell as a model and altering its symmetry until lattice parameters that were compatible with a sensible structure and the remaining peaks were generated.

5.1.3 Non-separable peaks.

Once the unit cells of the two forms of cyclopentadienylrubidium had been determined it was possible to assign many peaks in the biphasic powder pattern to a particular phase. However, some peaks could not be assigned to either phase. If peaks that each result from diffraction by different crystal structures overlap, the position and intensity of the resultant peak may not be compatible with the unit cell of either structure. Thus for successful indexing of either unit cell overlapped peaks must be identified and excluded.

5.2 Indexing multiphasic diffraction patterns

5.2.1 The pattern subtraction method

A technique that can be used to do this is the pattern subtraction method. [5,12-14] If a multiphasic pattern is recorded for a multiphasic material containing one unidentified crystal phase, a diffraction pattern can be recorded for all the identifiable phases and superimposed on the multiphasic pattern. Peaks resulting from the identified phases can then be subtracted from the multiphasic pattern, generating a pseudo-monophasic pattern. Although some overlapped peaks may remain in the pseudo-monophasic pattern, most of the remaining peaks will result from diffraction by the unidentified phase. A process of elimination during indexing is likely to reveal which are the overlapped peaks, and once these overlapped peaks are excluded, indexing is likely to be successful. It is not necessary to know the crystal structure of the identifiable phases to use the pattern subtraction method, [5] it is only necessary to be able to identify and record a diffraction pattern for all but one phase. Three examples of cocrystals, 1:1 caffeine : acetic acid and forms [I] and [II] of 1:1 caffeine : trifluoroacetic acid have been prepared, [5] by combining the respective components without solvent, using pestle and mortar (a technique known as 'cocrystal controlled solid-state synthesis', [15] or 'mechanochemistry'. [16] Since no [15] or little [16] solvent is used during this technique the product is not automatically purified by recrystallisation. Therefore if the starting materials are not combined in exact stoichiometric quantities or not ground for sufficient time, traces of unreacted starting material may persist amongst the product. [5,16] Examination of the powder patterns recorded for the three cocrystals [5] revealed that traces of unreacted crystalline anhydrous caffeine were present. Although the crystal structure of anhydrous caffeine is not known [5] it was possible to record a powder pattern for a pure sample of crystalline anhydrous caffeine and subtract this pattern from the biphasic pattern recorded for each cocrystal and caffeine impurity.

Diflorasone diacetate, a steroidal anti-inflammatory drug, has been prepared in three anhydrous and one solvated form. [12] The anhydrous forms [I] and [III] and the solvate can all be produced as monophasic materials, and the crystal structures of these three forms have been determined from monophasic powder patterns. However, the anhydrous form [II] is obtained as a mixture of anhydrous forms [I] and [II] by heating the solvated form to 90°C. [12] Instead of physically separating form [I] from form [II], a biphasic powder pattern was recorded for a mixture of forms [I] and [II] and the powder pattern recorded for form [I] was subtracted from the biphasic pattern. The crystal structure of form [II] was successfully determined using the remaining peaks in the pseudo monophasic pattern. [12]

However, the pattern subtraction method can only be used when all but one of the phases in a multi-phasic material can be identified. If a triphasic diffraction pattern is recorded, and only one of the phases can be identified, subtracting a pattern recorded for this one identifiable phase from the triphasic pattern will produce a pseudo-biphasic pattern. Thus in this case the subtraction method does not aid in the assignment of peaks to a particular phase. Also, if a multiphasic pattern is recorded and a significant number of peaks resulting from diffraction by an identifiable phase and an unidentified phase overlap, subtracting a pattern recorded for the identifiable phase destroys much of the information about the unidentified phase.

5.2.2 Anisotropic thermal expansion

A technique that has been used to increase the number of individual peaks that can be resolved in monophasic powder diffraction patterns relies on the fact that many low symmetry molecular crystals exhibit anisotropic thermal expansion. [17-20] When the temperature of a crystal exhibiting anisotropic thermal expansion is increased, the lengths of different crystal axes change at different rates. Since the position of peaks in a diffraction pattern are dependent on the lattice parameters, recording multiple diffraction patterns at different temperatures for a crystalline material exhibiting anisotropic thermal expansion causes different peaks to ‘shift’ different amounts. Thus peaks that overlap in a pattern recorded at one temperature may shift different amounts and not overlap in another pattern recorded at a different temperature. Thermally induced change in the overlap of diffraction peaks is best revealed using synchrotron data. [19] The high resolution intrinsic to synchrotron data means that small changes in peak position can be more accurately measured.

The anisotropic thermal expansion technique for resolving overlapped peaks in monophasic powder diffraction patterns has been applied to multiphasic patterns. [17] The technique is most effective when the different crystal phases display considerably different rates of thermal expansion as this maximises the relative amount of observable peak shift. Figure 5.4 shows biphasic diffraction patterns recorded at different temperatures for one sample of an alloy composed of 98% plutonium and 2% uranium. Many peaks corresponding to Pu and U which are superimposed in a pattern recorded at one temperature can be resolved into individual peaks if a different pattern is recorded at a different temperature. [17]

Since the number of overlapping peaks in a multiphasic diffraction pattern increases as the number of different crystal phases in a multiphasic material increases, more patterns would need to be recorded at different temperatures for materials containing more phases.

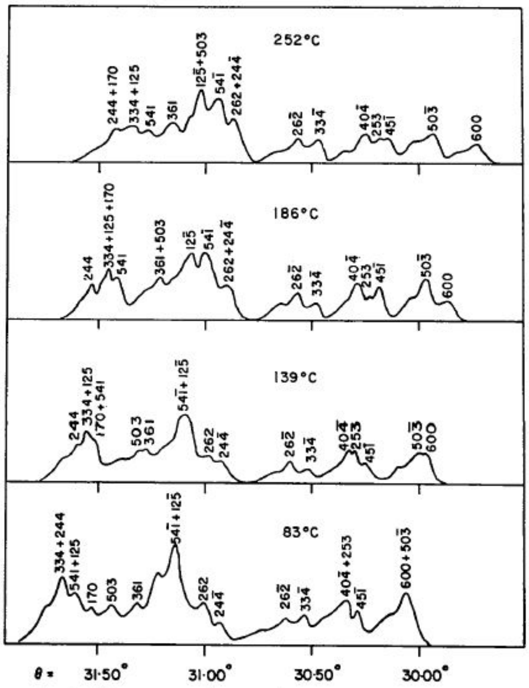


Figure 5.4. Biphasic powder diffraction patterns recorded at different temperatures for one alloy composed of 98% plutonium and 2% uranium. Figure taken from reference 17.

This technique has a significant advantage over the pattern subtraction method. In order to use this technique, it is not necessary to first identify any phases present in a multiphasic material in order to identify a discrete set of peaks that can be assigned to one phase. Thus, if multiple new crystalline materials are synthesised forming a 'uniform' white powder (as in the case of cyclopentadienylrubidium), [3] anisotropic thermal expansion can be used to identify a discrete set of peaks that result from diffraction by one phase, whereas the pattern subtraction method could not be used.

As the temperature range over which multiphasic powder patterns are collected is increased, the probability that a crystal phase change is induced in one phase increases. This is potentially useful. The change to a different crystal structure is likely to cause peaks that result from diffraction by that phase to significantly shift position. However, the position of peaks that result from diffraction by other phases that do not undergo a phase change at that particular temperature will not shift significantly. Thus a crystal phase change can be used to identify and separate peaks into discrete sets that each result from diffraction by different phases.

5.2.3 Indexing multiphasic patterns by global optimisation.

Section 2.1.1.1 of this thesis discusses how the overlap of peaks in a monophasic powder diffraction pattern can prevent the indexing of the pattern. A technique that can index monophasic patterns despite peak overlap treats indexing as a global optimisation problem. [21] In this technique, a genetic algorithm generates a population of model unit cells. The values of the lattice parameters of each unit cell are treated as chromosomes, thus the GA can evolve the population of model unit cells. A powder diffraction pattern is simulated for each model and quantitatively compared with the real pattern that is being indexed using a cost function based on *R* factor. Model unit cells that generate simulated patterns that are a better fit with the real pattern are assigned an *R* factor with a lower value. Evolution of the population improves the quality of the models and a model unit cell that is assigned a significantly low *R* factor can be assumed to be a good representation of the real unit cell and used in the following stages of profile fitting and structure solution. Since a cost function based on *R* factor is used, peak overlap is taken into account and the need to fit peak spacing is negated by matching the whole profile shape.

Since any powder diffraction pattern can be indexed using a unit cell of sufficient volume, the volume of a model unit cell is defined (within limits) by the user before the GA indexing procedure is initiated. This prevents a search from indexing a pattern by optimising an excessively large model. Since the volume of the unit cell is defined by the user the best model will only fit all the peaks that result from diffraction by one crystal phase. Thus when a biphasic powder pattern is recorded the procedure is prevented from indexing both sets of peaks using one model unit cell. Since peaks that result from diffraction by one phase do not fit the model used to index a different phase, when the quality of model unit cells is assessed, peaks resulting from other phases contribute a constant amount to the *R* factor regardless of the quality of the model. Thus the best model unit cell can still be defined as the model that is assigned an *R* factor with the lowest value. This technique has been successfully used to index a biphasic pattern. [21]

5.3 Quantitative Phase Analysis by X-ray Powder Diffraction

5.3.1 Using predetermined structural information

In a multiphasic diffraction pattern, the intensity of a peak that results from diffraction by a particular crystal phase relative to the intensity of peaks that result from diffraction by a different crystal phase is proportional to the relative abundance of each phase. [22-29] This relationship forms the basis of quantitative phase analysis (QPA) by Rietveld refinement. [22-25,27-36] As discussed in section 2.1.3 of this thesis, the Rietveld method [37] involves simulating a diffraction pattern for a computer generated model of a crystal structure and quantitatively comparing the simulated pattern with the real pattern. Refinement of parameters defining the model causes the model to become more realistic and improves the fit between simulated and real patterns. During each cycle of refinement, a scale factor is refined so that simulated diffraction peaks have the same magnitude of intensity as equivalent real peaks. If the scale factor is not refined it is unlikely that a diffraction pattern simulated for a model that is a good representation of the real crystal structure fits the real pattern. This means that a good model is assigned an *R* factor with an inappropriately high value and decreases the probability of a successful refinement. Rietveld refinement can be used to simulate multiphasic diffraction patterns and quantitatively compare them with real multiphasic patterns. Each phase in the simulated pattern is treated separately and assigned its own phase-specific scale factor. During multiphasic Rietveld refinement, each phase-specific scale factor is also refined separately. Hence simulated peaks resulting from diffraction by one phase can be fitted to equivalent peaks in the real multiphasic pattern. Once a good fit between simulated and real multiphasic patterns is achieved, the value of each phase-specific scale factor can be used to calculate the relative abundance of each crystalline phase in the real sample.

The intensity of a peak that results from diffraction by one crystal phase is related to the abundance of that phase by a phase specific calibration constant shown in equation 5.1.

$$w_i = \frac{S_i(ZMV)_i}{\sum_{j=1}^{n} S_j(ZMV)_j} \tag{5.1}$$

where w_i is the weight fraction of phase *i*, *S* is the refined Rietveld scale factor, *ZMV* is the calibration constant, *Z* is the number of formula units per unit cell, *M* is the mass of the formula unit and *V* is the volume of the unit cell [24].

The structure specific calibration constant is calculated using the volume of the unit cell, the number of formula units inside one unit cell and the mass of one formula unit. [24] Hence if the structure of a crystal is known, it is possible to calculate the structure specific calibration constant. If all the structures of all the crystal phases present in a multiphasic material are known and a multiphasic diffraction pattern is recorded for the material, the Rietveld technique can be used to refine each phase-specific scale factor and calculate the relative abundance of each of the phases.

5.3.2 Relative and absolute abundance

The Rietveld technique can only calculate a relative abundance because Rietveld refinement does not account for any amorphous material (such as glassy materials) [32] that may be present in a material. Since amorphous material does not scatter X-rays coherently, Rietveld refinement cannot simulate sharp diffraction peaks for amorphous material. To determine the absolute abundance of a crystalline phase in a multiphasic material containing amorphous material it is necessary to use a calibration standard. [28,32] If the crystal structure of a material is known, it can be used as a calibration standard. A measured mass of the calibration standard is added to a measured mass of the multiphasic material, thus the total abundance of the standard in the material is known. A multiphasic diffraction pattern is then recorded. Since the crystal structure and absolute abundance of the standard is known, the intensity of peaks resulting from diffraction by the standard provides a standard intensity, against which the intensity of peaks resulting from diffraction by the other crystal phases can be measured. Thus it is possible to determine the absolute abundance of each of the crystal phases. If the total abundance of all crystal phases does not equal one, it indicates that amorphous material is present in the material and that the absolute abundance of amorphous material is the difference between the total abundance of all crystal phases and one.

5.3.3 The reliance on previously acquired structural knowledge

Previously the Rietveld refinement technique has only been used for QPA when the crystal structure of each phase in a multiphasic material is known *a priori*. A study [34] was conducted to investigate the accuracy of QPA by Rietveld refinement using test multiphasic materials prepared containing known quantities of: (a) different inorganic crystalline minerals or (b) different molecular crystals. The study [34] demonstrated that

Rietveld refinement could more accurately determine the composition of multiphasic material comprised of mineralogical phases than molecular phases. This is because molecular crystals generally have more complex crystal structures that are more difficult to determine than inorganic minerals. This means that more accurate structural models of the minerals can be generated and so more realistic multiphasic patterns are simulated for multiphasic material comprised of mineral phases.

Section 1.1 of this thesis discusses how different polymorphs of an active pharmaceutical ingredient (API) can have different biological activity. For example, mebendazole has been recrystallised in three anhydrous polymorphic forms. [38] The presence of form B in medication increases the toxicity of the medication, whereas form A has no anthelminthic properties and renders medication useless when form A is present in concentrations greater than 30%, thus form C is the most suitable form for pharmaceutical application. However, traces of the A and B forms have been detected in samples of packaged medication. [38] If sufficiently accurate structural models of the three polymorphs of mebendazole could be generated, QPA by Rietveld refinement could be used as a quality control technique and to determine the abundance of each polymorph in the final product. However, the inability to generate sufficiently accurate structural models of APIs can hinder the pharmaceutical industry in using this method of quality control.

5.4 Direct space structure solution from biphasic powder diffraction data

Despite the peak overlap observed in monophasic powder diffraction patterns, direct space methods can successfully solve crystal structures. The following work discussed in this chapter investigates the application of direct space methods to solving molecular crystal structures from biphasic powder diffraction data. Quantitative phase analysis by Rietveld refinement using the models generated by the direct space method is also attempted

5.5 Sample preparation

Biphasic materials were prepared by manually mixing two crystalline organic amides in known quantities using a pestle and mortar. Since both molecules contain the same functional group a reaction forming a new third phase is prevented. Biphasic materials were prepared by mixing different quantities of: (a) the triclinic form of adipamide (1,6-hexanediamide $C_6H_{12}N_2O_2$) with nicotinamide (3-pyridinecarboxamide $C_6H_6N_2O$), or (b) the triclinic form of adipamide with oxamide (1,2-ethyldiamide $C_2H_4N_2O_2$).

The published crystal structure of the triclinic form of adipamide was determined from powder diffraction data recorded at 273 K. [39] The structure adopts the space group P-1 and has the following lattice parameters: a = 5.1097(2)Å, b = 5.5722(2)Å, c = 7.0473(3)Å, α = 69.575(1)°, β = 87.120(3)° and γ = 75.465(3)°, giving a unit cell volume=181.87(2)Å^3.

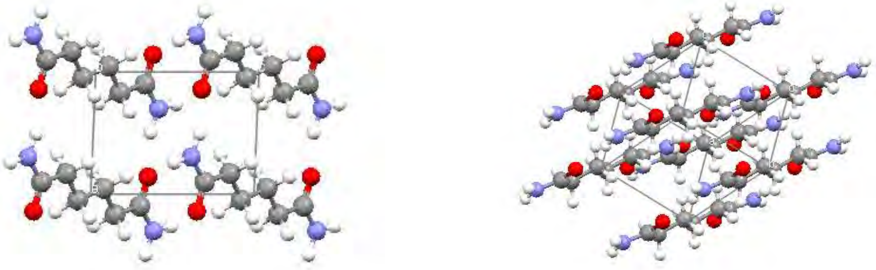


Figure 5.5, Two views of the published crystal structure of the triclinic form of adipamide [39].

The published crystal structure of nicotinamide was determined from single crystal diffraction data recorded at 295 K. [40] The structure adopts the space group $P2_1/c$ and has the lattice parameters: a = 3.975(5)Å, b = 15.632(8)Å, c = 9.422(4)Å and β = 99.03(7)° giving a unit cell volume = 578 Å^3.

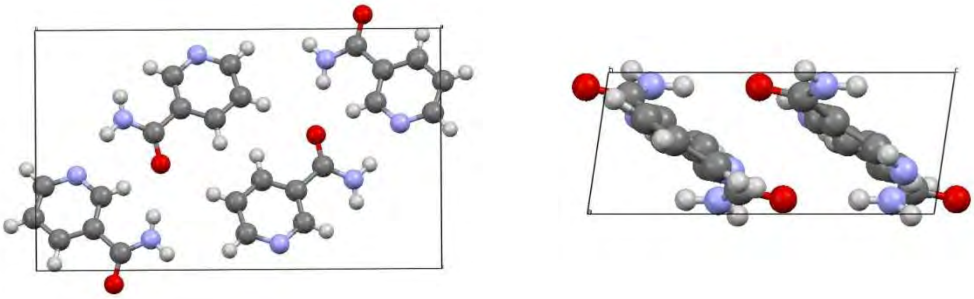


Figure 5.6, Two views of the published crystal structure of nicotinamide [40].

The published crystal structure of oxamide was determined from single crystal diffraction data recorded over the temperature range 283-303 K. [41] The structure adopts the space group P-1 and has the following lattice parameters: a=3.618(1)Å, b=5.180(1)Å, c=5.651(1)Å, α=83.77(1), β=113.97(1) and γ=114.94(1)° giving a unit cell volume=87.497Å^{3}.

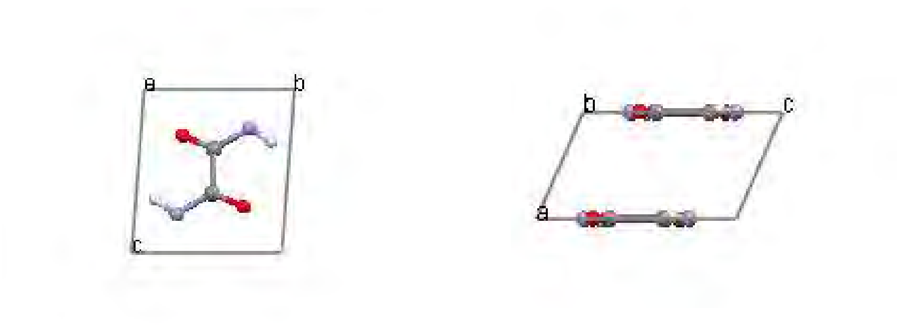


Figure 5.7, Two views of the published triclinic crystal structure of oxamide [41].

Since the crystal structure of these amides is already known, the ability of the direct space method to solve the crystal structures from biphasic powder data can be evaluated. Similarly, since the biphasic materials were prepared by mixing the two amides in known quantities the accuracy of quantitative phase analysis using models generated by our direct space method can be evaluated.

Adipamide and nicotinamide were chosen because their molecular structures are considerably different. Adipamide is a linear molecule with five internal degrees of freedom. However, nicotinamide is based on a rigid pyridine ring substituted with one amide group and only has one internal degree of freedom. This provides a suitable case to test if the direct space method is equally capable of solving both fairly flexible and rigid structures from the same biphasic pattern.

Adipamide and oxamide were chosen because the two molecules have similar structures, (both linear diamides) unlike adipamide and nicotinamide. However, oxamide has only three internal degrees of freedom. This provides a second opportunity to test the capability of direct space structure solution from a biphasic pattern.

5.5.1 Methodology

Since the crystal structures of the amides used in these experiments are known, no attempt was made to index biphasic patterns recorded for the samples containing: (a) adipamide and nicotinamide or (b) adipamide and oxamide. Instead, the position of peaks in the biphasic patterns was inspected and was confirmed to be compatible with the published structures. Appendix A describes in detail the experimental apparatus used to record the biphasic patterns and also shows some of the powder profiles that were recorded. A Le Bail fit was generated for each biphasic pattern using the published lattice parameters. Initially the experimentally estimated relative abundance of each phase was used to specify values of phase abundance variables used by GSAS [43] to fit the pattern. In GSAS, the abundance of a phase is expressed as a mass rather than a molar ratio.

Diffraction data recorded in the angular range 10-40° was used for the generation of the Le Bail fits and structure solution. As discussed in section 2.1.2, once a good Le Bail fit is achieved, scattering matter is introduced into the refined unit cells manually and through the structure determination process. Initially the DE structure solution was used to solve the crystal structure of one amide from the biphasic powder pattern whilst the crystal structure of the other amide was manually completed. In the work discussed in this chapter, the structure that is actively solved is defined as the 'target' structure and the manually completed structure is defined as the 'recreated' structure. Thus biphasic patterns are simulated for pairs of target and manually recreated structural models.

5.5.2 Biphasic sample 1. Crystalline adipamide and nicotinamide combined in a molar ratio of 1:3

The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample 1 was assigned an *R* factor = 10.3%. The refined lattice parameters of adipamide were assigned the following values: a = 5.150(3)Å, b = 5.64(3)Å, c = 7.03(1)Å, α = 69.18°, β = 85.84(5)° and γ = 73.12° giving a unit cell volume = 182.5Å^3. The refined lattice parameters of nicotinamide were assigned the following values: a = 3.970(3)Å, b = 15.600(2)Å, c = 9.410(4)Å, α = γ = 90°, β = 99.05(3)°, giving a unit cell volume = 575.5Å^3. Due to the crystal symmetry of nicotinamide there are four equivalent molecules inside the unit cell. Using the published crystal structure of nicotinamide, [40] the structure of one molecule was manually completed (the other three are generated by symmetry).

As discussed in section 3.3.1, the crystal structure of adipamide is solved using a model defined by eight parameters. Five DE searches were run and the DE was assigned the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully. The mean *R* factor assigned to these solutions was 17.78%. Figure 5.8 shows two views of the solution that was assigned the *R* factor with the lowest value of 17.74%.

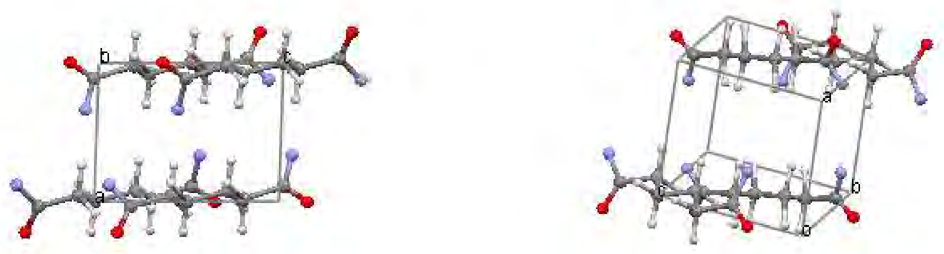


Figure 5.8, The best model structure of adipamide located by direct space structure solution from the biphasic powder pattern recorded for a sample of crystalline adipamide and nicotinamide in a 1:3 molar ratio.

The left hand structure in figure 5.8 is a side-view of the solution. This view shows the molecule orientated along the *b* axis as in the published structure (fig 5.5). Fig 5.8 shows that the solution is correctly orientated inside the unit cell but is shifted slightly along the *b* axis. The right hand structure in figure 5.8 is an end-on-view of the solution. Both views show that the solution is in the correct confirmation forming the characteristic amide dimer motif. It is likely that this solution would refine successfully.

The structure solution of nicotinamide was attempted using the same biphasic pattern recorded for sample 1. Using the published crystal structure of adipamide, [39] the structure of adipamide was manually completed and the structure of nicotinamide deleted. Since nicotinamide adopts $P2_1/c$ symmetry, three parameters define the position of a model inside the unit cell, three parameters define the orientation of the model and one torsion parameter defines the rotation of the amide group with respect to the pyridine ring; therefore a total of seven parameters are used to define the structural model. Five DE searches were used to solve the crystal structure of nicotinamide. The DE was assigned the control parameters, *NP* = 140, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully and were assigned a mean *R* factor = 16.54%. Figure 5.9

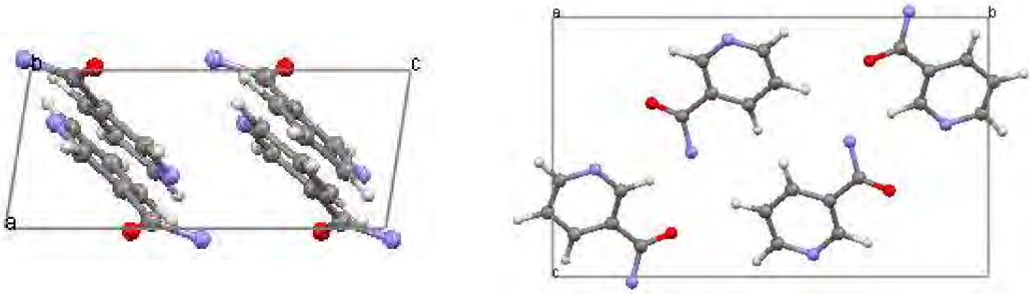


Figure 5.9, The best model structure of nicotinamide located by direct space structure solution from the biphasic powder pattern recorded for sample 1.

shows two views of the solution that was assigned an *R* factor with the lowest value of 16.07%.

Comparing the solution shown in fig 5.9 with the published structure shown in fig 5.6 demonstrates that the solution is in the correct orientation but that there is a slight translation of the model along the *b* axis. Fig 5.9 also shows the alternate flip of the amide group. In the published structure the two nitrogen atoms adopt a *syn* conformation whereas in this solution the amide group has rotated so the nitrogen atoms adopt an *anti* conformation. Since O and N atoms have nearly the same X-ray scattering power it is difficult to detect this flip during the DE search which uses the *R* factor to evaluate possible solutions. However, it is strightforward to determine the correct orientation during Rietveld refinement by manually flipping between the *syn* and *anti* conformations and selecting the solution that is assigned the lowest *R* factor.

5.5.3 Biphasic sample 2. Crystalline adipamide and nicotinamide combined in a molar ratio of 3:1

The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample 2 was assigned an *R* factor = 12.2%. The refined lattice parameters of adipamide were assigned the following values: a = 5.126(9)Å, b = 5.589(1)Å, c = 7.063(9)Å, α = 69.5°, β = 87.1°, γ = 75.4° giving a unit cell volume = 183.3 Å^3. The refined lattice parameters of nicotinamide were assigned the following values: a = 3.984(6)Å, b = 15.72(3)Å, c = 9.44(1)Å, α = γ = 90.0°, β = 99.1, giving a unit cell volume = 583.8 Å^3.

Using the published crystal structure of nicotinamide, [40] the structure of one molecule was manually completed, and as described previously, the crystal structure of adipamide was solved using a model defined by eight parameters. Five DE searches were run with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99 and Gmax = 2000. The five searches converged successfully and were assigned a mean *R* factor of 18.91%. Figure 5.10 shows two views of the solution that was assigned an *R* factor with the lowest value=18.83%. Comparing the side view of the solution shown in fig 5.10 with the published structure shown in fig 5.5, demonstrates that the solution is tilted slightly away from the *b* axis. The end-on-view shows that the solution is in the wrong position and that the confirmation is incorrect.

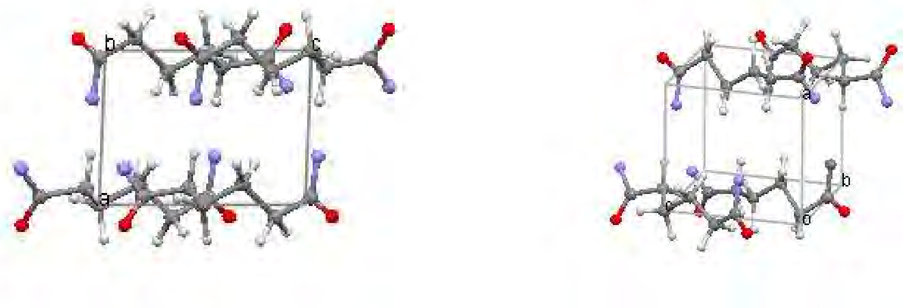


Figure 5.10, The best model structure of adipamide located by direct space structure solution from the biphasic powder pattern recorded for sample 2.

The structure solution of nicotinamide was attempted using the same pattern recorded for sample 2. Using the published crystal structure of adipamide, [39] the structure of adipamide was manually completed and the structure of nicotinamide deleted. As previously discussed, the crystal structure of nicotinamide was solved using a model defined by seven parameters. Five DE searches were used to solve the crystal structure of nicotinamide. The DE was assigned the control parameters, *NP* = 140, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully and the five solutions gave a mean *R* factor = 34.18%. Figure 5.11 shows two views of the solution that was assigned an *R* factor with the lowest value of 34.15%.

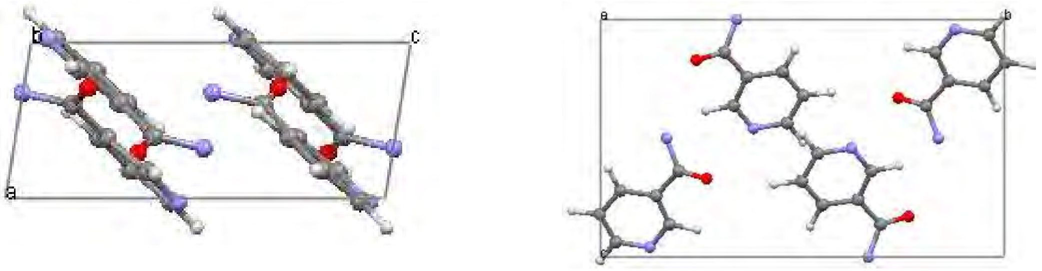


Figure 5.11, The best model structure of nicotinamide located by direct space structure solution from the biphasic powder pattern recorded for sample 2.

Comparison of the solution shown in fig 5.11 with the published structure shown in fig 5.6 demonstrates that although the solution is in roughly the correct position the orientation is wrong. Fig 5.11 shows that the solution has rotated, so that atoms forming the amide group and part of the aromatic ring occupy space that should be occupied by atoms forming the ring, and that part of the ring is where the amide group should be. This incorrect orientation is probably due to C, N and O having nearly equal X-ray scattering factors. Comparing fig 5.11 with the published structure fig 5.6, shows that although the C, N and O atoms are in the wrong place, they each occupy space that should be occupied by a different C, N or O atom. This is a good example of a search that has converged in a local minimum. Fig 5.11 also shows that the amide group has flipped and adopts the *anti* conformation as in fig 5.9.

5.5.4 Simultaneous Crystal Structure Solution and Quantitative Phase Analysis

Section 5.5.2 demonstrates that providing the abundance of each phase is known to a reasonable level of accuracy, direct space structure solution can solve a crystal structure from a biphasic powder diffraction pattern. Next, simultaneous structure solution and quantitative phase analysis was attempted. Each time a pair of target and recreated structural models was evaluated by Rietveld refinement, the value of each phase-specific scale factor was refined (as described in section 5.3.1) to improve the fit between the real and simulated biphasic patterns. GSAS then used the refined scale factors to determine the relative abundance of each phase.

The technique was initially tested using the diffraction data recorded for sample 1. The Le Bail fit previously generated for this biphasic pattern was used and the phase ratio parameters were each assigned a value of 1. An attempt was made to improve the Le Bail fit by using the Rietveld refinement function of GSAS to refine the value of the phase ratio parameters. However, this failed to improve the profile fit. Examination revealed that the value of the phase ratio parameters had not changed during refinement. Since the purpose of the Le Bail technique is to fit the diffraction pattern by refinement of profile variables that describe the pattern without any scattering matter in the unit cell[s] each phase-specific scale factor is assigned an arbitrary value. Hence it is not possible to determine the abundance of each phase during the Le Bail fit.

Using the published crystal structure of nicotinamide, [40] a complete structure of one molecule of nicotinamide was manually created and the crystal structure of adipamide solved using a model defined by eight parameters. Five DE searches were run with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully and the five solutions assigned a mean *R* factor = 16.42%. Figure 5.12 shows two views of the solution that was assigned an *R* factor with the lowest value of 16.33%. For comparison, the best model of adipamide solved from the pattern recorded for sample 1 without refinement of the phase ratio (shown in figure 5.8) was assigned an *R* factor = 17.74%.

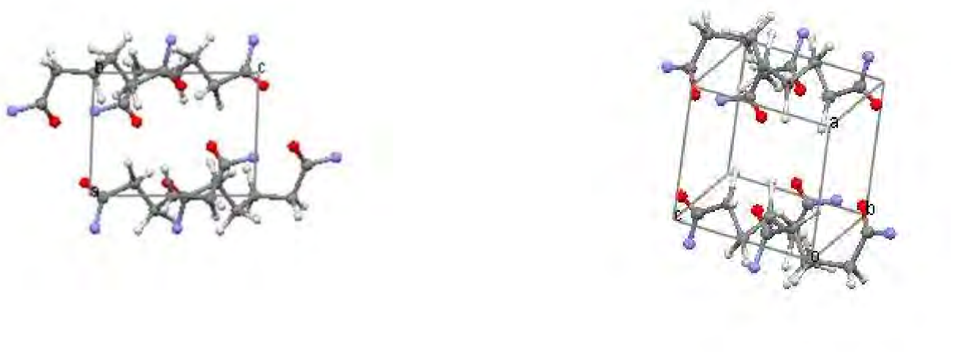


Figure 5.12, The best structure of adipamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 1.

The two views of the solution show that the model is in the correct position and orientation. However, the right hand end-on view shows that the conformation is wrong. The conformation of the carbon chain is distorted so that both amide groups are orientated in the same direction. In the published structure the amide groups are orientated in opposite directions forming the dimer motif. It is not possible to say if Rietveld refinement of this solution would be successful.

The structure solution of nicotinamide and simultaneous quantitative phase analysis from the same powder pattern recorded for sample 1 was then attempted, using the published crystal structure of adipamide [39] manually entered and the complete structure of nicotinamide unknown. As before, the crystal structure of nicotinamide was solved using a model defined by seven parameters. Five DE searches were used to solve the crystal structure of nicotinamide, and assigned the control parameters, *NP* = 140, *F* = 0.5, *K* = 0.99, Gmax = 2000. Five searches converged successfully and the five solutions were assigned a mean *R* factor = 14.66%. Figure 5.13 shows two views of the solution that was assigned an *R* factor with the lowest value of 14.46%. For comparison, the best model structure of nicotinamide solved from the pattern recorded for sample 1 without refinement of the phase ratio (figure 5.9) was assigned an *R* factor = 16.07%. Comparing the solution shown in fig 5.13 with the published structure fig 5.6, demonstrates that the solution is in the correct position and orientation but there has been a crystallographic translation of the unit cell from +*b* to -*b*. Fig 5.13 also shows that the amide group has adopted the incorrect *anti* conformation. Since this solution would refine to the correct structure it can be considered as successful.

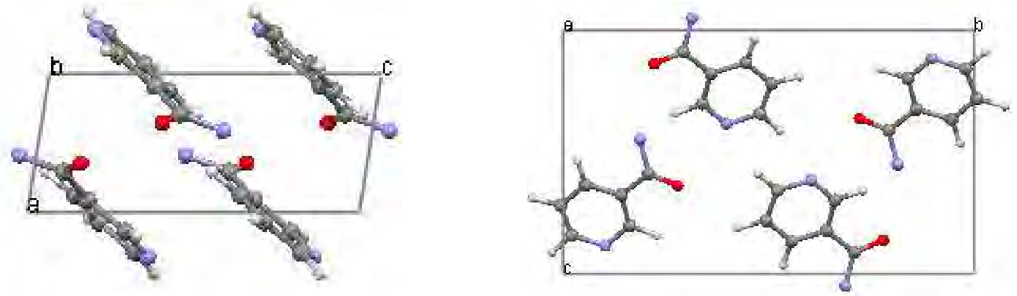


Figure 5.13, The best model structure of nicotinamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 1.

Determining the accuracy of the quantitative phase analysis.

Sample 1 was prepared by combining adipamide and nicotinamide in the molar ratio of 1:3 (adipamide, $C_6H_{12}N_2O_2$, molar mass = 144.16g/mol^{-1}; nicotinamide, $C_6H_6N_2O$, molar mass = 122.13g/mol^{-1}; molar mass of sample 1 = 510.55g/mol^{-1}). Table 5.1 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA function of GSAS during the structure solution of adipamide. The first column records the *R* factor assigned to the final solution located, the second column records the abundance of the adipamide phase and the third column records the abundance of the nicotinamide phase. The final row shows the mean value calculated for each column.

Table 5.1 shows that during solution of adipamide the Rietveld based QPA has decreased the abundance of the adipamide phase and increased the abundance of the nicotinamide phase from the prepared ratio of 1:3. Since the adipamide solution shown in figure 5.12 adopts the wrong conformation, in this case a biphasic pattern is simulated for a wrong structure of adipamide and a correct structure of nicotinamide. It is possible that some diffraction peaks resulting from simulated diffraction by the wrong adipamide model were in the wrong position or had the wrong intensity.

Table 5.1, Structure solution of adipamide and quantitative phase analysis for sample 1.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	16.55%	0.207	0.793
2	16.39%	0.208	0.792.
3	16.43%	0.210	0.790.
4	16.39%	0.207	0.793.
5	16.33%	0.209	0.791.
Mean	16.42%	0.208	0.792

Molar ratios calculated from mass ratios.
Molar ratio = (mass ratio of crystal * molar mass of sample)/molar mass of crystal.
Molar ratio adipamide = (0.208*510.55)/144.16 = 0.74mol.
Molar ratio nicotinamide = (0.792*510.55)/122.13 = 3.31mol.

This could potentially cause the Rietveld based QPA to improve the fit between simulated and real biphasic patterns by decreasing the abundance of adipamide to compensate for the wrong structure.

Table 5.2 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.2, Structure solution of nicotinamide and quantitative phase analysis for sample 2.

Run	Converged best *R*	NicMs*R*	AdipMs*R*
1	14.96%	0.780	0.213
2	14.47%	0.791	0.209
3	14.46%	0.791	0.209
4	14.96%	0.780	0.220
5	14.47%	0.791	0.209
Mean	14.66%	0.787	0.212

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.212*510.55)/144.16 = 0.75mol.
Molar ratio nicotinamide = (0.787*510.55)/122.13 = 3.29mol.

Table 5.2 also shows that during the structure solution of nicotinamide, the abundance of nicotinamide has been increased and the abundance of adipamide decreased. However, during solution of nicotinamide, the manually completed structure of adipamide is correct. Thus in this case, biphasic patterns are simulated for a structural model of nicotinamide and a correct structure of adipamide. Therefore all peaks resulting from simulated diffraction by adipamide should be correct and the Rietveld based QPA should not need to improve the fit between simulated and real patterns by decreasing the abundance of the correct adipamide phase.

Another possible cause for the error in this QPA technique is that diffraction data recorded on different diffractometers and under different conditions is combined in these experiments. In these experiments, lattice parameters of two crystal phases are refined using data extracted from a biphasic powder pattern recorded using one diffractometer under ambient conditions. However, the published crystal structures used for the non-target phase have been recorded under different diffraction conditions. The single crystal pattern recorded for nicotinamide was recorded at a higher temperature than the powder pattern recorded for adipamide. Due to thermal effects, it is unlikely that the atoms from the published structures are placed in exactly the correct positions for the experimental powder diffraction data used here. This causes a constant amount of mismatch between simulated and real biphasic patterns and could result in the Rietveld refinement of an incorrect phase ratio. Experimental error could also be the reason for the discrepancy between the experimentally prepared and QPA determined ratio, i.e. not combining adipamide and nicotinamide in the exact 1:3 ratio as intended.

However, figures 5.12 and 5.13 suggest that it is possible to use the direct space method to solve a crystal structure with reasonable accuracy directly from a biphasic pattern without subtracting peaks corresponding to the other phase. Tables 5.1 and 5.2 also demonstrate that it is not necessary to know the abundance of each phase before attempting structure solution and that it is possible to determine the abundance of each phase with reasonable accuracy using Rietveld based QPA simultaneously with structure solution.

5.5.5 Biphasic Sample 3. Crystalline adipamide and oxamide combined in a molar ratio of 1:1

Initially structure solution was attempted whilst the values of the phase ratio parameters were fixed at the ratio in which the sample was prepared as in section 5.5.2. The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample 3 was assigned an *R* factor = 16.5%. The refined lattice parameters of adipamide were assigned the following values: a = 5.105(1)Å, b = 5.565(1)Å, c = 7.042(1)Å, α = 69.5°, β = 87.1°, γ = 75.4° giving a unit cell volume = 181.2Å^3. The refined lattice parameters of oxamide were assigned the following values: a = 3.618(0)Å, b = 5.176(1)Å, c = 5.648(0)Å, α = 83.9°, β = 114.0°, γ = 115.0° giving a unit cell volume = 87.4Å^3.

Using the published crystal structure of oxamide, [41] the complete structure of oxamide was manually created and the crystal structure of adipamide was solved using a model defined by eight parameters. Five DE searches were used and assigned the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions that were assigned a mean *R* factor = 18.24%. Figure 5.14 shows two views of the solution that was assigned an *R* factor with the lowest value = 18.14%.

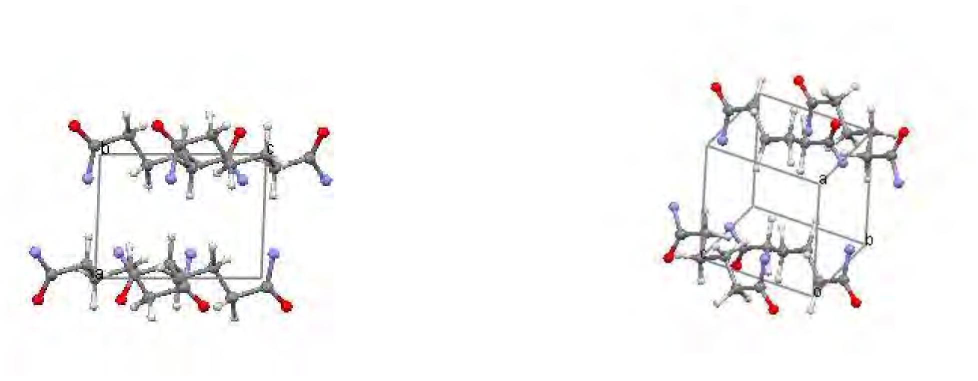


Figure 5.14, The best model structure of adipamide located by direct space structure solution using a biphasic powder pattern recorded for a sample of crystalline adipamide and oxamide in a 1:1 molar ratio.

The left hand view shows the solution running correctly along the b axis and both views show that the solution is correctly positioned. The right hand view shows that there is a slight distortion of the carbon chain but it is expected that this could be successfully resolved during refinement.

The structure solution of oxamide was also attempted using the same pattern recorded for sample 3. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the structure of oxamide was determined. Although oxamide adopts P-1 symmetry, here structure solution was attempted in P1 symmetry without the constraint of an internal inversion centre. Therefore the position of the model inside the unit cell is arbitrary and three positional parameters are not required. Three parameters were used to define the orientation of the model and three more to define its intramolecular flexibility, thus six parameters were used in total to define the model of oxamide. Five DE searches were used with the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 20.32%. Figure 5.15 shows two views of the solution that was assigned an *R* factor with the lowest value of 20.23%.

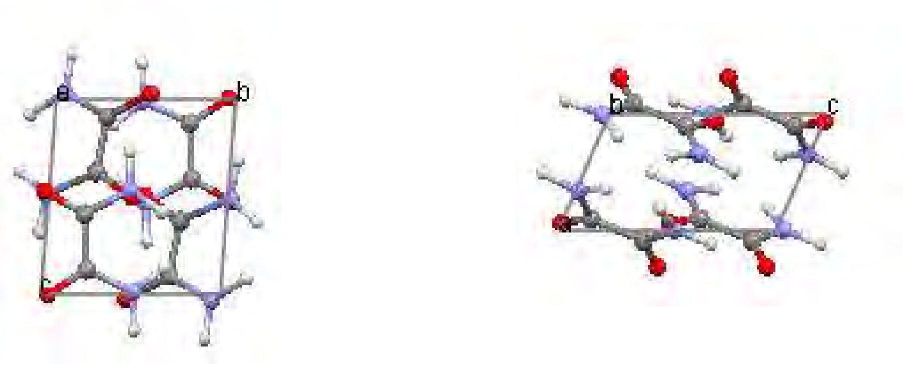


Figure 5.15, The best model structure of oxamide located by direct space structure solution from the biphasic powder pattern recorded for sample 3.

Comparison of the solution in fig 5.15 with the published structure of oxamide in fig 5.7 shows that this solution is incorrect. Fig 5.7 shows that oxamide is planar with a *trans* relationship between the nitrogen atoms of the two amide groups. The published structure is located in the middle of the unit cell and also lies in the a plane. Fig 5.15 shows that although the solution is planar, the two amide groups adopt a *cis* conformation. The solution is not quite located in the middle of the unit cell and although the model is centred on the a plane, it is slightly tilted out.

Next, simultaneous structure solution and QPA was attempted as in section 5.5.4. Simultaneous structure solution of adipamide and QPA was then attempted using the biphasic pattern recorded for sample 3. Using the published crystal structure of oxamide,[41] the complete structure of oxamide was manually created and the structure of adipamide solved using a model defined by eight parameters. Five DE searches were run with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. Five searches converged successfully. The five solutions were assigned a mean *R* factor = 17.20%. Figure 5.16 shows two views of the solution that was assigned an *R* factor with the lowest value of 17.14%. For comparison, the best model of adipamide located by structure solution from the biphasic pattern recorded for sample 3 without simultaneous quantitative phase analysis (figure 5.14) was assigned an *R* factor = 18.14%.

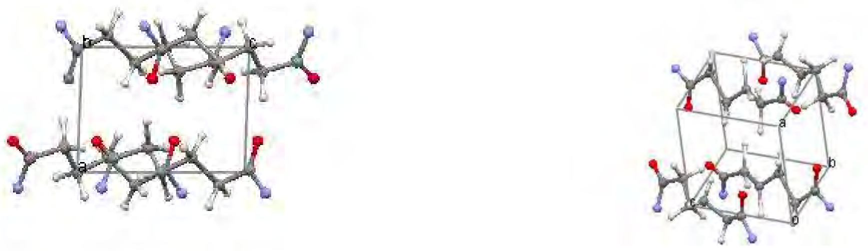


Figure 5.16, The best model structure of adipamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 3.

The solution shown in fig 5.16 compares favourably with the published structure fig 5.5. The solution is correctly positioned and runs along the b axis. The conformation of the carbon chain is nearly correct and it is likely that refinement of this structure would be successful.

Simultaneous structure solution of oxamide and quantitative phase analysis was then attempted using the biphasic pattern recorded for sample 3. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the structure of oxamide was solved using a model defined by six parameters. Five DE searches were run with the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully with the five solutions assigned a mean *R* factor = 18.87%. Figure 5.17 shows two views of the solution that was assigned an *R* factor with the lowest value of 18.79%. For comparison, the best model of oxamide located by structure solution using the biphasic pattern recorded for sample 3 without simultaneous quantitative phase analysis (figure 5.15) was assigned an R factor = 20.23%.

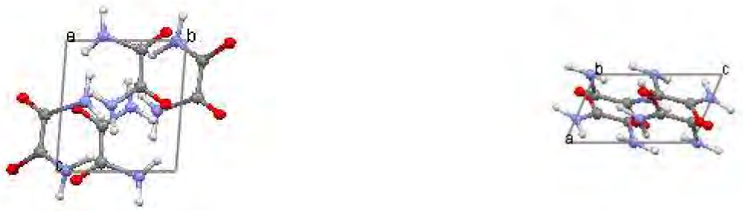


Figure 5.17, The best model structure of oxamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 3.

Comparison of the solution in fig 5.17 with the published structure fig 5.7 demonstrates that this solution is incorrect. Although the solution is planar, the two amide groups adopt a *cis* conformation. The position of the solution is also incorrect, not in the middle of the unit cell and tilted out of the a plane. Although the solution has shifted position by roughly half a unit cell, the crystal symmetry means that this shifted position is nearly equivalent with the correct position. It is unlikely that this solution would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Sample 3 was prepared by combining adipamide and oxamide in a molar ratio of 1:1 (oxamide, $C_2H_4N_2O_2$, molar mass = 88.07g/mol^{-1}; adipamide, $C_6N_2O_2H_{12}$, molar mass = 144.16g/mol^{-1}; molar mass of sample 3 = 232.23g/mol^{-1}).

Table 5.3 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during the structure solution of adipamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.3, Structure solution of adipamide and quantitative phase analysis for sample 3**.**

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	17.23%	0.538	0.462
2	17.20%	0.541	0.459
3	17.19%	0.538	0.462
4	17.24%	0.538	0.462
5	17.14%	0.539	0.461
Mean	17.20%	0.539	0.461

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.539*232.23)/144.16 = 0.87mol.
Molar ratio oxamide = (0.461*232.23)/88.07 = 1.22mol.

Table 5.4 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of oxamide. Data for each structure solution calculation is presented as in table 5.1. Although the solution for adipamide (fig 5.16) is nearly correct and much better quality than the solution for oxamide (fig 5.17), tables 5.3 and 5.4 show that in both cases the QPA has increased the abundance of oxamide and decreased the abundance of adipamide from the prepared 1:1 ratio.

Table 5.4, Structure solution of oxamide and quantitative phase analysis for sample 3.

Run	Converged best *R*	OxamMs*R*	AdipMs*R*
1	18.92%	0.479	0.521
2	18.92%	0.479	0.521.
3	18.80%	0.492	0.508
4	18.92%	0.480	0.520
5	18.79%	0.495	0.505
Mean	18.87%	0.485	0.515

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.515*232.23)/144.16 = 0.83mol.
Molar ratio oxamide = (0.485*232.23)/88.07 = 1.28mol.

5.5.6 Biphasic sample 4. Crystalline adipamide and oxamide combined in a molar ratio of 1:2

Initially the structure solution was attempted whilst the values of the phase ratio parameters were fixed using the ratio in which the sample was prepared as in section 5.5.2. The Le Bail fit generated for the biphasic pattern recorded for sample 4 was assigned an *R* factor = 15.0%. The refined lattice parameters of adipamide were assigned the following values: a = 4.975(2)Å, b = 5.611(3)Å, c = 6.973(6)Å, α = 69.5°, β = 86.9°, γ = 75.5° giving a unit cell volume = 176.4 Å^3. The refined lattice parameters of oxamide were assigned the following values: a = 3.618(1)Å, b = 5.181(2)Å, c = 5.642(1)Å, α = 83.8°, β = 113.8°, γ = 115.0° giving a unit cell volume = 87.4 Å^3.

Using the published crystal structure of oxamide, [41] the complete structure of oxamide was manually created and the structure of adipamide solved using a model defined by eight parameters. Five DE searches were run with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully with the five solutions assigned a mean *R* factor = 19.64%. Figure 5.18 shows two views of the solution that was assigned an *R* factor with the lowest value of 19.51%.

Comparison of fig 5.18 with the published structure fig 5.5 demonstrates that this is a poor solution. Although the solution is orientated along the b axis the conformation of the carbon chain is significantly distorted. It is unlikely that this solution would refine successfully.

Figure 5.18, The best model structure of adipamide located by direct space structure solution from the biphasic powder pattern recorded for sample 4.

The structure solution of oxamide was then attempted from the same biphasic pattern recorded for sample 4. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the crystal structure of oxamide solved using a model defined by six parameters. Five DE searches were run with the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. Four searches converged successfully and the four solutions were assigned a mean *R* factor = 19.27%. Figure 5.19 shows two views of the solution that was assigned an *R* factor with the lowest value of 19.08%.

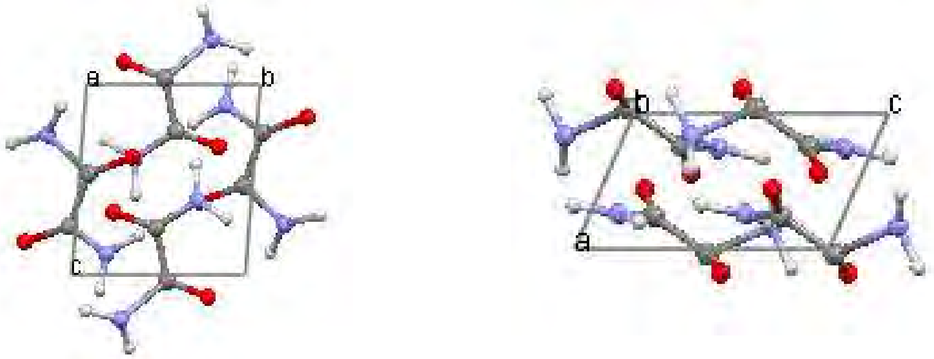


Figure 5.19, The best model structure of oxamide located by direct space structure solution from the biphasic powder pattern recorded for sample 4**.**

Comparison of the solution in fig 5.19 with the published structure fig 5.7 shows that the solution has adopted the correct *trans* relationship between the two amide groups. However, the solution is not planar. The solution is correctly centred on the a plane but is tilted slightly out of the plane. Although the solution has shifted position by roughly half a unit cell, the crystal symmetry of oxamide means that this shifted position is nearly equivalent with the correct position. It is likely that this solution would refine successfully.

Simultaneous structure solution of adipamide and quantitative phase analysis was attempted from the biphasic pattern recorded for sample 4. Using the published crystal structure of oxamide, [41] the complete structure of oxamide was manually created and the crystal structure of adipamide was solved using the model defined by eight parameters. Five DE searches were run with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 19.07%. Figure 5.20 shows two views of the solution that was assigned an *R* factor with the lowest value of 18.94%. For comparison, the best model structure of adipamide solved using the pattern recorded for sample 4 without refinement of the phase ratio (figure 5.18) was assigned an *R* factor = 19.51%.

Figure 5.20, The best model structure of adipamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 4.

Comparison of the solution in fig 5.20 with the published structure fig 5.5 demonstrates that this is a poor solution. Although the solution is orientated along the b axis the conformation of the carbon chain is significantly distorted. It is unlikely that this solution would refine successfully.

Simultaneous structure solution of oxamide and quantitative phase analysis was then attempted from the biphasic pattern recorded for sample 4. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the structure of oxamide solved using the model defined by six parameters. Five DE searches were run with the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 19.18%. Figure 5.21 shows two views of the solution that was assigned an *R* factor with the lowest value of 19.04%. For comparison, the best model structure of oxamide solved from the pattern recorded for sample 4 without refinement of the phase ratio (figure 5.19) was assigned an *R* factor = 19.08%.

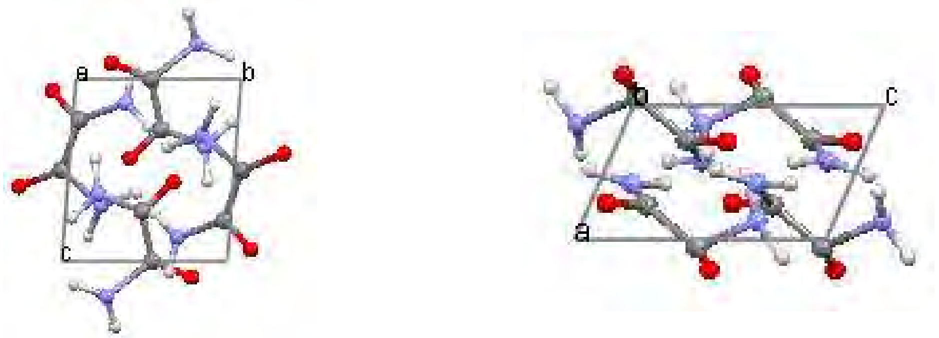


Figure 5.21, The best model structure of oxamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 4.

The solution shown in fig 5.21 is similar to the solution shown in fig 5.19, however, in fig 5.21 the incorrect *cis* conformation has been adopted. It is likely that during refinement, the correct *trans* conformation could be determined by manually flipping one amide group and that overall refinement of solution 5.21 would be successful.

Determining the accuracy of the quantitative phase analysis.

Sample 4 was prepared by combining crystalline adipamide and oxamide in a molar ratio of 1:2 (molar mass of sample 4 = 320.3g/mol^{-1}).

Table 5.5 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of adipamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.5, Structure solution of adipamide and quantitative phase analysis for sample 4.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	19.10%	0.563	0.437
2	19.04%	0.572	0.428
3	19.21%	0.565	0.435
4	18.94%	0.573	0.428
5	19.04%	0.572	0.428
Mean	19.07%	0.569	0.431

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.569*320.3)/144.16 = 1.26mol.
Molar ratio oxamide = (0.431*320.3)/88.07 = 1.57mol.

Fig 5.20 shows that the best model of adipamide located by these searches is a poor solution. However, table 5.5 shows that the QPA has increased the abundance of adipamide and decreased the abundance of oxamide from the prepared 1:2 ratio. This is an interesting result, since in this case the manually created structure of oxamide is correct. Hence in this case the QPA has improved the fit between simulated and real biphasic patterns by increasing the abundance of a wrong solution and decreasing the abundance of the correct structure.

Table 5.6 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of oxamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.6, Structure solution of oxamide and quantitative phase analysis for sample 4.

Run	Converged best *R*	OxamMs*R*	AdipMs*R*
1	19.28%	0.607	0.393.
2	19.09%	0.615	0.385
3	19.04%	0.609	0.391
4	19.38%	0.606	0.394
5	19.09%	0.615	0.385
Mean	19.18%	0.610	0.390

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.390*320.3)/144.16 = 0.87mol.
Molar ratio oxamide = (0.610*320.3)/88.07 = 2.22mol.

Figure 5.21 shows that although the solution has adopted the wrong *cis* conformation it is a good solution that could be refined successfully. Table 5.6 shows that in this case the QPA has increased the abundance of oxamide and decreased the abundance of adipamide from the prepared 1:2 ratio. Hence in this case the QPA has increased the abundance of a nearly correct structure and decreased the abundance of the correct structure.

5.5.7 Biphasic sample 5. Crystalline adipamide and oxamide combined in a molar ratio of 2:1

Initially the structure solution was attempted whilst the values of the phase ratio parameters were fixed using the ratio in which the sample was prepared. The Le Bail fit generated for the biphasic pattern recorded for sample 5 was assigned an *R* factor = 13.6%. The refined lattice parameters of adipamide were assigned the following values: a = 5.107(6)Å, b = 5.57(0)Å, c = 7.042(9)Å, α = 69.6°, β = 87.1°, γ = 75.4° giving a unit cell volume = 181.5Å^3. The refined lattice parameters of oxamide were assigned the following values: a = 3.621(4)Å, b = 5.179(9)Å, c = 5.651(7)Å, α = 83.9°, β = 114.0°, γ = 115.0° giving a unit cell volume = 87.5Å^3.

Using the published crystal structure of oxamide, [41] the complete structure of oxamide was manually created and the crystal structure of adipamide solved using the model defined by eight parameters. Five DE searches were run with control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 17.39%. Figure 5.22 shows two views of the solution that was assigned an *R* factor with the lowest value of 17.16%.

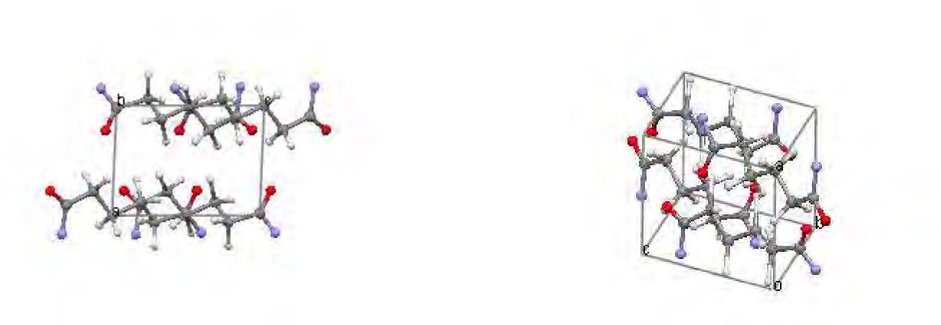


Figure 5.22, The best model structure of adipamide located by direct space structure solution from the biphasic powder pattern recorded for sample 5.

Comparison of the solution shown in fig 5.22 with the published structure fig 5.5 demonstrates that this is a good solution. The solution is orientated along the b axis and the carbon chain is not significantly distorted. Although both amide groups are flipped the same way, it would be possible to determine which group is incorrect by manually flipping each group during refinement. It is likely that Rietveld refinement of this solution would be successful.

Structure solution of oxamide was then attempted from the same pattern recorded for sample 5. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the structure of oxamide solved using the model defined by six parameters. Five DE searches were used and assigned the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully. The five solutions were assigned a mean *R* factor = 16.12%. Figure 5.23 shows two views of the solution that was assigned an *R* factor with the lowest value of 16.01%.

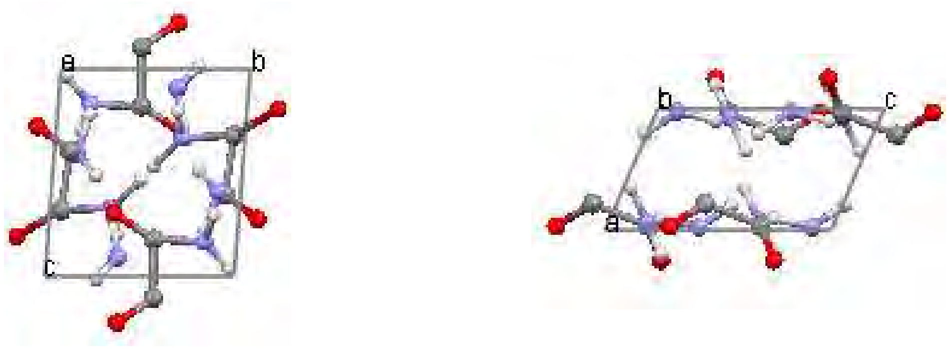


Figure 5.23, The best model structure of oxamide located by direct space structure solution from the biphasic powder pattern recorded for sample 5.

Comparison of this solution with the published structure fig 5.7 demonstrates that this solution is wrong and that refinement of this solution is unlikely to be successful. The solution in fig 5.23 does lie in the a plane but the conformation of the model is not planar. Simultaneous structure solution of adipamide and quantitative phase analysis was then attempted from the pattern recorded for sample 5. Using the published crystal structure of oxamide, [41] the complete structure of oxamide was manually created while the structure of adipamide was solved using the model defined by eight parameters. Five DE searches were used with the control parameters, *NP* = 160, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 16.57%. Figure 5.24 shows two views of the solution that was assigned an *R* factor with the lowest value of 16.50%. For comparison, the best model structure of adipamide solved from the pattern recorded for sample 5 without refinement of the phase ratio (figure 5.22) was assigned an *R* factor = 17.16%.

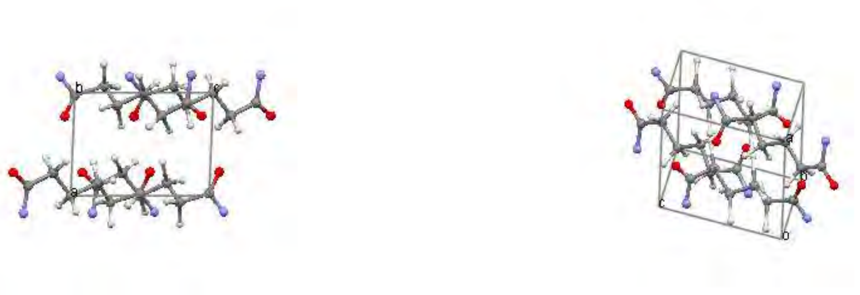


Figure 5.24, The best model structure of adipamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 5.

The solution shown in fig 5.24 is similar to the solution in fig 5.22. Solution 5.24 is orientated along the b axis and the carbon chain is not significantly distorted. Although both amide groups are flipped the same way, it would be possible to determine which group is incorrect by manually flipping each group during refinement. It is likely that

Rietveld refinement of solution 5.24 would be successful.

Simultaneous structure solution of oxamide and quantitative phase analysis was then attempted from the pattern recorded for sample 5. Using the published crystal structure of adipamide, [39] the complete structure of adipamide was manually created and the structure of oxamide solved using the model defined by six parameters. Five DE searches were used with the control parameters, *NP* = 120, *F* = 0.5, *K* = 0.99, Gmax = 2000. The five searches converged successfully giving five solutions with a mean *R* factor = 15.67%. Figure 5.25 shows two views of the solution that was assigned an *R* factor with the lowest value of 15.55%. For comparison, the best model structure of oxamide solved from the pattern recorded for sample 5 without refinement of the phase ratio (figure 5.23) was assigned an *R* factor = 16.01%.

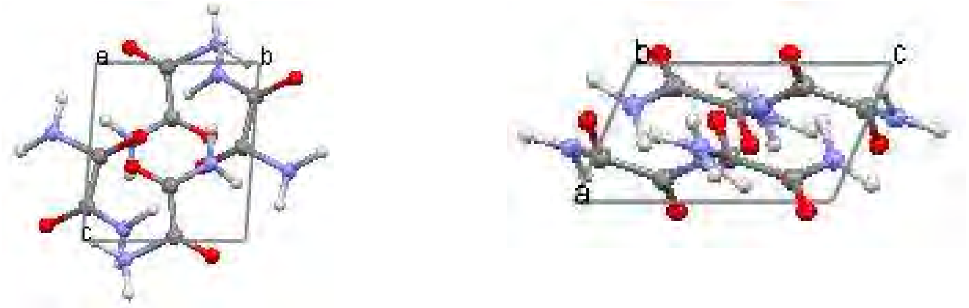


Figure 5.25, The best model structure of oxamide located by direct space structure solution and simultaneous quantitative phase analysis from the biphasic powder pattern recorded for sample 5.

The solution shown in fig 5.25 is similar to the solution shown in fig 5.19. Solution 5.25 has adopted the correct *trans* relationship between the two amide groups. However, the solution is not planar. The solution is correctly centred on the a plane but is tilted slightly out of the plane. Although the solution has shifted position by roughly half a unit cell, the crystal symmetry means that this shifted position is nearly equivalent with the correct position. It is likely that solution 5.25 would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Sample 5 was prepared by combining adipamide and oxamide in a molar ratio of 2:1 (molar mass of sample 5 = 376.4g/mol^{-1}).

Table 5.7 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of adipamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.7, Structure solution of adipamide and quantitative phase analysis for sample 5.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	16.50%	0.819	0.181
2	16.62%	0.819	0.181
3	16.51%	0.819	0.181
4	16.62%	0.819	0.181
5	16.60%	0.819	0.181
Mean	16.57%	0.819	0.181

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.819*376.4)/144.16 = 2.1mol.
Molar ratio oxamide = (0.181*376.4)/88.07 = 0.8mol.

Figure 5.24 shows that the best model for adipamide located during these searches is a good solution. Table 5.7 shows that during these searches the QPA has slightly increased the abundance of adipamide and decreased the abundance of oxamide from the prepared 2:1 ratio.

Table 5.8 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during structure solution of oxamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.8, Structure solution of oxamide and quantitative phase analysis for sample 5.

Run	Converged best *R*	OxamMs*R*	AdipMs*R*
1	15.56%;	0.188	0.812
2	15.95%;	0.172	0.828
3	15.64%;	0.186	0.814.
4	15.67%;	0.188	0.812
5	15.55%;	0.190	0.810
Mean	15.67	0.185	0.815

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.815*376.4)/144.16 = 2.1mol.
Molar ratio oxamide = (0.185*376.4)/88.07 = 0.8mol.

Figure 5.25 shows that the best model of oxamide located by these searches is a good solution. Table 5.8 shows that during these searches the QPA has slightly increased the abundance of adipamide and decreased the abundance of oxamide from the prepared 2:1 ratio.

5.5.8 Summary

Section 5.5.2 demonstrates that if a biphasic powder pattern is recorded for a material containing one known and one unidentified crystal phase, providing that the pattern can be indexed to reveal the lattice parameters of the unidentified phase, it is sometimes possible to use direct space methods to solve the structure of the unidentified phase directly from the biphasic pattern without the necessity of subtracting peaks corresponding to the known phase from the biphasic pattern.

Although our combined direct space and QPA method is capable of locating a model solution that is a fair representation of the real crystal structure and estimating the abundance of each crystal phase, the structure solution experiments investigated here demonstrate that correct solutions are not found with high success rates. Additionally the QPA technique does not determine the abundance of each phase with reliable accuracy. This is illustrated by the simultaneous structure solution and QPA searches performed using sample 4; prepared by combining adipamide and oxamide in a ratio of 1:2. Fig 5.20 shows that the best model of adipamide located by these searches is a poor solution. However, table 5.5 shows that the QPA has increased the abundance of adipamide and decreased the abundance of oxamide from the prepared 1:2 ratio. Figure 5.21 shows that although the oxamide solution has adopted the wrong *cis* conformation it is a good solution that could be refined successfully. Table 5.6 shows that in this case the QPA has increased the abundance of oxamide and decreased the abundance of adipamide from the prepared 1:2 ratio. Hence in both cases the QPA has improved the fit between simulated and real biphasic patterns by increasing the abundance of the target structure and decreasing the abundance of the manually created correct structure.

This is in contrast with the results of the simultaneous structure solution and QPA searches performed using sample 5; prepared by combining adipamide and oxamide in a ratio of 2:1. Figure 5.24 shows that the best model for adipamide located during these searches is a good solution. Table 5.7 shows that during these searches the QPA has increased the abundance of adipamide and decreased the abundance of oxamide from the prepared ratio. Figure 5.25 shows that the best model of oxamide located by these searches is a good solution. Table 5.8 shows that during these searches the QPA has slightly increased the abundance of adipamide and decreased the abundance of oxamide from the prepared ratio. Hence in both cases the abundance of adipamide is increased.

As discussed in section 5.5.4, a possible reason for these inaccurate results is that diffraction data recorded on different diffractometers and under different conditions is combined in these experiments. In the above experiments, lattice parameters of two crystal phases are refined using data extracted from a biphasic powder pattern recorded using one diffractometer under ambient conditions. However, the published crystal structures used for the non-target phase have been recorded under different diffraction conditions, often single crystal experiments performed at different temperatures. Due to thermal effects, it is unlikely that the atoms from the published structures are placed in exactly the correct positions for the experimental powder diffraction data used here. This causes a constant amount of mismatch between simulated and real biphasic patterns and could result in either a) the Rietveld refinement of an incorrect phase ratio or b) failure of the structure solution.

In the following sections an attempt is made to simultaneously solve both structures directly from the biphasic powder pattern. This means that structural information obtained from different diffraction patterns recorded under different conditions is not needed. Since all the diffraction data used in each experiment comes from one pattern there should be no thermally induced mismatch between simulated and real biphasic patterns.

5.6 Simultaneous multiple direct space structure solution and quantitative phase analysis from biphasic powder diffraction data

The ‘Double’ DE implementation was developed from the original DE implementation. [42] This new ‘Double’ DE was created to investigate the possibility of simultaneously solving two independent crystal structures. To do this, the double DE uses two populations of structural models to represent the different crystal structures. Solutions are evaluated by selecting a model from each population and simulating a biphasic pattern for the pair. The simulated biphasic pattern is compared with the real biphasic pattern using the Rietveld refinement application of GSAS [43] and the resultant *R* factor is assigned to the pair. Quantitative phase analysis by the Rietveld refinement application of GSAS is performed simultaneously during the calculation of each *R* factor.

Since both crystal structures are simultaneously solved in pairs, one landscape represents both crystal structures. However, each model occupies separate landscape dimensions. Thus the total number of dimensions of the landscape is the sum of the number of parameters defining each model, e.g., if two models are each defined by seven and eight parameters respectively, the landscape is defined by a total of 15 dimensions.

5.6.1 Biphasic sample 6. Crystalline adipamide and nicotinamide combined in a molar ratio of 1:1

The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample 6 was assigned an *R* factor = 9.4%. The refined lattice parameters of adipamide were assigned the values: a = 5.184(2)Å, b = 5.654(2)Å, c = 7.044(1)Å, α = 69.4°, β = 86.0° and γ = 72.4° giving a unit cell volume = 184.1 Å^3. The refined lattice parameters of nicotinamide were assigned the values: a = 3.972(0)Å, b = 15.626(4)Å, c = 9.425(1)Å, α = γ = 90.0°, β = 99.0°, giving a unit cell volume = 577.7 Å^3.

As previously discussed, the structure of adipamide was solved using a model defined by eight parameters and the structure of nicotinamide defined by seven, thus the biphasic landscape is defined by 15 dimensions. Due to the assumed complexity of this double structure solution, the value of *NP* is calculated at 40 times the number of parameters required for landscape definition, i.e in this case *NP* = 600. Due to the large population size, it was assumed that a sufficiently rapid convergence rate could be achieved by assigning *F* = 0.3 without significantly increasing the probability of premature convergence. Five DE searches were used to simultaneously solve both structures. The DE was assigned the control parameters: *NP* = 600, *F* = 0.3, K = 0.99, Gmax = 10,000. The five searches converged successfully giving five solutions with a mean *R* factor = 15.59%. Figure 5.26 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.56%.

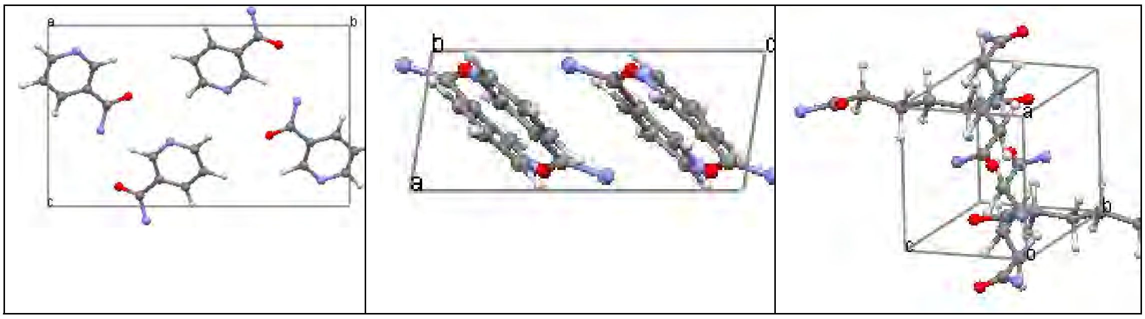


Figure 5.26, The best model structures of nicotinamide and adipamide simultaneously located by direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for a sample of crystalline adipamide and nicotinamide combined in a molar ratio of 1:1.

The two left hand views show the solution for nicotinamide, the right hand view shows the solution for adipamide. The two left hand views show that the solution for nicotinamide has adopted the wrong *anti* conformation. However, the solution is in the correct orientation and position. It is likely that this nicotinamide solution could be refined successfully. The right hand view shows that the solution for adipamide is wrong. Although the conformation of the carbon chain is correct the orientation and position of the model in the unit cell is wrong. It is unlikely that this solution would refine successfully.

<u>Determining the accuracy of the quantitative phase analysis.</u>

The molar mass of sample 6 = 266.29g/mol^{-1}. Table 5.9 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.9 shows that the abundance for the wrong adipamide solution is significantly decreased and the abundance of the correct nicotinamide solution is increased. Hence in these searches, the QPA has improved the fit between simulated and real biphasic patterns by almost eliminating peaks corresponding to adipamide from the simulated biphasic pattern.

Table 5.9, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	15.56%	0.065	0.935
2	15.57%	0.064	0.936
3	15.58%	0.065	0.935
4	15.67%	0.064	0.938
5	15.57%	0.064	0.936
Mean	15.59%	0.064	0.936

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.064*266.29)/144.16 = 0.12mol.
Molar ratio nicotinamide = (0.936*266.29)/122.13 = 2.04mol.

A possible cause for the successful solution for nicotinamide and failed solution for adipamide is the way in which the simultaneous double structure solution and QPA is performed. The *R* factor reflects how well a biphasic diffraction pattern simulated for a ‘pair’ of models fits the real biphasic pattern. Thus it is possible for a biphasic pattern simulated for a pair in which one model is a good and one model a poor representation of the respective real crystal structures to be fitted with the real biphasic pattern by increasing the abundance of the better quality model. From figure 5.26, this is clearly the case; as the nicotinamide solution matches well with the published structure (fig 5.6) but the adipamide solution does not. Hence in this case the abundance of the correct nicotinamide solution has been significantly increased and the abundance of the wrong adipamide solution (which does not exist in reality) significantly decreased.

This is potentially a significant problem if in reality, one phase is significantly more abundant, and if a model representing the more abundant phase is optimised before a model representing the less abundant phase.

A more rigorous strategy for evaluating individual models that can only be evaluated in pairs, is to pair each model with multiple ‘partners’ and calculate an average *R* factor. If a ‘test’ model that is a good representation of the real crystal structure is evaluated with multiple ‘partners’, the test model is more likely to be assigned an average *R* factor with a relatively low value. Conversely, if a test model that is a poor representation of the real crystal structure is evaluated with multiple partners, the test model is more likely to be assigned an average *R* factor with a relatively high value. Using such a strategy increases the probability that better quality models are located and reduces the probability that the fit between simulated and real biphasic patterns is improved by excessively increasing the abundance of a better quality model.

As the number of partners that a test model is evaluated with increases, the probability that the test model is assigned an average *R* factor with an appropriate value increases. Ideally, each model in each population is paired and evaluated with every model in the complementary population. However, the computational effort required to evaluate all combinations is significant, making this exhaustive selection impractical. An alternative selection strategy is an ‘elitist’ strategy.

5.6.2 Elitist selection strategy

In this implementation, two populations of models representing different crystal structures are generated. For this discussion, the two populations are labelled ‘A’ and ‘B’. Each initial model in population ‘A’ is paired with a different initial model in population ‘B’ and evaluated. The pair that is assigned an *R* factor with the lowest value is identified as the best pair. Model ‘A’ is copied from the best pair and stored as ‘elite model A’.

For a number of generations defined by the parameter elite paring ‘EP’, population ‘B’ evolves. Each child produced by parents in population ‘B’ is evaluated in combination with ‘elite model A’. If the *R* factor assigned to the child and ‘elite model A’ has a lower value than the *R* factor assigned to the parent ‘B’ paired with its previous partner ‘A’, the child replaces the parent ‘B’. After ‘EP’ generations, evolution of population ‘B’ is paused. The model in population ‘B’ that is assigned the lowest *R* factor when paired with ‘elite model A’ is then selected as ‘elite model B’. The procedure is reversed and for ‘EP’ generations, population ‘A’ evolves and the children paired and evaluated with ‘elite model B’. After population ‘A’ has evolved for ‘EP’ generations, both populations evolve simultaneously. A child produced by a parent ‘A’ is evaluated with a child produced by a parent ‘B’. Both children replace the parents if the *R* factor assigned to the pair of children has a lower value than the *R* factor assigned to the pair of parents.

Simultaneous evolution of both populations continues until convergence or until Gmax generations have been calculated. The value of ‘EP’ is defined by the user before the structure solution calculation.

Five DE searches using this elitist strategy and quantitative phase analysis were used to simultaneously solve the structures of adipamide and nicotinamide using the biphasic pattern recorded for sample 6. The searches were assigned the control parameters *NP* = 600, *F* = 0.3, *EP* = 2, *K* = 0.99, Gmax = 10,000. Four searches converged successfully giving four solutions with a mean *R* factor = 15.74%. Figure 5.27 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.57%.

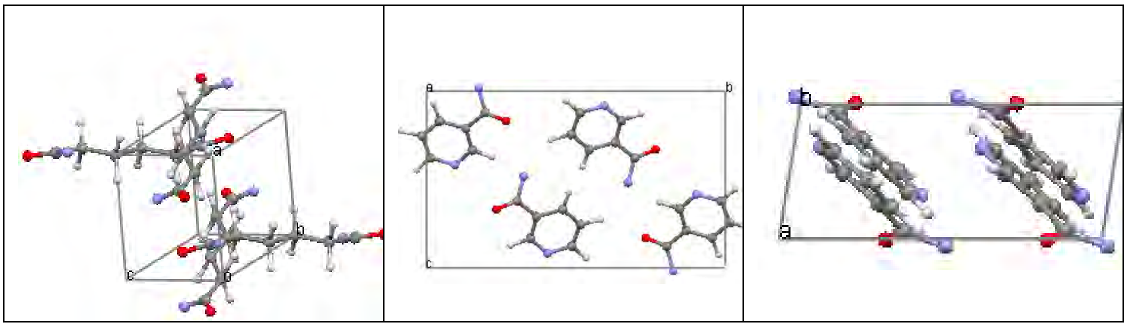


Figure 5.27, The best model structures of adipamide and nicotinamide simultaneously located by elitist direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide and the two right hand views show the solution for nicotinamide. The left view shows that although the conformation of the carbon chain is correct, the orientation and position of the solution is wrong. It is unlikely that this solution for adipamide would refine successfully. The two right hand views show that the solution for nicotinamide has adopted the wrong *anti* conformation. The solution is in the correct orientation and has undergone a crystallographic translation. However, as in fig 5.13, due to crystal symmetry this translation shown in fig 5.27 is equivalent with the published structure shown in fig 5.6. It is likely that this solution for nicotinamide would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.10 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Figure 5.27 demonstrates that although models in each population have been evaluated with two different partner models from the other population, the ‘double DE’ with elitist strategy has failed to solve the structure of adipamide. However, table 5.10 shows that as in table 5.9, the abundance of the wrong solution for adipamide is decreased and the abundance of the correct solution for nicotinamide is increased.

Table 5.10, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	Failed to Converge		
2	15.65%	0.064	0.936
3	16.09%	0.065	0.935
4	15.57%	0.064	0.936
5	15.64%	0.065	0.935
Mean	15.74%	0.065	0.936

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.065*(144.16+122.13))/144.16 = 0.12mol.
Molar ratio nicotinamide = (0.936*(144.16+122.13))/122.13 = 2.04mol.

A second set of five calculations was run to investigate the effect of increasing the value of *EP* from 2 to 8. Thus more children produced by each population were evaluated with an elitist standard. Five elitist DE searches were assigned the control parameters *NP* = 600, *F* = 0.3, *EP* = 8, *K* = 0.99, Gmax = 10,000. Five searches converged successfully giving five solutions with a mean *R* factor = 15.65%. Figure 5.28 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.58%.

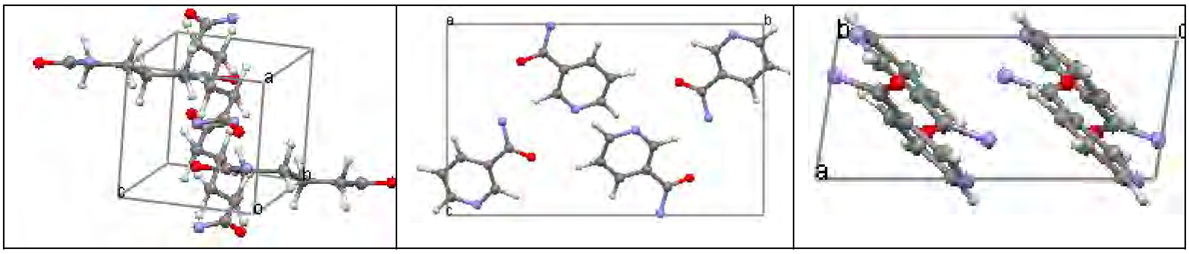


Figure 5.28, The best model structures of adipamide and nicotinamide simultaneously located by elitist direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide and the two right hand views show the solution for nicotinamide. The left view shows that as in the two previous figures, the solution for adipamide is wrong. This solution is unlikely to refine successfully. The two right hand views show that the solution for nicotinamide is in the correct orientation but the solution has adopted the wrong *anti* conformation and is not in the correct position. However, it is likely that these errors could be resolved and that this solution would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.11 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.11, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	15.61%	0.065	0.935
2	15.67%	0.066	0.934
3	15.65%	0.067	0.933
4	15.75%	0.062	0.938
5	15.58%	0.063	0.937
Mean	15.65%	0.065	0.935.

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.065*(144.16+122.13))/144.16 = 0.12mol.
Molar ratio nicotinamide = (0.935*(144.16+122.13))/122.13 = 2.04mol.

Table 5.11 shows that as in tables 5.9 and 5.10, the abundance of the wrong solution for adipamide is decreased and the abundance of the correct solution for nicotinamide is increased.

A third set of five calculations was run to investigate the effect of increasing the value of *EP* from 8 to 10. Five elitist DE searches were assigned the control parameters *NP* = 600, *F* = 0.3, *EP* = 10, *K* = 0.99, Gmax = 10,000. Four searches converged successfully giving four solutions with a mean R factor = 16.40%. Figure 5.29 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.89%.

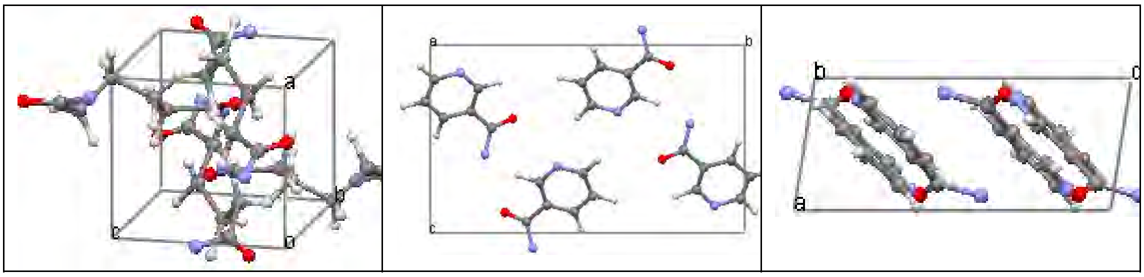


Figure 5.29, The best model structures of adipamide and nicotinamide simultaneously located by elitist direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide, the two right hand views show the solution for nicotinamide. The left view shows that the solution for adipamide cannot be refined; the conformation, orientation and position are all wrong. The two right hand views show that the solution for nicotinamide has adopted the wrong *anti* conformation but the orientation and position are correct. It is likely that this solution would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.12 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.12, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	16.58%	0.025	0.975
2	15.89%	0.062	0.938
3	16.55%	0.026	0.974
4	Failed to Converge		
5	16.57%	0.026	0.974.
Mean	16.40%	0.035	0.965

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.035*(144.16+122.13))/144.16 = 0.065mol.
Molar ratio nicotinamide = (0.965*(144.16+122.13))/122.13 = 2.10mol.

Table 5.12 shows that as in the previous tables the abundance of the wrong solution for adipamide is decreased and the abundance of the correct solution for nicotinamide is increased.

Comparing the convergence rate of the traditional type ‘double DE’ that evolves both populations of structural models simultaneously for all generations (shown in figure 5.30) with the convergence rate of the elitist type ‘double DE’ (presented in figures 5.31-5.33) clearly shows that use of the elitist strategy slows the convergence rate.

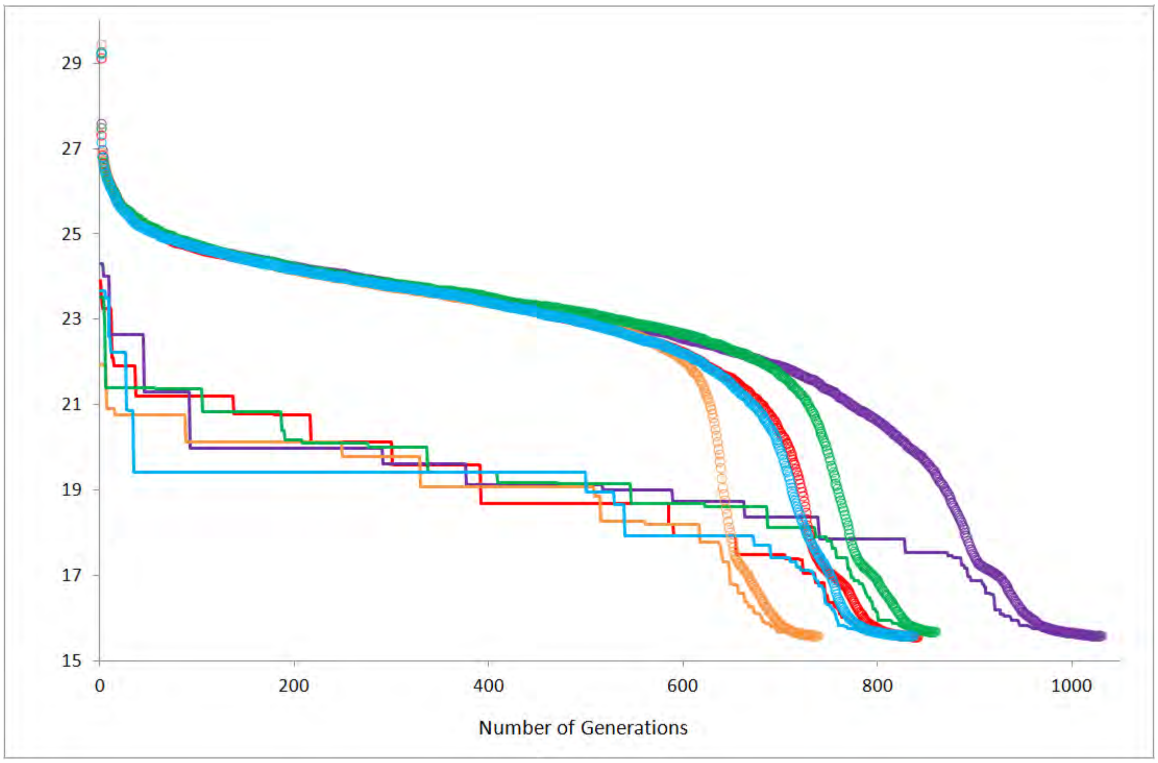


Figure 5.30, The convergence rate of the traditional type ‘double DE’ for five runs. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 861 generations.

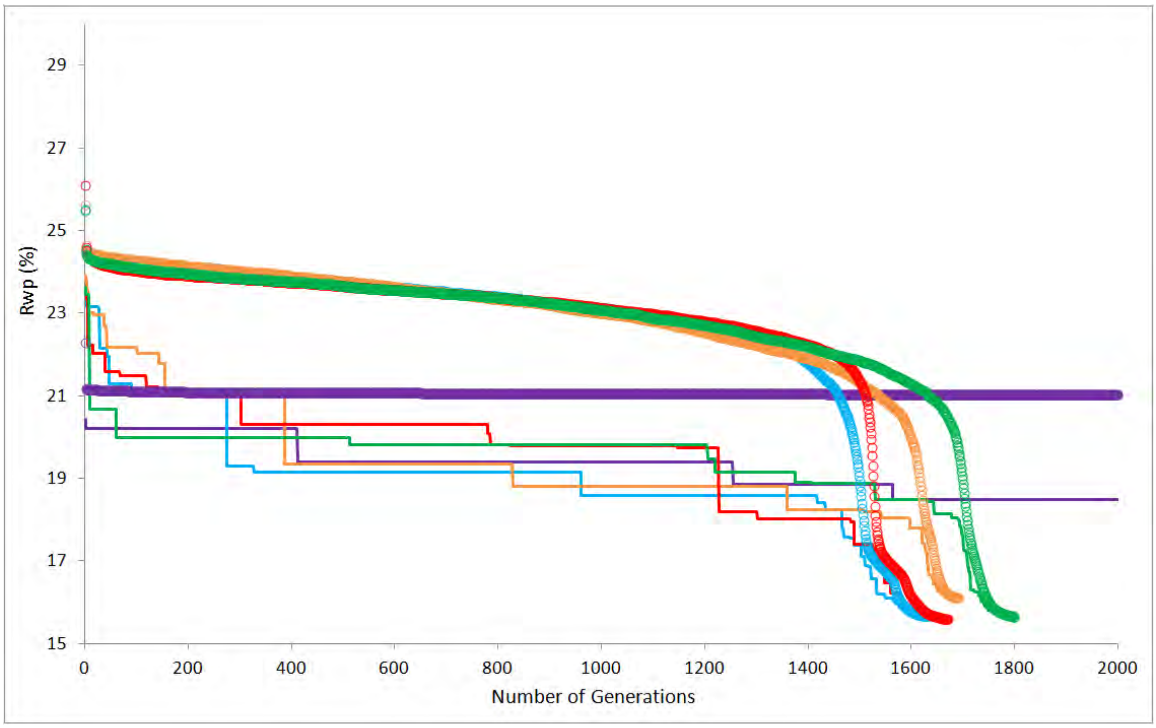


Figure 5.31, The convergence rate of the elitist type ‘double DE’ for five runs with *EP* = 2. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 1698 generations.

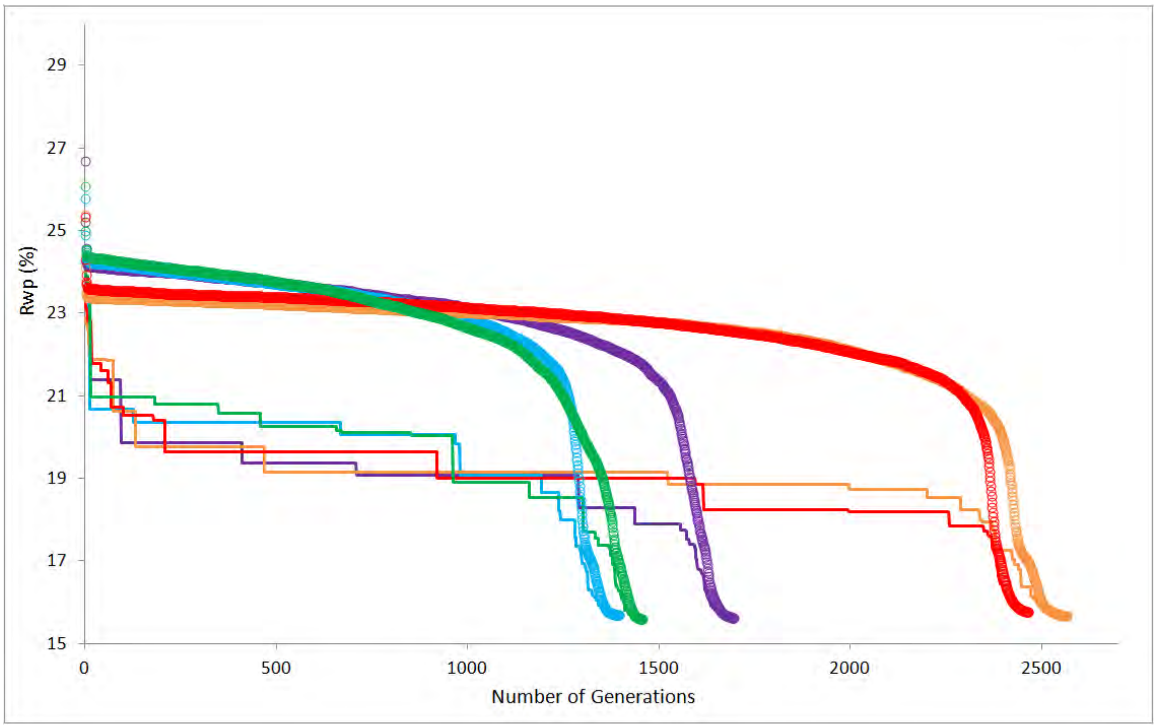


Figure 5.32, The convergence rate of the elitist type ‘double DE’ for five runs with *EP* = 8. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 1916 generations.

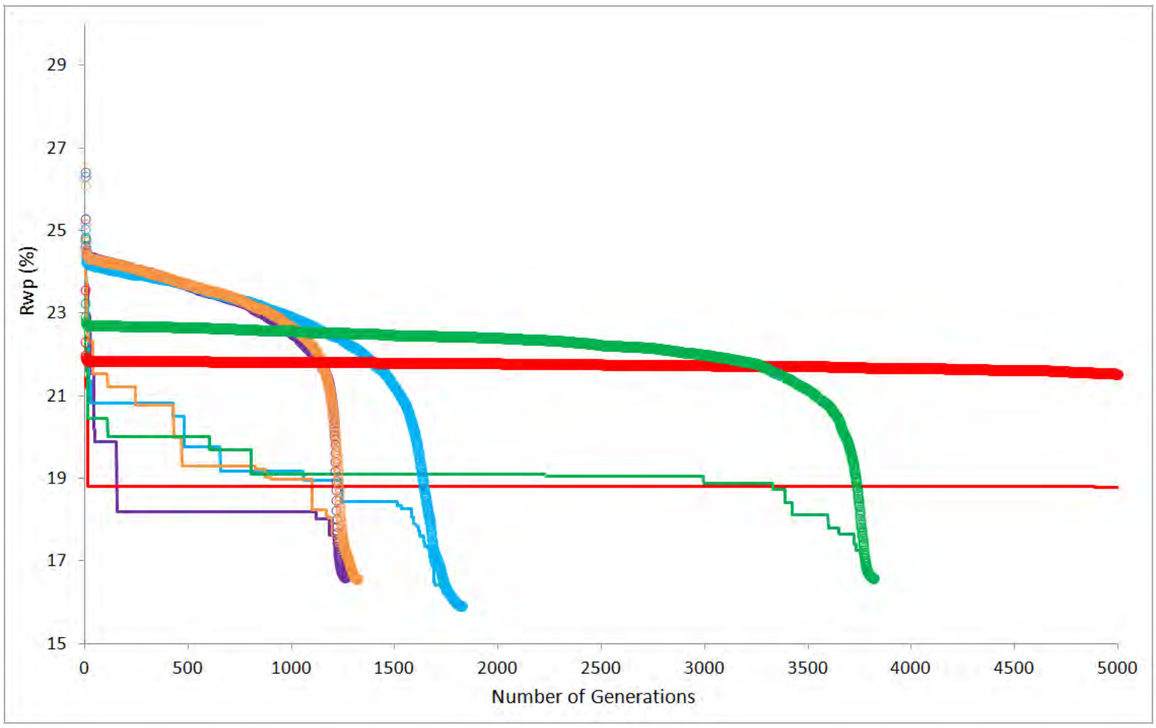


Figure 5.33, The convergence rate of the elitist type ‘double DE’ for five runs with *EP* = 10. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 2057 generations.

The plots shown in figures 5.31-5.33 suggest that as the value of *EP* is increased, the average number of generations required for convergence increases. A possible explanation for the decreased search efficiency of elitist DE searches is that use of the elitist strategy increases the probability that models become trapped in local minima.

The elitist strategy selects ‘elite model A’ from the initial best pair of models. The lower the value of the *R* factor assigned to the best pair, the higher the probability that ‘elite model A’ occupies a minimum. The children produced by parents in population ‘B’ will be assigned an *R* factor with a lower value if the pattern simulated for the child and ‘elite model A’ is a good fit with the experimental biphasic pattern. Thus if ‘elite model A’ occupies a local minimum, evolution of population ‘B’ increases the probability that children are ‘biased’ and only assigned an *R* factor with a low value if paired with a model in population ‘A’ that occupies this particular minimum. If many biased children replace the parents ‘B’, it increases the probability that a significant number of models in population ‘B’ become biased. Thus when the elitist strategy selects ‘elite model B’, there is a high probability that a biased model is selected. Evolution of population ‘A’ and evaluating children with a biased ‘elite model B’ increases the probability that parents ‘A’ are replaced by children occupying the minimum located by ‘elite model A’. Thus the elitist strategy causes models in each population to cluster, but not necessarily near the global minimum. Since models are evaluated in pairs, a pair of models is more likely to be assigned an *R* factor with a relatively low value if the models are located in the clusters. This ‘feedback loop’ between the populations reduces the probability that clustered models explore the landscape, and locate the global minimum. Increasing the value of *EP* increases the number of models clustered together. Therefore, increasing the value of *EP* increases the number of generations required for the cluster to disperse and the models to locate the global minimum.

5.6.3 Systematic selection strategy

An alternative ‘systematic’ strategy was developed to select a ‘standard’ model to represent each population. This strategy selects a model regardless of the *R* factor assigned to the model.

In this strategy, two populations (labelled ‘A’ and ‘B’) of models representing different crystal structures are generated. Each initial model in population ‘A’ is paired with a different initial model in population ‘B’ and evaluated. Model (1) in population ‘A’ is copied and stored as ‘standard model A’. Population ‘B’ is evolved for one generation and each child is evaluated with ‘standard model A’. If the *R* factor assigned to the child and ‘standard model A’ has a lower value than the *R* factor assigned to the parent ‘B’ paired with its previous partner ‘A’, the child replaces the parent ‘B’. After one generation model (2) is copied from population ‘A’ and stored as the new ‘elite model A’. This cycle iterates for a number of generations defined by the value of the parameter standard pair ‘SP’. After ‘SP’ generations, evolution of population ‘B’ is paused. Model (1) is copied from population ‘B’ and stored as ‘standard model B’. The procedure is then reversed. For one generation, population ‘A’ evolves and children are evaluated with ‘standard model B’. After one generation, model (2) is copied from population ‘B’ and stored as the new ‘standard model B’. This cycle iterates for ‘SP’ generations. After population ‘A’ has evolved for ‘SP’ generations, both populations evolve simultaneously. A child produced by a parent ‘A’ is evaluated with a child produced by a parent ‘B’. Both children replace the parents if the *R* factor assigned to the pair of children has a lower value than the *R* factor assigned to the pair of parents. Simultaneous evolution of both populations continues until convergence or until Gmax generations have been calculated. The value of ‘SP’ is defined by the user before the structure solution calculation.

Since a standard model is selected regardless of *R* factor, a standard model has an equal probability of occupying any part of the landscape. This reduces the probability that children evaluated with a standard model are only assigned an *R* factor with a relatively low value if paired with a standard model occupying a minimum. Additionally, since a new standard model is selected each generation, there is a high probability that children are evaluated with standard models occupying different parts of the landscape. This reduces the probability that many children become biased.

Five DE searches using the systematic strategy and quantitative phase analysis were used to simultaneously solve the structures of adipamide and nicotinamide using the biphasic pattern recorded for sample 6. The searches were assigned the DE control parameters *NP* = 600, *F* = 0.3, *SP* = 2, *K* = 0.99, Gmax = 10,000. The five searches converged successfully giving five solutions with a mean *R* factor = 15.72%. Figure 5.34 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.53%.

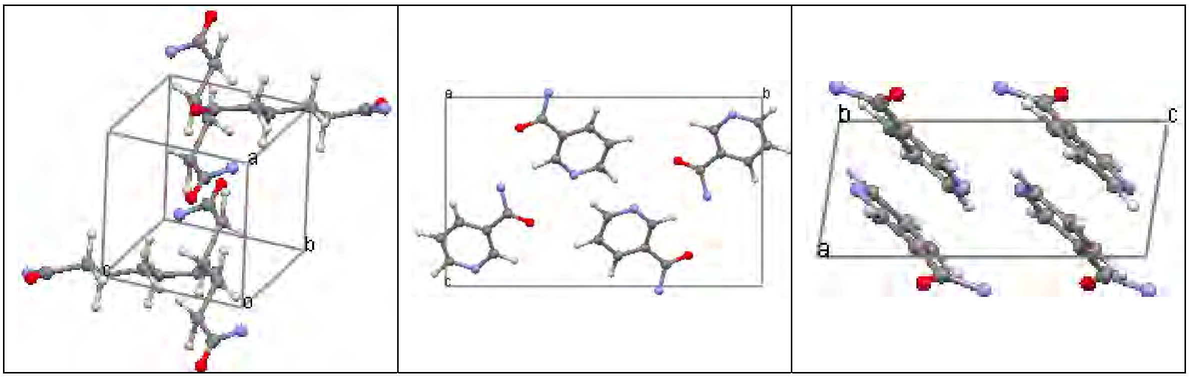


Figure 5.34, The best model structures of adipamide and nicotinamide simultaneously located by systematic direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide, the two right hand views show the solution for nicotinamide. The left hand view shows that the solution for adipamide cannot be refined. The two right hand views show that the solution for nicotinamide is in the correct orientation. However, the solution has adopted the wrong *anti* conformation and is in the wrong position. It is not possible to say if this solution could be refined successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.13 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1. Table 5.13 shows that as in the previous tables the abundance of the wrong solution for adipamide is decreased and the abundance of the correct solution for nicotinamide increased.

Table 5.13, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	16.17%	0.049	0.951
2	15.55%	0.064	0.936.
3	15.67%	0.065	0.935
4	15.66%	0.064	0.936
5	15.53%	0.063	0.937
Mean	15.72%	0.061	0.939.

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.061*(144.16+122.13))/144.16 = 0.11mol.
Molar ratio nicotinamide = (0.939*(144.16+122.13))/122.13 = 2.05mol.

A second set of five calculations was run to investigate the effect of increasing the value of ‘SP’ from 2 to 8. Five systematic DE searches were assigned the control parameters *NP* = 600, *F* = 0.3, *SP* = 8, *K* = 0.99, Gmax = 10,000. The five searches converged successfully giving five solutions with a mean *R* factor = 15.66%. Figure 5.35 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.56%.

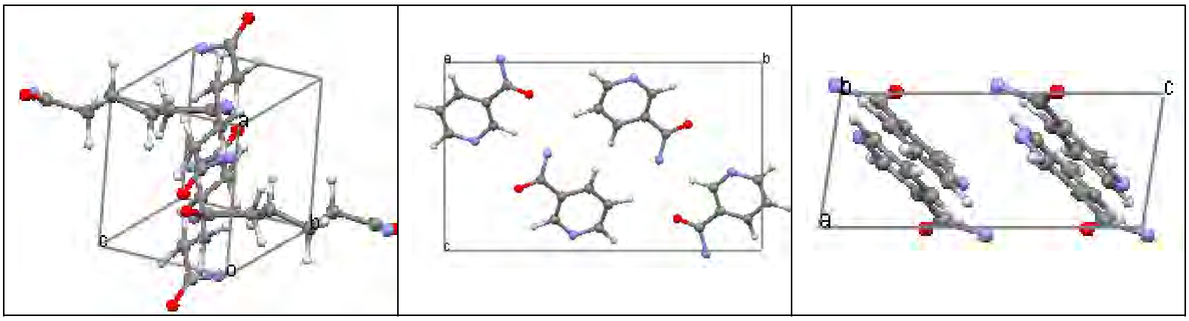


Figure 5.35, The best model structures of adipamide and nicotinamide simultaneously located by systematic direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide, the two right hand views show the solution for nicotinamide. The left hand view shows that the solution for adipamide cannot be refined. The two right hand views show that the solution for nicotinamide is in the correct orientation but has adopted the wrong *anti* conformation. The solution has shifted position but due to symmetry, this position is crystallographically equivalent with the position shown in the published structure (fig 5.6). It is likely that this solution would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.14 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1. Table 5.14 shows that as in the previous tables the abundance of the wrong solution for adipamide is decreased and the abundance for the correct solution for nicotinamide is increased.

Table 5.14, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis for sample 6

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	16.01%	0.059	0.941
2	15.56%	0.064	0.936.
3	15.56%	0.064	0.936
4	15.59%	0.065	0.935
5	15.60%	0.065	0.935.
Mean	15.66%	0.063	0.937

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.063*(144.16+122.13))/144.16 = 0.12mol.
Molar ratio nicotinamide = (0.937*(144.16+122.13))/122.13 = 2.04mol.

Figures 5.36 and 5.37 show the convergence rate of DE searches using the systematic selection strategy.

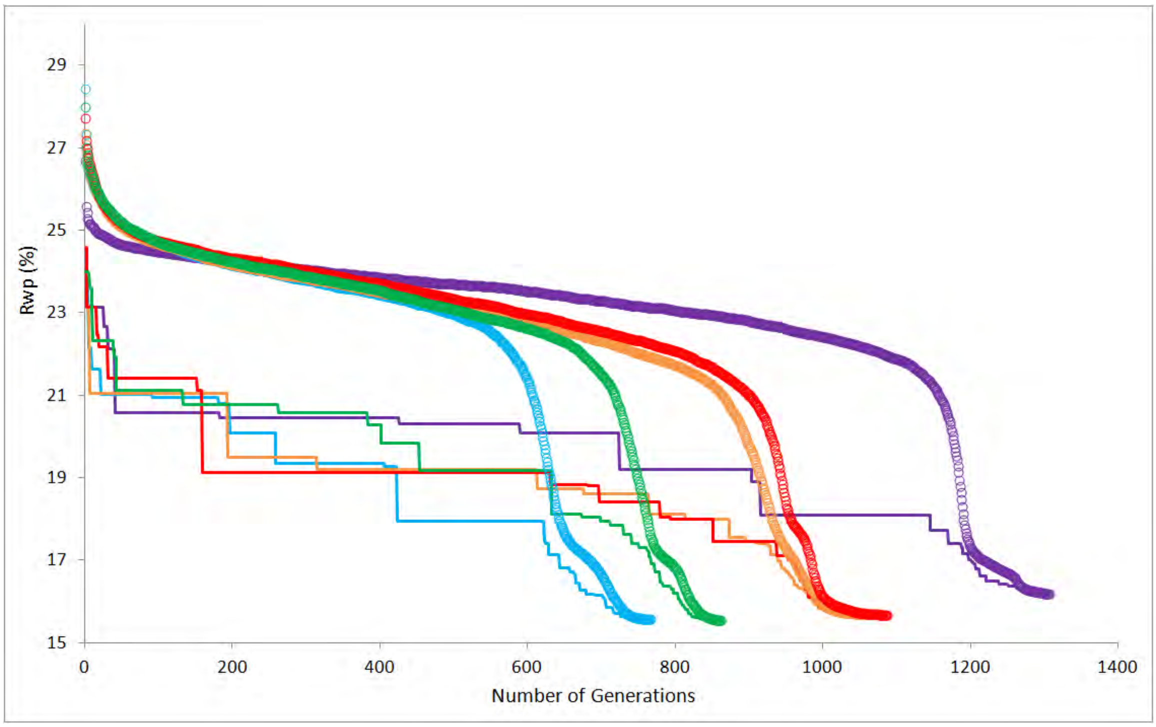


Figure 5.36, The convergence rate of the DE using systematic selection with *SP* = 2. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 1015 generations.

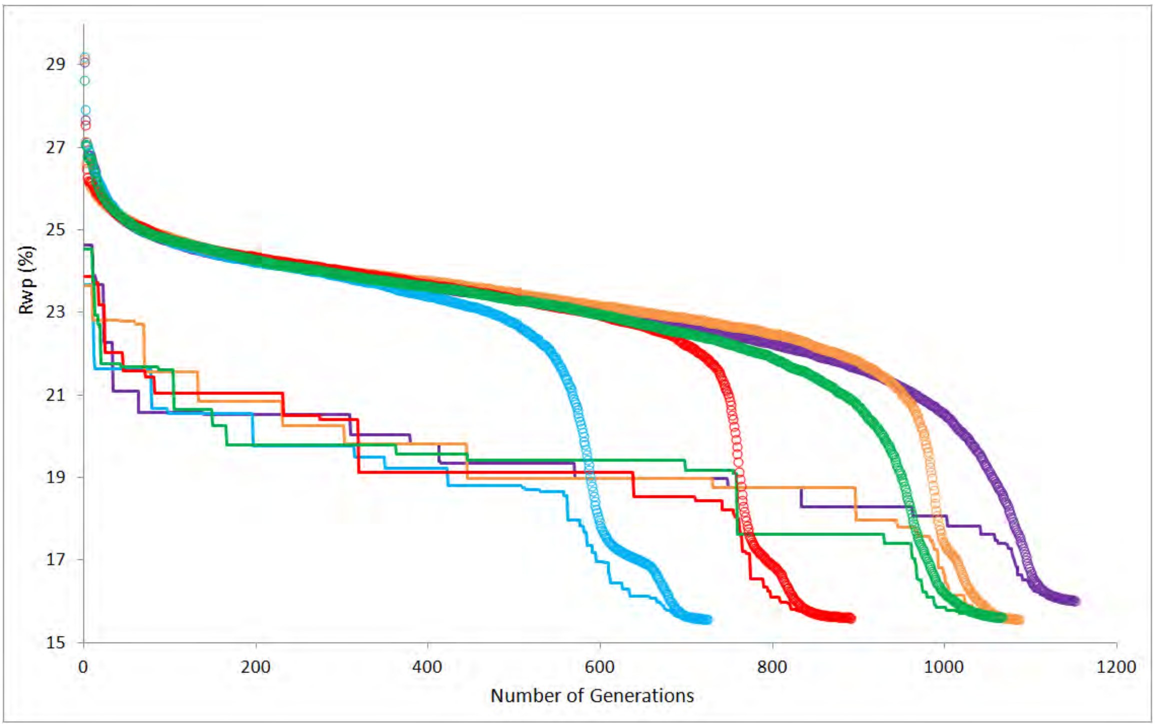


Figure 5.37, The convergence rate of the DE using systematic selection with *SP* = 8. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 984 generations.

Comparison of figures 5.36 and 5.37 with figures 5.31-5.33 demonstrates that the searches using the systematic selection converge faster than the searches using the elitist selection strategy. From the limited results presented in figures 5.36 and 5.37, it is not possible to conclude whether using different values of SP to control the systematic selection strategy has a significant effect on the convergence rate.

5.6.4 Random selection strategy

To determine why searches using the elitist selection strategy require more generations to converge, a third selection strategy was investigated. This strategy selects a standard model from each population regardless of the *R* factor. However, a new standard model is not selected each generation.

In this implementation, two populations (labelled ‘A’ and ‘B’) of models representing different crystal structures are generated. Each initial model in population ‘A’ is paired with a different initial model in population ‘B’ and evaluated. Model (1) in population ‘A’ is copied and stored as ‘standard model A’. Population ‘B’ evolves for a number of generations defined by the value of the parameter fixed standard pair ‘FSP’. Each child produced by a parent ‘B’ is evaluated with ‘standard model A’. If the *R* factor assigned to the child and ‘standard model A’ has a lower value than the *R* factor assigned to the parent B paired with its previous partner ‘A’, the child replaces the parent ‘B’. After ‘FSP’ generations, evolution of population ‘B’ is paused and model (1) is copied from population ‘B’ and stored as ‘standard model B’. The procedure is then reversed. For ‘FSP’ generations, population ‘A’ evolves and the children are evaluated with the ‘standard model B’. After population ‘A’ has evolved for ‘FSP’ generations, both populations evolve simultaneously. A child produced by a parent ‘A’ is evaluated with a child produced by a parent ‘B’. Both children replace the parents if the *R* factor assigned to the pair of children has a lower value than the *R* factor assigned to the pair of parents. Simultaneous evolution of both populations continues until convergence or until Gmax generations have been calculated. The value of ‘FSP’ is defined by the user before the structure solution calculation.

Since a standard model is selected regardless of the *R* factor, a standard model has an equal probability of occupying any part of the landscape. This reduces the probability that children evaluated with a standard model are only assigned an *R* factor with a relatively low value if paired with a model that occupies a minimum. However, since the standard model is not changed after each successive generation, all children produced by one population are paired and evaluated with one standard model for multiple generations. This increases the probability that many biased children are produced in each population and cluster in a small area of the landscape. Thus compared with the elitist strategy, the random strategy decreases the probability that children cluster in local minima, but it does not decrease the probability that children cluster in one area of the landscape.

Five DE searches using the random strategy and quantitative phase analysis were used to simultaneously solve the structures of adipamide and nicotinamide using the biphasic pattern recorded for sample 6. The searches were assigned the control parameters *NP* = 600, *F* = 0.3, *FSP* = 8, *K* = 0.99, Gmax = 10,000. The five searches converged successfully giving five solutions with a mean *R* factor = 15.91%. Figure 5.38 shows three views of the solution that was assigned an *R* factor with the lowest value of 15.55%.

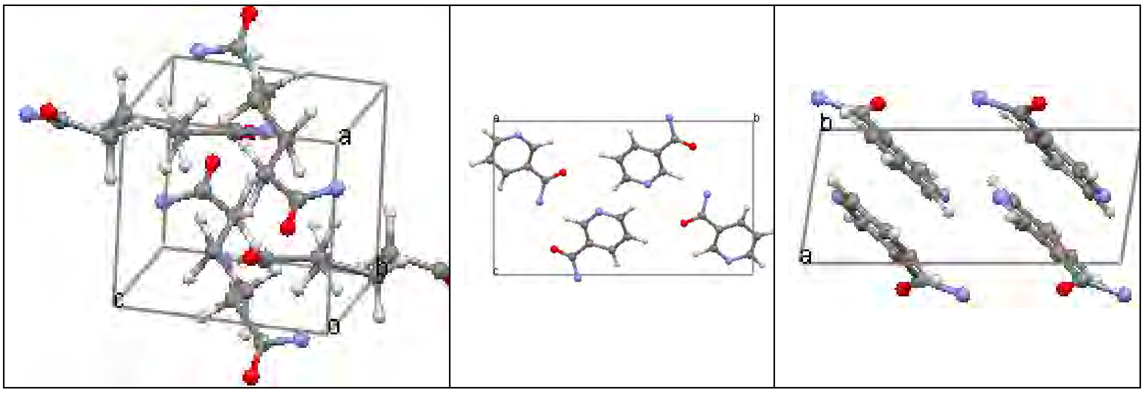


Figure 5.38, The best model structures of adipamide and nicotinamide simultaneously located by random direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 6.

The left hand view shows the solution for adipamide and the two right hand views show the solution for nicotinamide. The left view shows that the solution is in the wrong conformation, position and orientation. It is unlikely that this solution would refine successfully. The two right hand views show that the solution for nicotinamide has adopted the wrong *anti* conformation and the model is not exactly in the correct position. However, the solution is in the correct orientation and it is likely that this solution would refine successfully.

Determining the accuracy of the quantitative phase analysis.

Table 5.15 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and nicotinamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.15, Simultaneous structure solution of adipamide and nicotinamide and quantitative phase analysis. for sample 6.

Run	Converged best *R*	AdipMs*R*	NicMs*R*
1	15.63%	0.065	0.935
2	17.06%	0.027	0.973
3	15.72%	0.063	0.937
4	15.55%	0.064	0.936
5	15.57%	0.064	0.936
Mean	15.91%	0.057	0.943

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.057*(144.16+122.13))/144.16 = 0.11mol.
Molar ratio nicotinamide = (0.943*(144.16+122.13))/122.13 = 2.06mol.

Table 5.15 shows that the abundance of the wrong solution for adipamide is decreased and the correct solution for nicotinamide is increased.

Figure 5.39 shows the convergence rate of these DE searches using the random selection strategy. Comparison of figure 5.39 with figures 5.31-5.33 demonstrates that although the random and elitist selection strategies both select one model from each population to act as the standard model and reuse this standard model for multiple generations, DE searches using the random strategy converge in fewer generations. Although both strategies are equally likely to cause children to cluster in one area of the landscape, the elitist strategy is more likely to cause children to cluster in local minima whereas the random strategy can cause children to cluster in any part of the landscape with equal probability. However, figure 5.30 shows that traditional type DE searches that evolve both populations simultaneously for all generations are still the fastest to converge.

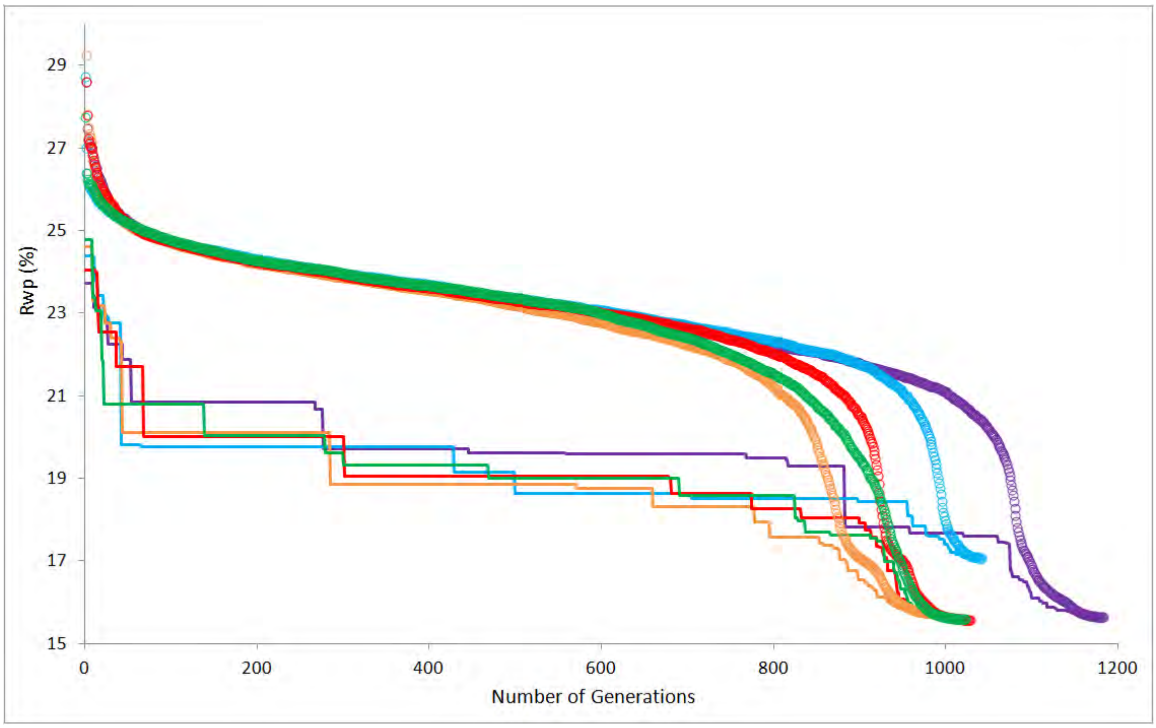


Figure 5.39, The convergence rate of the DE using random selection with *FSP* = 8. Circles indicate the mean fitness of the population, whereas the line shows the evolution of the fittest within each generation. On average, searches converge in 1052 generations.

The four different implementations of the ‘double DE’ investigated here can locate models that are reasonable representations of the real crystal structure of nicotinamide but fail to solve the structure of adipamide. As a consequence, the QPA improves the fit between simulated and real biphasic patterns by significantly reducing the abundance of adipamide and eliminating peaks corresponding to adipamide from the biphasic pattern. Hence the fit between simulated and real biphasic patterns is almost entirely influenced by the quality of the model for nicotinamide. Although the QPA has failed to determine the 1:1 ratio in which sample 6 was prepared, the elimination of peaks corresponding to adipamide and the successful solution for nicotinamide demonstrates that the success of solving one crystal structure is not dependent on the success of solving the other.

A possible reason for the successful solution for nicotinamide but failed solution for adipamide is preferred orientation. The triclinic form of adipamide tends to crystallise in significantly large sheets whereas nicotinamide tends to crystallise in much smaller platelets. Thus the probability that crystallites are packed into the sample holder in one orientation is much higher for adipamide than nicotinamide. This increases the probability that the biphasic patterns recorded for samples 1, 2 and 6 contain significantly more structural information for nicotinamide than adipamide.

Although ‘double DE’ searches using the systematic selection strategy require more generations to converge than the traditional type ‘double DE’ that evolves both populations simultaneously for all generations, searches using the systematic selection strategy are potentially more robust than the traditional type ‘double DE’ searches. This is because in the systematic searches each child is initially paired and evaluated with multiple standard models. This decreases the probability that a good model is assigned an unrepresentatively high *R* factor and missed simply because it is paired and evaluated with a much poorer model.

5.6.5 Biphasic sample 3. Crystalline adipamide and oxamide combined in a molar ratio of 1:1

The DE using the systematic selection strategy was used to solve the crystal structures. This implementation was chosen because it is potentially more robust than the traditional type ‘double DE’.

The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample

3 was used. The value of the phase ratio parameters were each assigned a value of 1. The Le Bail fit was assigned an *R* factor = 16.5%. The crystal structure of adipamide was solved using a model defined by eight parameters and the structure of oxamide by a model defined by six. The combined landscape is therefore defined by a total of 14 parameters. Five systematic DE searches were used to simultaneously solve the structures of adipamide and oxamide. The searches were assigned the control parameters *NP* = 280, *F* = 0.3, *SP* = 8, *K* = 0.99, Gmax = 3000. The five searches converged successfully giving five solutions with a mean *R* factor = 20.74%. Figure 5.40 shows three views of the solution that was assigned an *R* factor with the lowest value of 18.93%.

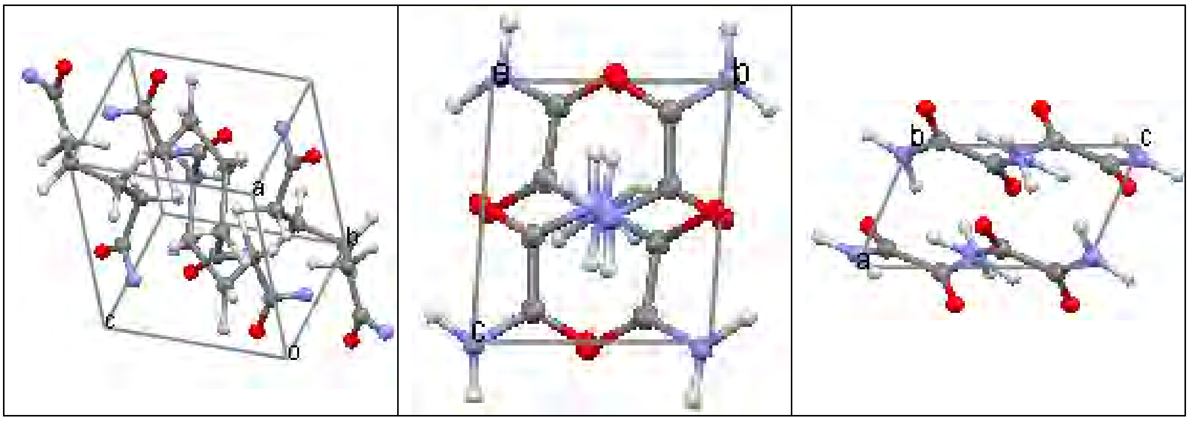


Figure 5.40, The best model structures of adipamide and oxamide simultaneously located by direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 3.

The left hand view shows the solution for adipamide, the two right hand views show the solution for oxamide. The left view shows that the two amide groups are in the correct orientation and that the overall position and orientation of the model is similar to the published structure (fig 5.5). However, the distorted conformation of the carbon chain and the translation of the model from the correct position means that successful refinement of this solution is unlikely. The two right hand views show that the solution for oxamide has adopted the correct planar *trans* conformation. The model is centred on the a plane but is tilted slightly outwards. The model is located sufficiently near the middle of the unit cell for successful refinement.

Determining the accuracy of the quantitative phase analysis.

Sample 3 was prepared by combining adipamide and oxamide in a molar ratio of 1:1 (molar mass of sample 3 = 232.23g/mol^{-1}). Table 5.16 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and oxamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.16, Simultaneous structure solution of adipamide and oxamide and quantitative phase analysis. for sample 3.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	21.57%	0.529	0.471
2	21.34%	0.566	0.434
3	18.93%	0.521	0.479
4	22.39%	0.602	0.398
5	19.45%	0.593	0.407
Mean	20.74%	0.562	0.438.

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.562*(144.16+88.07))/144.16 = 0.905mol.
Molar ratio oxamide = (0.438*(144.16+88.07))/88.07 = 1.155mol.

Although the solution for adipamide shown in fig 5.40 is wrong, table 5.16 shows that the QPA is close to confirming the prepared 1:1 ratio of sample 3. In an attempt to improve the quality of the final solution, the rate of mutation was increased from 0.3 to 0.4. Five systematic DE searches were used to simultaneously solve the structures of adipamide and oxamide. The searches were assigned the control parameters *NP* = 280, *F* = 0.4, *SP* = 8, *K* = 0.99, Gmax = 3000. Four searches converged successfully giving four solutions with a mean *R* factor = 18.97%. Figure 5.41 shows three views of the solution that was assigned an *R* factor with the lowest value of 18.63%.

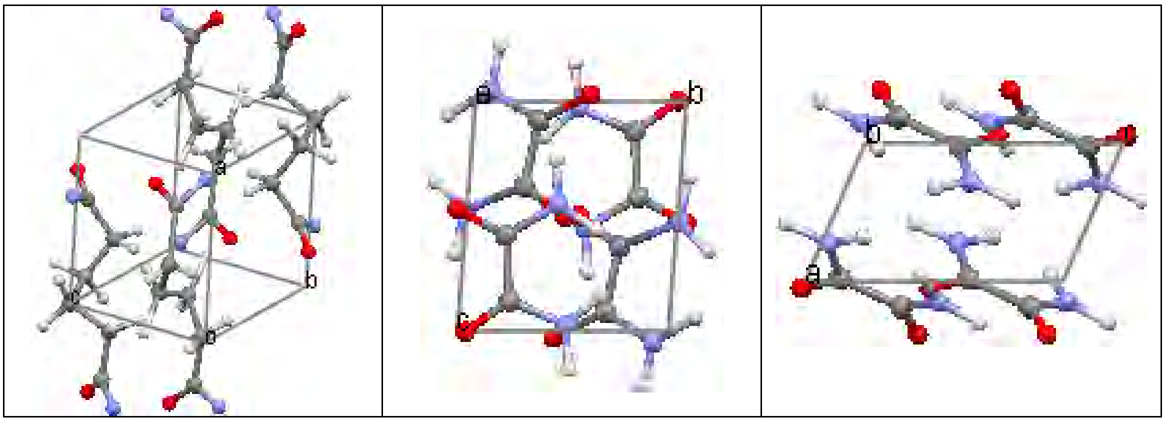


Figure 5.41, The best model structures of adipamide and oxamide simultaneously located by direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 3.

The left hand view shows the solution for adipamide, the two right hand views show the solution for oxamide. The left view shows that the two amide groups are correctly orientated and the overall position and orientation of the model is reasonable. The conformation of the carbon chain is slightly less distorted than in fig 5.40. It may be possible to successfully refine the solution for adipamide shown in fig 5.41. The two right hand views in fig 5.41 show that the solution for oxamide is planar but has adopted the wrong *cis* conformation. The model is located near the middle of the unit cell and is centred on the a plane. However, the significant tilt of this model means that refinement may not be successful. Hence, in this case, increasing the mutation rate has resulted in a better solution for adipamide but a poorer solution for oxamide.

Determining the accuracy of the quantitative phase analysis.

Table 5.17 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and oxamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.17, Simultaneous structure solution of adipamide and oxamide and quantitative phase analysis for sample 3.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	19.13%	0.594	0.406
2	18.63%	0.553	0.447.
3	18.86%	0.625	0.375.
4	Failed To Converge		
5	19.26%	0.523	0.477
Mean	18.97%	0.574	0.426

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.574*(144.16+88.07))/144.16 = 0.92mol.
Molar ratio oxamide = (0.426*(144.16+88.07))/88.07 = 1.12mol.

Table 5.17 shows that the QPA is close to confirming the prepared 1:1 ratio. However, figures 5.40 and 5.41 show, that in each case, only one crystal structure has been solved successfully. These results suggest that the apparent accuracy of the QPA is merely chance.

5.6.6 Biphasic sample 4. Crystalline adipamide and oxamide combined in a molar ratio of 1:2

The DE using the systematic selection strategy was used to solve the structures. The Le Bail fit generated for the biphasic powder diffraction pattern recorded for sample 4 was used. The value of the phase ratio parameters were each assigned a value of 1. The Le Bail fit was assigned an *R* factor = 15.0%. The crystal structures of adipamide and oxamide were solved using models defined by eight and six parameters respectively. Five searches were used with the control parameters *NP* = 280, *F* = 0.3, *SP* = 8, *K* = 0.99, Gmax = 3000. Five searches converged successfully. The five solutions were assigned a mean *R* factor = 18.41%. Figure 5.42 shows three views of the solution that was assigned an *R* factor with the lowest value of 17.34%.

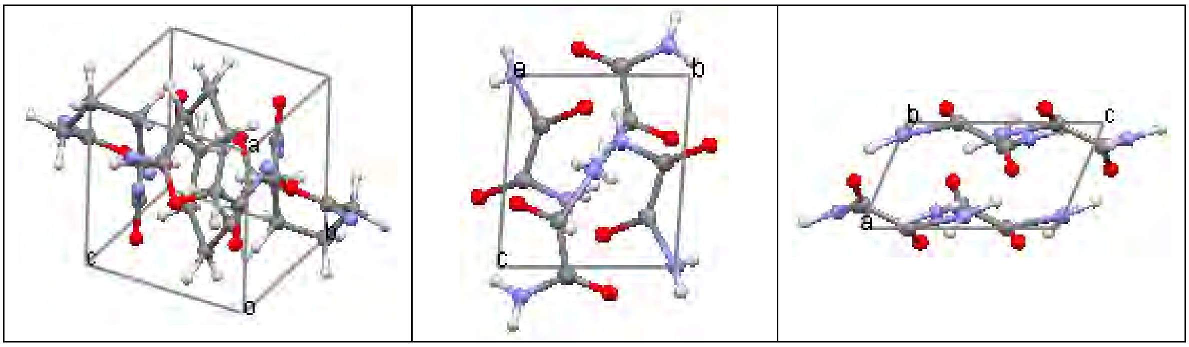


Figure 5.42, The best model structures of adipamide and oxamide simultaneously located by direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 4.

The left hand view shows the solution for adipamide, the two right hand views show the solution for oxamide. The left view shows that the conformation of the carbon chain is significantly distorted. This solution for adipamide will not refine. The two right hand views show that intramolecular geminal bond lengths and angles are distorted, the N-H bond is unusually long. However, the model has adopted the correct *trans* planar conformation and is in the correct position and orientation. Once the obvious intramolecular distortion is manually corrected successful refinement of this model is likely.

Determining the accuracy of the quantitative phase analysis.

Sample 4 was prepared by combining adipamide and oxamide in a molar ratio of 1:2 (molar mass of sample 4 = 320.3 g/mol^{-1}).

Table 5.18 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and oxamide. Data for each structure solution calculation is presented as in table 5.1. Table 5.18 shows that during these searches the QPA has increased the abundance of the wrong adipamide solution and decreased the abundance of the better oxamide solution.

Table 5.18, Simultaneous structure solution of adipamide and oxamide and quantitative phase analysis for sample 4.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	18.86%	0.477	0.523
2	18.26%	0.600	0.400
3	17.34%	0.593	0.407.
4	19.08%	0.550	0.450.
5	18.51%	0.667	0.334
Mean	18.41%	0.577	0.423.

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.577*320.3)/144.16 = 1.3mol.
Molar ratio oxamide = (0.423*320.3)/88.07 = 1.5mol.

In an attempt to improve the quality of final solutions, the rate of mutation was increased from 0.3 to 0.4. Five systematic DE searches were used to simultaneously solve the structures of adipamide and oxamide. The searches were assigned the control parameters *NP* = 280, *F* = 0.4, *SP* = 8, *K* = 0.99, Gmax = 3000. Four searches converged successfully giving four solutions with a mean *R* factor = 17.92%. Figure 5.43 shows three views of the solution that was assigned an *R* factor with the lowest value of 17.47%.

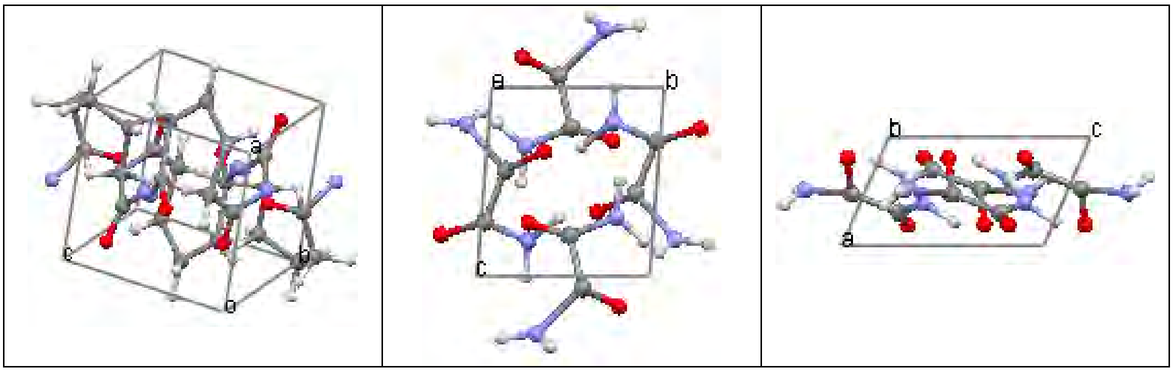


Figure 5.43, The best model structures of adipamide and oxamide simultaneously located by direct space structure solution and quantitative phase analysis from the biphasic powder pattern recorded for sample 4.

The left hand view shows the solution for adipamide, the two right hand views show the solution for oxamide. The left view shows that the conformation of the carbon chain is significantly distorted. Successful refinement of this solution for adipamide is unlikely. The two right hand solutions show a solution for oxamide similar to the oxamide solution 5.42. Here the N-H bond is significantly stretched. However, once the obvious intramolecular distortions are manually corrected refinement is likely to be successful.

Determining the accuracy of the quantitative phase analysis.

Table 5.19 shows the abundance of each phase (presented as a mass fraction) as determined by the Rietveld based QPA during simultaneous structure solution of adipamide and oxamide. Data for each structure solution calculation is presented as in table 5.1.

Table 5.19, Simultaneous structure solution of adipamide and oxamide and quantitative phase analysis for sample 4.

Run	Converged best *R*	AdipMs*R*	OxamMs*R*
1	17.78%	0.561	0.439
2	18.19%	0.563	0.437
3	Failed to Converge		
4	17.47%	0.546	0.454
5	17.95%	0.601	0.391
Mean	17.92%	0.568	0.430

Molar ratios calculated from mass ratios.
Molar ratio adipamide = (0.568*320.3)/144.16 = 1.3mol.
Molar ratio oxamide = (0.430*320.3)/88.07 = 1.6mol.

Table 5.19 shows that during these searches the QPA has increased the abundance of the wrong solution for adipamide and decreased the abundance of the better solution for oxamide.

5.6.7 Conclusions

Section 5.5.2 demonstrates that if a biphasic powder pattern is recorded and the structure of one crystal is already known, together with the abundance of each of the two crystal phases, the direct space method is capable of solving the structure of the other crystal. However, section 5.5.3 demonstrates a case where the same technique fails to solve either structure.

Section 5.5.7 demonstrates that if a biphasic pattern is recorded and the structure of one crystal is known but the abundance of each phase is not, it is possible to perform simultaneous structure solution and quantitative phase analysis and solve the structure of the unknown crystal. However, section 5.5.6 demonstrates a case where this same technique fails to solve the crystal. Section 5.5.7 also demonstrates that the QPA does not accurately determine the abundance of each phase even when the structure solution is successful.

Figure 5.41 and table 5.17 in section 5.6.5 suggest that it is possible to perform QPA whilst simultaneously solving both crystal structures from a biphasic pattern. However, all other attempts to achieve this have failed, so the apparent success may only be chance. Clearly, these limited successes do not prove that simultaneous two-structure solution and QPA is possible. Significantly more work is needed to test and develop the simultaneous two-structure solution and QPA technique.

Most of these attempts to solve the crystal structure of nicotinamide have located solutions that adopt the wrong *anti* conformation between the nitrogen of the amide group and the nitrogen of the ring. Since nitrogen and oxygen have very similar X-ray scattering powers, a 180° rotation of the amide group will have little effect on the *R* factor assigned to a solution. Thus it is difficult for our direct space technique to determine the correct orientation of an amide group. Although it is trivial to determine the correct orientation during Rietveld refinement by manually flipping the amide group, the fact that most of these structure solution searches find the wrong *anti* conformation demonstrates a bias in our technique. As discussed in section 2.1.2, the POSSUM package uses standard bond lengths to generate structural models. The standard C-O bond used is 1.45Å and the standard C-N bond = 1.55Å. Since this work was carried out, DE searches have been conducted for nicotinamide using different standard bond lengths. It has been discovered that using a slightly shorter standard C-N bond and a slightly longer standard C-O bond increases the probability that DE searches locate the correct orientation of the amide group.

5.6.8 Further work

A potentially more successful way to perform simultaneous direct space crystal structure solution and QPA is to use the DE to optimise both the structural models and parameters defining the abundance of each phase. This could be done by assigning one additional chromosome to each model to define the phase mass fraction for that model. Thus for this hypothetical technique, when a biphasic pattern is simulated for a pair of models the values of the mass fraction parameters of each model are used to calculate each phase specific scale factor and Rietveld refinement would only be used to compare the simulated and real biphasic patterns. In this technique the determination of the phase abundance would become part of the optimisation problem.

The mass fraction and structure parameters could be optimised either: (a) simultaneously for all generations or (b) sequentially. For example, the structure parameters could be optimised until a certain quality of fit was achieved between simulated and real biphasic patterns when the mass fraction parameters would also be included in the search.

References.

[1] M. L. Cheney, N. Shan, E. R. Healey, M. Hanna, L. Wojtas, M. J. Zaworotko, V. Sava, S. Song and J. R. Sanchez-Ramos. Effects of Crystal Form on Solubility and Pharmacokinetics, A Crystal Engineering Case Study of Lamotrigine. *Cryst. Growth. Des*. (2010). **10**. 394.

[2] Z. Hugonin, M. Johnsson and S. Lidin. Two for the Price of One, Resolvable Polymorphism in a Single Crystal of Sb3O4I. *Sol. Stat. Sci*. (2009) **11**. 24.

[3] R. E. Dinnebier, F. Olbrich, S. van Smaalen and P. W. Stephens. Ab Initio Structure Determination of Two Polymorphs of Cyclopentadienylrubidium in a Single Powder Pattern. *Acta. Cryst. B.* (1997). **53**. 153.

[4] T. R. Shattock, P. Vishweshwar, Z. Wang and M. J. Zaworotko. 18-Fold Interpenetration and Concomitant Polymorphism in the 2,3 Co-Crystal of Trimesic Acid and 1,2-Bis(4-pyridyl)ethane. *Cryst. Growth. Des*. (2005). **5**. 2046.

[5] A. V. Trask, J. van de Streek, W. D. S. Motherwell and W. Jones. Achieving Polymorphic and Stoichiometric Diversity in Cocrystal Formation, Importance of Solid-state Grinding, Powder X-ray Structure Determination, and Seeding. *Cryst. Growth. Des*. (2005). **5**. 2233.

[6] J. A. His, P. Vishweshwar, R. A. Middleton and M. J. Zaworotko. Concomitant and Conformational Polymorphism, Conformational Isomorphism, and Phase Relationships in 4-Cyanopyridine'4,4'-biphenol Cocrystals. *Cryst. Growth. Des*. (2006). **6**. 1048.

[7] K. F. Bowes, G. Ferguson, A. J. Lough and C. Glidewell. The 1,1 Adduct of triphenylsilanol and 4,4-bipyridyl and Three Pairwise-concomitant Triclinic Polymorphs of the 4,1 Adduct Having Z' = 0.5, 1 and 4. *Acta. Cryst. B.* (2003). **59**. 277.

[8] J. A. His, P. Vishweshwar, D. R. Weyna and M. J. Zaworotko. Hierarchy of Supramolecular Synthons, Persistent Hydroxyl Pyridine Hydrogen Bonds in Cocrystals That Contain a Cyano Acceptor. *Mol. Pharm*. (2007). **4**. 401.

[9] A. V. Trask, N. Shan, W. D. S. Motherwell, W. Jones, S. Feng, R. B. H. Tan and K. J. Carpenter. Solvent-drop Grinding, Green Polymorph Control of Cocrystallisation. *Chem. Commun*. (2005). 880.

[10] D. R. Weyna, T. Shattock, P. Vishweshwar and M. J. Zaworotko. Synthesis and Structural Characterization of Cocrystals and Pharmaceutical Cocrystals, Mechanochemistry vs Slow Evaporation from Solution. *Cryst. Growth. Des*. (2009). **9**. 1106.

[11] J. Bernstein, R. J. Davey and J. O. Henck. Concomitant Polymorphs. *Angew. Chem. Int. Ed*. (1999). **38**. 3440.

[12] E. Maccaroni, G. B. Giovenzana, G. Palmisano, D. Botta, P. Volante and N. Masciocchi. Structures from Powders, Diflorasone diacetate. *Steroids*. (2009). **74**. 102.

[13] S. L. Wang, P. C. Wang and Y. P. Nieh. Structure Determination of LiMoP2O7 from Multiphase Powder X-ray Diffraction Data. *J. Appl. Cryst*. (1990). **23**. 520.

[14] P. J. Bendall, A. N. Fitch and B. E. F. Fender. The Structure of Na_2UCl_6 and Li_2UCl_6 from Multiphase Powder Neutron Profile Refinement. *J. Appl. Cryst*. (1983). **16**. 164.

[15] M. L. Cheney, M. J. Zaworotko, S. Beaton and R. D. Singer. Cocrystal Controlled Solid-State Synthesis. A Green Chemistry Experiment for Undergraduate Organic Chemistry. *J. Chem. Ed*. (2008). **85**. 1649.

[16] N. Shan, F. Toda and W. Jones. Mechanochemistry and co-crystal formation, effect of solvent on reaction kinetics. *Chem. Commun*. (2002). 2372.

[17] W. H. Zachariasen and F. H. Ellinger. The Crystal Structure of Beta Plutonium Metal. *Acta. Cryst. A*. (1963). **16**. 369.

[18] K. Shankland, W. I. F. David and D. S. Sivia. Routine ab initio Structure Determination of Chlorothiazide by X-ray Powder Diffraction using Optimised Data Collection and Analysis Strategies. *J. Mater. Chem*. (1997). **7**. 569.

[19] M. Brunelli, J. P. Wright, G. B. M. Vaughan, A. J. Nora and A. N. Fitch. Solving Larger Molecular Crystal Structures from Powder Diffraction Data by Exploiting Anisotropic Thermal Expansion. *Angew. Chem. Int. Ed*. (2003). **42**. 2029.

[20] P. Fernandes, K. Shankland, W. I. F. David, A. J. Markvardsen, A. J. Florence, N. Shankland and C. K. Leechh. A Differential Thermal Expansion Approach to Crystal Structure Determination from Powder Diffraction Data. *J. Appl. Cryst*. (2008). **41**. 1089.

[21] B. M. Kariuki, S. A. Belmonte, M. I. McMahon, R. L. Johnston, K. D. M. Harris and R. J. Nelmes. A New Approach for Indexing Powder Diffraction Data Based on Whole-profile Fitting and Global Optimization using a Genetic Algorithm. *J. Synchrotron Rad*. (1999). **6**. 87.

[22] N. A. S. Webster, I. C. Madsen, M. J. Loan, R. B. Knott, F. Naim, K. S. Wallwork and J. A. Kimpton. An Investigation of Goethite-Seeded $Al[OH]_3$ Precipitation using *In Situ* X-ray Diffraction and Rietveld-Based Quantitative Phase Analysis. *J. Appl. Cryst*. (2010). **43**. 466.

[23] L. Leon-Reina, A. G. De la Torre, J. M. Porras-Vazquez, M. Cruz, L. M. Ordonez, X. Alcobe, F. Gispert-Guirado, A. Larranaga-Varga, M. Paul, T. Fuellmann, R. Schmidt and M. A. G. Arand. Round Robin on Rietveld Quantitative Phase Analysis of Portland Cements. *J. Appl. Cryst*. (2009). **42**. 906.

[24] N. V. Y. Scarlett, I. C. Madsen, C. Manias and D. Retallack. On-Line X-ray Diffraction for Quantitative Phase Analysis: Application in the Portland Cement Industry. *Powder Diff*. (2001). **16**. 71.

[25] F. Guirado and S. Gali. Quantitative Rietveld Analysis of CAC Clinker Phases using Synchrotron Radiation. *Ceme. Conc. Res*. (2006). **36**. 2021.

[26] A. W. Hull. A New Method of Chemical Analysis. *J. Am. Chem. Soc*. (1919). **41**. 1168.

[27] R. J. Hill and C. J. Howard. Quantitative Phase Analysis from Neutron Powder Diffraction Data using the Rietveld Method. *J. Appl. Cryst*. (1987). **20**. 467.

[28] D. L. Bish and S. A. Howard. Quantitative Phase Analysis using the Rietveld Method. *J. Appl. Cryst.* (1988). **21**. 86.

[29] v. Esteve, L. E. Ochando, M. M. Reventos, G. Peris and X. M. Amigo. Quantitative Phase Analysis of Mixtures of Three Components using Rietveld and Rius Standardless Methods: Comparative Results. *Cryst. Res. Tech*. (2000). **35**. 1183.

[30] N. V. Y. Scarlett, I. C. Madsen, M. I. Pownceby and A. N. Christensen[b]· In situ X-ray Diffraction Analysis of Iron Ore Sinter Phases. *J. Appl. Cryst*. (2004). **37**. 362.

[31] L. P. Aldridge. Accuracy and Precision of Phase Analysis in Portland Cement by Bogue, Microscopic and X-ray Diffraction Methods. *Ceme. Conc. Res*. (1982). **12**. 381.

[32] X. Orlhac, C. Fillet, P. Deniard, A. M. Dulac and R. Brec· Determination of the Crystallized Fractions of a Largely Amorphous MultiPhase Material by the Rietveld Method. *J. Appl. Cryst*. (2001). **34**. 114.

[33] I. C. Madsen, N. V. Y. Scarlett, L. M. D. Cranswick and T. Lwin. Outcomes of the International Union of Crystallography Commission on Powder Diffraction Round Robin on Quantitative Phase Analysis: Samples 1a to 1h. *J. Appl. Cryst*. (2001). **34**. 409.

[34] N. V. Y. Scarlett, I. C. Madsen, L. M. D. Cranswick, T. Lwin, E. Groleau, G. Stephenson, M. Aylmore and N. AgronOlshin. Outcomes of the International Union of Crystallography Commission on Powder Diffraction Round Robin on Quantitative Phase Analysis: samples 2, 3, 4, synthetic bauxite, natural granodiorite and pharmaceuticals. *J. Appl. Cryst*. (2002). **35**. 383.

[35] R. J. Hill. International Union of Crystallography Commission on Powder Diffraction Rietveld Refinement Round Robin. I. Analysis of Standard X-ray and Neutron Data for $PbSO_4$. *J. Appl. Cryst*. (1992). **25**. 589.

[36] R. J. Hill and L. M. D. Cranswick. International Union of Crystallography Commission on Powder Diffraction Rietveld Refinement Round Robin. II. Analysis of Monoclinic ZrO_2 . *J. Appl. Cryst*. (1994). **27**. 802.

[37] H. M. Rietveld. A Profile Refinement Method for Nuclear and Magnetic Structures. *J. Appl. Cryst.* (1969). **2**. 65.

[38] K. Kachrimanis, M. Rontogianni and S. Malamataris. Simultaneous Quantitative Analysis of Mebendazole Polymorphs A-C in Powder Mixtures by DRIFTS Spectroscopy and ANN Modeling. *J. Pharm. Biomed. Anal*. (2010). **51**. 512.

[39] C. C. Seaton and M. Tremayne. Differential Evolution, Crystal Structure Determination of a Triclinic Polymorph of Adipamide from Powder Diffraction Data. *Chem. Commun*. (2002). 880.

[40] Y. Miwa, T. Mizuno, K. Tsuchida, T. Taga and Y. Iwata. Experimental Charge Density and Electrostatic Potential in Nicotinamide. *Acta. Cryst. B*. (1999). **55**. 78.

[41] G. de With and S. Harkema. Structure and Charge Distribution of Oxamide as Determined from High-Order X-ray Data. *Acta. Cryst. B*. (1977). **33**. 2367.

[42] C. C. Seaton and M. Tremayne, POSSUM. Programs for Direct-Space Structure Solution from Powder Diffraction Data. School of Chemistry. University of Birmingham UK. (2002).

[43] A. C. Larson and R. B. Von Dreele. GSAS Generalized Structure Analysis System. Manual LAUR 86-748. Los Alamos National Laboratory. Los Alamos. NM. USA. (1986).

[44] ChemOffice Pro 2010. CambridgeSoft, 1 Signet Court, Swanns Road, Cambridge, CB5 8LA.

6 Conclusions and Further Work

6.1 Optimising DE based direct space crystal structure solution

Although results presented in chapter 3 demonstrate that it may be possible to predict how changing the value of a specific DE control parameter may affect the progress of a structure solution calculation, the results demonstrate that there is no optimal combination of DE control parameters ‘universal’ to all crystal structures.

Table 3.1 shows that the success rate of searches for the structure of baicalein using the same mutation rate generally increases as the population size increases. Since the *R* factor landscape representing the crystal structure of baicalein contains relatively few local minima, increasing the population size decreases the probability that a significant number of models cluster in one local minimum causing premature convergence. Increasing the mutation rate decreases the convergence rate but does not significantly decrease the probability that a search converges prematurely.

Searches using larger population sizes and small mutation rates converge in fewer generations than searches using smaller populations and larger mutation rates. Table 3.1 shows that searches with NP = 280 and F = 0.2 converge with 100% success in the least generations, but searches with NP = 105 and F = 0.3 converge in the least time as the smaller population size means that fewer child fitness evaluations are calculated.

Table 3.4 shows that the success rate of searches for the structure of adipamide generally increases as the mutation rate increases, but the success rate is not significantly affected by the population size. Since the *R* factor landscape representing the crystal structure of adipamide contains numerous local minima, it is likely that many models explore many local minima before locating the global minimum. Increasing the mutation rate increases the probability that models escape local minima and locate the global minimum. Increasing the population size increases the proportion of models in a population that explore local minima before converging, thus searches using smaller populations are not significantly more likely to converge prematurely.

Comparing tables 3.1 and 3.4 shows that although the crystal structure of adipamide is represented by a landscape with greater dimensionality and more local minima than the landscape for baicalein, searches with smaller population sizes are significantly more successful at solving the structure of adipamide. The quickest searches for adipamide with 100% success use *NP* = 56 and *F* = 0.6. The quickest searches for baicalein with 100% success use *NP* = 105 and *F* = 0.3.

Table 3.7 shows that searches using larger populations and mutation rates for the structure of acetarsone are generally more successful. This suggests that the landscape representing the structure of acetarsone contains sufficient local minima to trap models and cause premature convergence. Acetarsone contains an arsenic atom (a relatively strong X-ray scatterer) and certain local minima may represent near optimal positions for the arsenic atom but a non-optimal position and orientation for the rest of the molecule. Thus models that have optimised the position of the arsenic atom can be trapped in a local minimum. Increasing the population size reduces the proportion of models that locate one local minimum and increasing the mutation rate increases the probability that models escape local minima and converge in the global minimum.

Crystal structures that are likely to be represented by *R* factor landscapes containing relatively few local minima are more likely to be solved efficiently by DE searches using moderate population sizes and small mutation rates. However, structures represented by *R* factor landscapes containing many local minima are likely to be most efficiently solved with larger mutation rate and possibly large population size.

6.2 Optimising cultural DE

Tables 3.3 and 3.6 demonstrate that cultural DE searches can converge in fewer generations than the analogous static DE. Increasing the value of *NUT* generally decreases the number of generations required for convergence but if *NUT* is too large it can reduce the success rate compared with analogous DE searches. Searches using larger population sizes or larger mutation rates can support more pruning. However, the complex relationship between *NP*, *F* and *NUT* means that much experimentation may be required to discover the optimal combination for a particular crystal structure. Since it is unlikely an optimal combination is chosen for a new crystal structure, a cultural search is unlikely to be significantly more efficient or successful than a static DE search when used to solve a new structure.

Table 3.9 demonstrates that cultural searches are not suitable for solving some crystal structures. If the best model frequently leads the population by a significant distance, the area occupied by the best model can be pruned from the search space. Tables 3.11a-b demonstrate that preventing a search from pruning the area of landscape occupied by the best model can increase the success rate of cultural searches.

Changing how the clustering of models is measured, can affect the convergence and success rate of cultural searches. Table 3.11c shows that pbCDE searches are more successful at solving the structure of acetarsone than oCDE and cbCDE searches (tables 3.11a-b). However pbCDE searches that control the position of the landscape boundaries by consideration of the accepted children and unbeaten parents are generally slower to converge than oCDE and cbCDE searches using analogous *NP*, *F* and *NUT* combinations.

Auspicious cultural pruning

Tables 3.12 and 3.13 demonstrate that searches used to solve different crystal structures develop clustering behaviour at different rates. If pruning is initiated too early, models may be encouraged to remain in local minima, increasing the probability that searches converge prematurely and increasing the number of generations required for successful convergence. However, if pruning is initiated too late, many models are already clustered near the global minimum and restricting the search space does not guide many models towards the global minimum and increase the convergence rate compared with an analogous static DE. Therefore to increase the efficiency of cultural DE, pruning should be initiated by criteria that indicate the clustering of models near the global minimum: pruning should not be initiated after an arbitrary number of generations. In a similar way to the eugenic DE, cultural pruning could be initiated when a certain proportion of the models in a population are assigned an *R* factor with a relatively low value.

6.3 Eugenic DE

The results presented in section 4.4 demonstrate that eugenic DE is more efficient than static DE. The robust nature of eugenic DE is demonstrated by the successful solution of crystal structures of considerably different complexity using the same combination of control parameters in less time than DE searches. This means that when the technique is used to solve a new crystal structure, eugenic DE using a general combination of control parameters is likely to be more efficient than static DE and it is not necessary for the user to have a detailed understanding of the landscape representing the crystal structure to solve the structure quickly. This is in contrast with the cultural DE where the speed of convergence and success rate of searches is more dependent on the combination of control parameters used for the particular structure.

Since the eugenic and accelerator searches prune a population when a significant number of models are near the global minimum, these two search techniques are potentially more robust than FixedGenPrune searches that prune a population after an arbitrary number of generations (regardless of how many models are near the global minimum). Table 4.11 shows the solution of baicalein by FixedGenPrune searches. Table 4.11 shows that for searches with primary NP = 280, delaying pruning generally increases the efficiency of searches. However for searches with primary NP = 560, it is not clear that delaying pruning increases efficiency. Similar results are presented in table 4.12 which shows the structure solution of adipamide by FixedGenPrune searches with primary NP = 320 and 640.

6.4 Crystal structure solution from Biphasic Powder diffraction data

Section 5.5.2 demonstrates that if a biphasic powder diffraction pattern is recorded for a biphasic sample containing one known and one unknown crystal phase, providing the abundance and the lattice parameters of each crystal are known, it is possible to use the direct space method to solve the unknown crystal directly from the biphasic pattern without subtracting peaks corresponding to the known structure from the biphasic pattern. This is a potentially useful technique if a significant number of peaks corresponding to each phase are overlapped in the biphasic pattern, since in this case subtraction of peaks corresponding to the known structure is likely to destroy many peaks corresponding to the unknown structure, decreasing the probability that this unknown structure is solved. However, section 5.5.3 demonstrates a similar case where the direct space method is unable to solve a crystal structure from a biphasic pattern.

Section 5.5.7 demonstrates that if a biphasic pattern is recorded and the structure of one crystal is known but the abundance of each crystal is not, it is possible to perform simultaneous structure solution and quantitative phase analysis and solve the structure of the unknown crystal and determine the abundance of each crystal. However, section 5.5.6 demonstrates a case where this same technique fails.

The direct space solution of one crystal structure and the simultaneous determination of the abundance of each crystal is a challenging problem because in this case the structure solution and QPA processes are linked. The structure is solved and the abundance of each phase is determined by quantitatively comparing the fit between simulated and real biphasic patterns. Thus the quality of fit is affected by the quality of the model and the abundance of each phase combined. If a biphasic pattern is simulated for a poor quality model it is unlikely that peaks corresponding to the model will be correctly located or have the correct intensity in the simulated pattern. This increases the probability that the Rietveld refinement based QPA fits the simulated and real biphasic patterns by reducing the abundance of the phase represented by the model. Hence, once the quality of the model has little effect on the quality of fit between simulated and real biphasic patterns it is unlikely that the direct space method locates a model that is a good representation of the real crystal structure.

The simultaneous direct space solution of two crystal structures and Rietveld based quantitative phase analysis from a biphasic powder pattern is clearly an even harder problem. Here, only figure 5.41 and table 5.17 presented in section 5.6.5 suggest that this might be possible.

Figure 5.26 in section 5.6.1 demonstrates that if the direct space technique only locates one model that is a good representation of the real crystal structure, the QPA can fit the simulated and real biphasic patterns by significantly increasing the abundance of this solved structure and decreasing the abundance of the structure represented by the poorer quality model. Decreasing the abundance of the structure represented by the poorer quality model, means that simulated peaks corresponding to this model have little effect on the fit between simulated and real biphasic patterns. Although this demonstrates that successful solution of one structure is not dependent on successful solution of the other, it does suggest that if one structure is solved before the other, the QPA may increase the abundance of the solved structure, preventing successful solution of the other structure.

When simultaneously solving two structures from a biphasic pattern, evaluating models with different partners as in ‘double DE’ searches using the elitist, systematic and random selection strategy may decrease the probability that a model that is a good representation of one real crystal structure is assigned an *R* factor with a high value, and missed because it is paired and evaluated with a model that is a significantly bad representation of the other real crystal structure. It also reduces the probability that the QPA improves the fit between simulated and real biphasic patterns by significantly increasing the abundance of the structure represented by the better model which potentially prevents solution of the second structure. However, the results presented in sections 5.6.2-5.6.4 show that DE searches using the elitist, systematic and random strategies are not more successful or efficient than the traditional type ‘double DE’ (section 5.6.1) that does not evaluate each model with different partners.

Section 5.6.2 shows that the use of an elitist strategy significantly reduces the efficiency of structure solution. Elitist strategies can require twice as many generations to converge than traditional type ‘double DE’ searches.

Comparison of the results of searches using the elitist strategy with the results of searches using the random selection strategy (section 5.6.4) suggests that use of the elitist strategy increases the probability that a significant proportion of the models in one or both populations cluster in local minima and are prevented from exploring the landscape and converging in the global minimum.

Section 5.6.3 demonstrates that searches using the systematic selection strategy require slightly more generations to converge than the traditional type ‘double DE’. However, searches using the systematic selection strategy converge in significantly fewer generations than those using the elitist strategy.

Although no attempt has been made to index the biphasic patterns recorded for the six samples used in this thesis, two pattern decomposition/indexing techniques are identified and discussed in sections 5.2.2 and 5.2.3 as possible alternatives to the established pattern subtraction method 5.2.1. Although one of these techniques (5.2.3) has already been used to index a biphasic powder pattern recorded for a biphasic sample containing a trace impurity phase, further work is needed to test if this technique is capable of indexing a biphasic powder pattern recorded for a sample in which the two phases are equally abundant.

As discussed in section 5.6.8, further work is also needed to investigate the development of an algorithm in which the phase abundance calculation is an integral part of the optimisation process. It is theoretically feasible to use an additional structure parameter to define the abundance of a phase and use the DE algorithm to optimise the phase abundance instead of the Rietveld refinement calculation.

A potential advantage of this hypothetical technique is that it is possible to prevent the DE algorithm from improving the fit between simulated and real biphasic patterns by increasing the abundance of the structure represented by a better quality model. This hypothetical DE algorithm could be programmed to optimise both structural and abundance parameters simultaneously for the first few initial generations of a search, and then to only optimise the structural parameters until the search is close to convergence, when optimisation of the abundance parameters could be continued. During the initial few generations of a search, all models are likely to be equally poor solutions, hence the DE is unlikely to significantly increase the abundance of a relatively good model. However, initial optimisation of the abundance parameters would allow the DE to evaluate and reject significantly incorrect abundance values. Allowing the DE to only optimise the structure parameters until the search is close to convergence, forces the DE to solve both structures simultaneously rather than solving one structure and improving the fit between simulated and real biphasic patterns by significantly increasing the abundance of this better model. Once both models are relatively good representations of the real crystal structures it would then be possible to optimise the abundance parameters without sacrificing one model. However, there is no guarantee that if developed, such an algorithm would be more successful at simultaneous multiple crystal structure solution and QPA than the technique investigated here, using the direct space method to solve the crystal structures and Rietveld refinement based QPA.

6.5 Further work

a) Develop the cultural search so that pruning is initiated once a proportion of the models in the population are near the global minimum rather than after an arbitrary number of generations. One potential way of initiating the cultural pruning is to use the RequireFrac criteria used by the eugenic DE. Once a certain proportion of the models in the population are assigned a relatively low *R* factor, pruning is initiated and the boundaries used to confine the search.

b) Determine what clustering behaviour should be used to control the movement of the boundaries in the cultural search. The ChildBest implementation considers the clustering of all children whereas the PopulationBest considers the clustering of the accepted children and unbeaten parents.

c) Determine whether the accelerator search is more efficient at solving significantly complex crystal structures (such as organic salts and cocrystals) than the eugenic search.

d) Continue to test the technique of simultaneous direct space crystal structure solution and quantitative phase analysis. Searches using the elitist strategy are clearly the least efficient. However, more experiments are needed to determine whether the traditional type ‘double DE’ is the fastest to converge and is equally robust as the searches using the systematic selection strategy, especially when one crystal phase is significantly more abundant.

e) Use the pattern decomposition techniques (discussed in chapter 5 as alternatives to the pattern subtraction method) to index multiphasic powder diffraction patterns. Evaluate the practicalities of recording multiphasic diffraction patterns at different temperatures and use anisotropic thermal expansion of crystal structures to resolve overlapped diffraction peaks. Determine if the GA based indexing algorithm is able to index biphasic patterns recorded for materials in which the two crystal phases are equally abundant and if the technique can index patterns recorded for materials containing more than two crystal phases.

f) Develop a DE algorithm capable of simultaneously solving two crystal structures and determining the abundance of each phase from a biphasic pattern rather than using Rietveld refinement for quantitative phase analysis.

Appendix A: Experimental

A.1 Collection of X-ray data

In this work, powder samples are prepared by placing crystalline material between two pieces of transparent Scotch tape, creating a circular sample area of approximately 0.5-1 cm^2. A Bruker AXS D5000 high resolution powder diffractometer with Ge-monochromated CuKα_1 (λ=1.54056 Å) radiation and a small angle position sensitive detector covering 8° in 2θ is used to record powder diffraction patterns. Data is collected using the D5000 diffractometer in transmission mode under ambient pressure and temperature. All data collection details are given below, except for Sample 6 (adipamide and nicotinamide in a molar ratio of 1:1) for which the data are not shown and the data set was recorded over the range $5^o<2\theta<85^o$ in 0.0202° steps over a period of 15 hours.

A.2 Data processing

Le Bail fits[1] are generated for experimental powder diffraction patterns using the Le Bail fit application of the GSAS[2] package. Direct space crystal structure solution is performed using the differential evolution algorithm written in the Perl language, version 5.0[3], implemented in the Possum package[4]. An initial model that is manually 'drawn' using ChemOffice Chem[5] is used to supply the Possum package with structural information such as molecular connectivity, bond angles and bond lengths. Model structures generated by the DE algorithm are evaluated using the Rietveld refinement[6] application of GSAS to simulate a powder diffraction pattern for each model and using the R_{wp} cost function, quantitatively compare the simulated pattern with the experimental pattern and thus assign an *R* factor to each model. Differential evolution is used to evolve populations of structural models and thus solve the crystal structure. A search is judged to have converged when all models are assigned an *R* factor of the same value. However, a search is automatically terminated if the search fails to converge within *Gmax* generations. Quantitative phase analysis is performed using the Rietveld refinement application of GSAS. Structure solution calculations are performed on individual desktop PCs running SuSe Linux.

References

[1] A. Le Bail, H. Duroy and J. L. Fourquet. Ab-initio Structure Determination of LiSbWO6 by X-ray Powder Diffraction. *Mater. Res. Bull*. (1988). **23**. 447.

[2] A. C. Larson and R. B. Von Dreele. GSAS: Generalized Structure Analysis System. Manual LAUR 86-748. Los Alamos National Laboratory. Los Alamos. USA. (1986).

[3] R. L. Schwartz, T. Phoenix and B. D. foy. Learning Perl, Fourth Edition. O'Reilly Media Inc. California. (2005).

[4] C. C. Seaton and M. Tremayne, POSSUM. Programs for Direct-Space Structure Solution from Powder Diffraction Data. School of Chemistry. University of Birmingham UK. (2002).

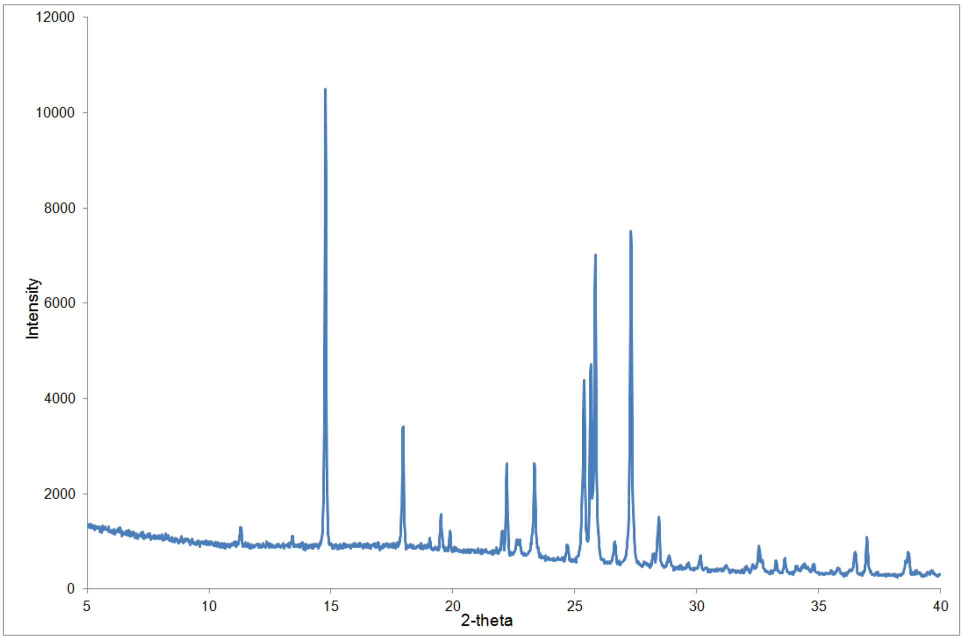
[5] ChemOffice Pro 2010. CambridgeSoft, 1 Signet Court, Swanns Road, Cambridge, CB5 8LA.

[6] H. M. Rietveld. A Profile Refinement Method for Nuclear and Magnetic Structures. *J. Appl. Cryst.* (1969). **2**. 65.

A.3 Experimental Powder Diffraction Profiles

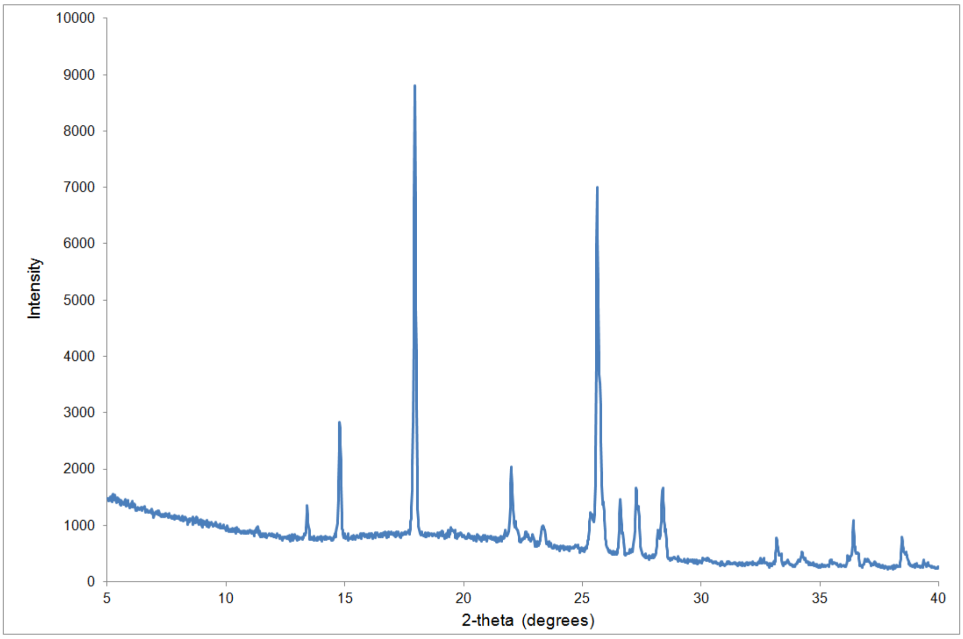
Sample 1. Adipamide and nicotinamide combined in a molar ratio of 1:3

This data set was recorded over the range 5º<2θ<40º in 0.0197º steps over a period of 2 hours.



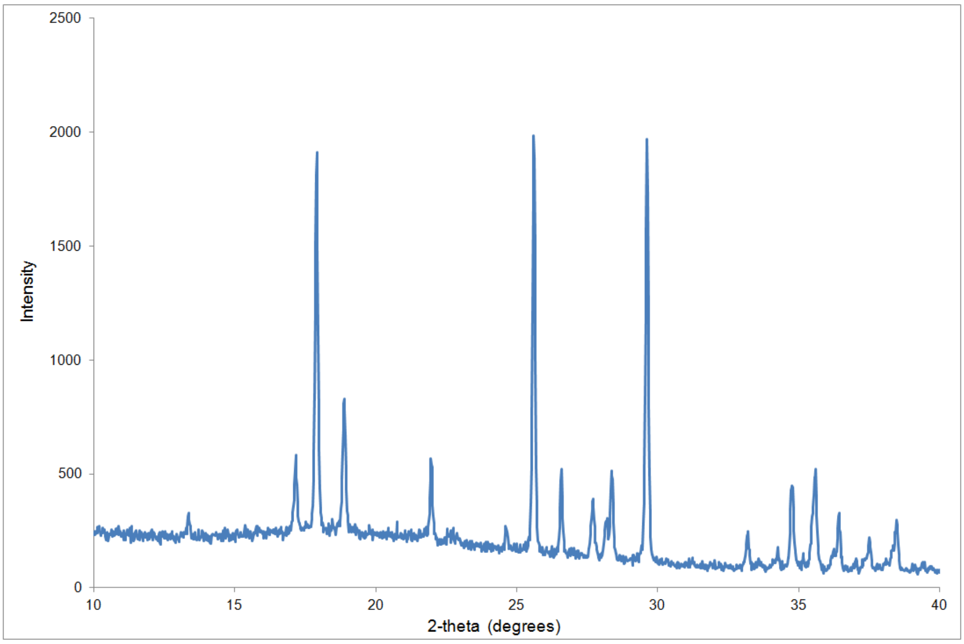
Sample 2. Adipamide and nicotinamide combined in a molar ratio of 3:1

This data set was recorded over the range 5º<2θ<40º in 0.0197º steps over a period of 2 hours.



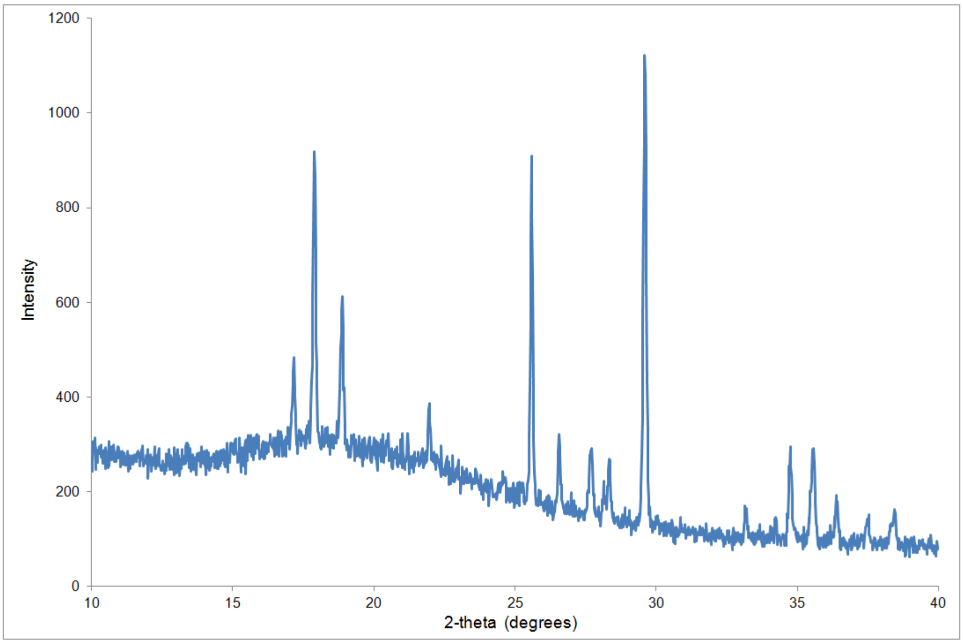
Sample 3. Adipamide and oxamide combined in a molar ratio of 1:1

This data set was recorded over the range 10º<2θ<40º in 0.0197º steps over a period of 1 hour.



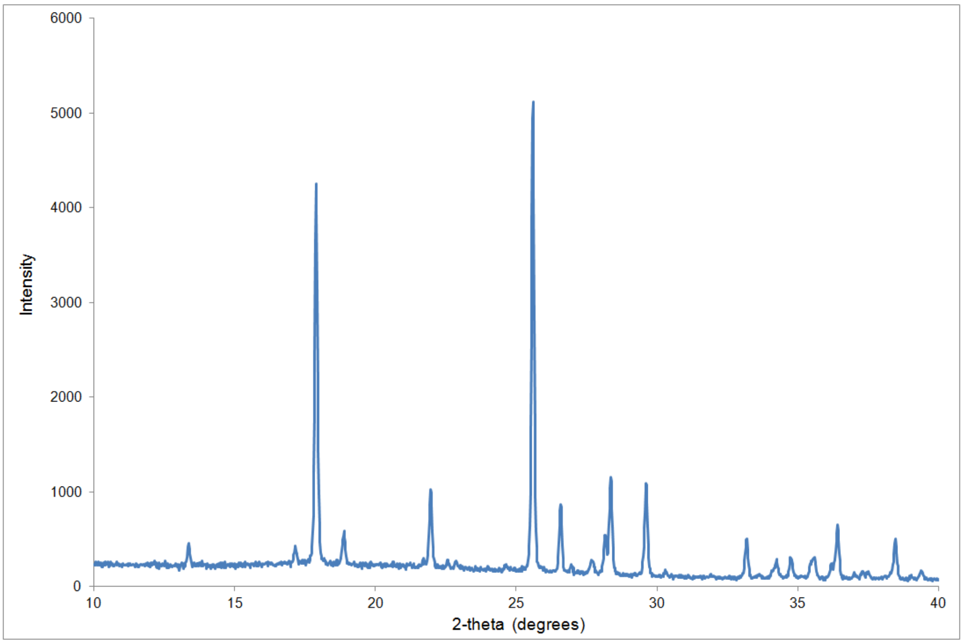
Sample 4. Adipamide and oxamide combined in a molar ratio of 1:2

This data set was recorded over the range 10º<2θ<40º in 0.0197º steps over a period of 1 hour.



Sample 5. Adipamide and oxamide combined in a molar ratio of 2:1

This data set was recorded over the range 10º<2θ<40º in 0.0197º steps over a period of 2 hours.



Appendix B: Eugenic DE Subroutine

```
#! /usr/bin/perl
# This code is for Eugenic DE.
#### Duncan Bell.
#### May 12th 2010.
#### Use this to control the primary population size.
   # The primary population size will be calculated by
   # multiplying the number here by the number of landscape dimensions
$PopSize = 40;
#### This controls what fraction of the secondary population will already have beaten the initial best to initiate pruning
   # example 4 = quarter.
$RequireFrac = 4;
#### This sets the secondary F value
$Secondary_F = 0.6;
#### maximum number of generations allowed before run terminated.
$Gmax = 1000;
$total_no_runs = 10;
#### The recombination rate
$K = 0.99;
my $SET_NP = $PopSize*$D;
my $SET_F = 0.1;
$best_division = ($PopSize/10);
sub differential_evolution
{
  @population_best = ();
  @rank = ();
 initialise_data(\@population,\@atom_vectors,\@atom_names,\
@orthogonal_matrix,\@inverse_matrix,\@SpaceGroup,$total_no_atoms,$NP,
$D,\@fragData,\@no_atom, $no_indep_frag);
   @rank = sort ascend @r;
  my $best_r = @rank[0];
 my $total = 0;
  for (my $i=0; $i<$NP; $i++)
 {
   $total += $r[$i];
   if ($r[$i] == $best_r)
  {
   for (my $j=0; $j<$D; $j++)
{
  $population_best[0][$j] = $population[$i][$j];
}
  }
 }
 $mean_r = $total/$NP;
my $best_r = $r[0];
foreach my $r (@r)
{
if ($r <= $best_r)
{
$best_r = $r;
$best_solution = $pos;
}
$pos++;
}
foreach (0..$D-1)
{
$best[$_] = $population[$best_solution][$_];
}
  save_best_from_initial(\@best,\@atom_vectors,\@atom_names,\
@orthogonal_matrix,\@inverse_matrix,\@SpaceGroup,$total_no_atoms,\
@fragData,\@no_atom, $no_indep_frag);
open (RESULTS, ">>$filename1") or die "Can't access $filename1:$!\n";
 print RESULTS "from the initial population ";
 printf RESULTS ("best r = %.2f ", $best_r);
```

```
  printf RESULTS ("Mean r = %.4f", $mean_r);
  print RESULTS "\n";
  close RESULTS;

#### emptying rank array.
@rank = ();
my $initial_best = $best_r;
$Prune_count = 1;
$no_fes = 0;
$generation = 1;
while ($generation <= $Gmax)
  {
####  Reset Counters for each generation
    $accepted = 0;
    $total = 0;
$Num_Good_Models = 0;
    for (my $i = 0;$i<$NP; $i++)
    {
#### Pick 3 random relatives.
            do { $r1 = int(rand()*($NP-1)); }
         while ($r1 == $i);
            do { $r2 = int(rand()*($NP-1)); }
         while ($r2 == $i or $r2 == $r1);
            do { $r3 = int(rand()*($NP-1)); }
         while ($r3 == $i or $r3 == $r1 or $r3 == $r2);
#### Create child from parent and relatives.
            for (my $j=0;$j<$D;$j++)
            {
         $tmp[$j] =
$population[$i][$j] + $K*($population[$r3][$j] - $population[$i][$j]) +
$F*($population[$r1][$j]-$population[$r2][$j]);
            #### Checking that parameters of child are inside boundaries
         $tmp[$j] =
($population[$i][$j] + $lo[$j])/2 if ($tmp[$j] < $lo[$j]);
        $tmp[$j] =
($population[$i][$j] + $hi[$j])/2 if ($tmp[$j] > $hi[$j]);
           }

#### Child is formed and checked, now evaluate.
           $trial                                                   =
evaluate_cost(\@tmp,\@atom_vectors,\@atom_names,\@orthogonal_matrix,\@inverse_matrix,
\@SpaceGroup,$total_no_atoms,\@fragData,\@no_atom, $no_indep_frag);
#### If child better than parent, replace parent with child.
           if ($trial <= $r[$i])
           {
         $accepted++;
         for (my $j=0;$j<$D;$j++)
         {
                 $population[$i][$j] = $tmp[$j];
         }
         $r[$i] = $trial;
#### If child is better than current population best, replace current best with child.
         if ($trial <= $best_r)
         {
my $count = 0;
foreach (@tmp)
{
$best[$count] = $_;
$count++;
}
                for (my $j=0; $j<$D; $j++)
               {
                      $population_best[0][$j] = $tmp[$j];
               }
               $best_r = $trial;
               copy ("MC.EXP", "BEST_SOL$no_run.EXP");
```

```
            copy ("trial.pdb", "best_sol$no_run.pdb");
       }
     }
     $total += $r[$i];
if ($Prune_count < 1.5)
{
        if ($r[$i] <= $initial_best)
                {
$Num_Good_Models++;
        }
}
    }
$mean_r = $total/$NP;
#### % of children that have replaced their parents during generation
    $percent_acc = $accepted/$NP * 100;
#### total number of children evaluated since start of DE run
$no_fes += $NP;
open (RESULTS, ">>$filename1") or die "Can't open $filename1:$!\n";
print RESULTS "generation $generation, ";
printf RESULTS ("best_r %.2f, ", $best_r,);
printf RESULTS ("mean_r %.4f, ", $mean_r);
printf RESULTS ("%.2f", $percent_acc);
print RESULTS ("% of children accepted.\n");
close (RESULTS);
#### If population has not been pruned yet,
#### determine how many models have R factors as good as initial best.
if (($Prune_count < 1.5) &&
  ($Num_Good_Models >= (($NP/$best_division)/$RequireFrac)))
{
   $Prune_count = 10;
#### Primary population is ready to be pruned.
#### Calculate how many models to transfer into secondary population.
@rank = sort ascend @r;
```

```
my $best_fraction = @rank[($NP/$best_division)-1];
$F = $Secondary_F;
#### Move the selected models into holding array and bin the rest.
@store_division = ();
$split_count = 0;
for (my $i = 0; $i < $NP; $i++)
{
#### If model is good enough to go into secondary population and
#### holding array is not yet full, save this model.
if (($split_count < ($NP/$best_division)) && ($r[$i] <= $best_fraction))
{
$store_division[$split_count] = $r[$i];
for (my $j = 0; $j <$D; $j++)
{
$store_division[$split_count][$j] = $population[$i][$j];
}
$split_count++;
}
#### If model is not good enough for secondary population
#### or holding array is full, bin this model.
else
{
$rank[0] = $r[$i];
for (my $j = 0; $j < $D; $j++)
{
$rank[0][$j] = $population[$i][$j];
}
}
}
#### Prepare the secondary population.
@rank = ();
@r = ();
$NP = $NP/$best_division;
```

```
#### Now move the saved models from holding array into secondary population.
for (my $i = 0; $i < $NP; $i++)
{
$r[$i] = $store_division[$i];
for (my $j = 0; $j <$D; $j++)
{
$population[$i][$j] = $store_division[$i][$j];
}
}
#### Secondary population is created and ready to go!
open (RESULTS, ">>$filename1") or die "Can't open $filename1:$!\n";
print RESULTS "pruned population to $NP structures, F = $F\n";
close RESULTS;
}
$generation++;
#### If mean R very similar to best R terminate the search.
    last if ( ($mean_r - $best_r) < 0.01)
  }
  return ($best_r);
}
```

Appendix C: Coevolution

C.1 The intrinsic inefficiency of one-population algorithms

Our implementation of DE uses one population of individuals each defined by a complete set of parameters (chromosomes) to represent a complete (though not necessarily optimal structural model). The fitness value assigned to an individual reflects how well all the parameters defining the individual combine to produce a complete solution. If one parameter defining an individual reaches its optimal value whilst the remaining parameters remain significantly non-optimal it is likely that the individual is assigned a low fitness value and not identified as a ‘promising’ member of the population. An individual defined by no optimal parameters but fewer significantly non-optimal parameters is likely to be assigned a higher fitness value. Since in DE parent and child compete in a ‘knockout tournament’ this increases the probability that a child with one optimal parameter is rejected and a parent individual with no optimal parameters is retained. This makes the use of just one population in the evolution of complete solutions intrinsically inefficient.

Cooperative coevolutionary algorithms have been implemented using separate populations to optimise different parameters of the same problem.[1-5] Relatively simple mathematical functions have been solved using coevolutionary algorithms.[1,3] In this work the value of each function variable is optimised by a separate population. The separately optimised variables are periodically combined into a ‘whole solution’ and evaluated to assess how well the separate variables solve the function. In Simao *et al* [5], a cooperative coevolutionary genetic algorithm is used to optimise the process schedule of an oil refinery. An oil refinery is a continuous processing system that simultaneously uses different pieces of equipment to process various crude oil fractions and produce a range of products. Due to the number of different tasks involved and by the necessity to use one piece of equipment to process different oil fractions it is a complex optimization problem. In addition, some tasks have precedence constraints and require to be scheduled first. In this work, one population is used to optimise the order in which processes are carried out and a second population is used to optimise the assignment of processes to equipment. Periodically two separate parts are selected from each population and combined to evaluate the proposed refinery schedule.

Some of the evolutionary techniques used by these algorithms could be used to create a cooperative coevolutionary DE applied to direct space crystal structure solution. As each population would only be used to optimise one specific parameter, individuals defined by that optimal parameter would be more conspicuous amongst individuals defined by less optimal parameters. As each parameter reached its optimal value it could be copied and combined with the other (separately optimised) parameters to produce a complete fully optimised solution.

Parameters defining a model used in direct space crystal structure solution have been determined sequentially (thus independently) using the Patterson method,[6,7] by attributing significantly intense peaks in a diffraction pattern to dominant X-ray scatterers in the crystal structure. For molecular structures, dominant scatterers can include rigid structural fragments such as aromatic systems that give characteristic sets of reflections in a powder diffraction pattern and "relatively" heavy atoms with greater scattering power than most atoms found in organic crystals (H,C,N,O).[6,7]

When the dominant scatterer is a single heavy atom joined to a more complex fragment, structure solution can begin by translating this dominant scatterer attempting to match some of the more dominant reflections in the experimental pattern. Once this heavy atom has been correctly located, it can act as a pivot point for the rest of the structure thus making the remainder of the structure solution more efficient. In cases where there are no ‘dominant scatterers’, the identification of the correct orientation of a rigid molecular fragment is more important in the direct space search, as the orientation of the fragment often has a far greater impact on the quality of fit between simulated and experimental powder patterns than the fractional position of the fragment. Thus, division of the DE process between multiple populations of a cooperative coevolutionary DE that sequentially explore the orientational and translational parameters could significantly enhance the efficiency of the structure solution. In this approach, the orientational parameters could be identified first and the translational parameters considered when a certain quality of fit between simulated and experimental patterns had already been achieved by the orientational search alone).

Alternatively, the R_{wp} landscape representing the complete optimisation problem could be divided into smaller ’territories’. Each ‘territory’ could then be explored by a single population that is confined by ‘territory-specific-population-boundaries’ (analogous to the independent evolution and branching of a species that is confined to isolated islands).[8] The increased efficiency of this island coevolution search is brought about by the use of multiple populations, each with its own unit cell volume and molecular orientational arrangements to explore. Although only one population would be able to locate the global minimum, (others are clearly excluded from finding the global minimum, through the definition of territory boundaries), this population would have less landscape area to explore, with fewer local minima, and thus find the global minimum faster than one population exploring the whole landscape. Knowledge of the crystal symmetry could be used to further reduce the unit cell volume searched. For example, if a mirror plane is known to bisect the unit cell or the molecule lies on an inversion centre it would not be necessary to divide up and search the whole unit cell. If through symmetry multiple equivalent territories exist, it would only be necessary to search one of the equivalent territories as the position of scattering matter in equivalent territories could be inferred and used to generate a complete model structure.

C.2 Aspects of cooperative coevolutionary algorithms

C.2.1 Separable and non-separable parameters

Although it has been demonstrated [1] that cooperative coevolutionary global optimisation algorithms (CCGOAs) (that simultaneously use multiple populations to separately optimise different problem parameters) can evolve each parameter to its optimal value and assemble a complete optimal solution in fewer generations than a traditional type algorithm using one population, it is not always possible to optimise each parameter in complete isolation.[1,4,5,9,10] The original CCGOA developed by Potter and De Jong [1] consists of associated genetic algorithms that each optimise one problem parameter. Potter and De Jong initially used their algorithm to solve mathematical functions by using each population to optimise the value of one function variable. Individuals in each population are selected and combined and the separately optimised function variables are evaluated as a whole solution to the equation. Functions consisting entirely of separable variables could be solved by the cooperative coevolutionary GA requiring the calculation of fewer fitness evaluations than a traditional type GA using one population. However functions that involved the calculation of a product term between non-separable variables generally could not be solved in fewer fitness evaluations by the coevolutionary GA. If each population is assigned one variable and is used to optimise that variable in isolation from other variables, it is possible for each population to determine a value for the specific variable that is optimal for that particular population but when combined with the other separately optimised variables produces a non-optimal solution. [10] This limitation of CCGOAs in theory extends to the problem of direct space structure solution.

If the algorithm developed by Potter and De Jong was used to solve the crystal structure of baicalein for example via the direct space method, seven populations of individuals would be generated to optimise the seven parameters defining a model. However, in each population only one parameter evolves. In theory, the individual in each population that is assigned an R factor with the lowest value is defined by the best evolving parameter. However, since only one parameter is optimised by each population, in this case each individual is defined by six parameters with randomly generated values. The molecular asymmetry of baicalein means that the orientation of a model in the unit cell has greater influence on the fit between a pattern simulated for the model and the experimental diffraction pattern than the fractional position of the model. Thus the value of an R factor assigned to the individual in a population used to optimise a translational parameter is influenced more by the parameters defining orientation. Thus the individual that is assigned an R factor with the lowest value is likely to be the individual that is defined by the best combination of randomly generated orientational parameters and not necessarily the optimal translational parameter. This simple example demonstrates that it is not necessarily more efficient to separate and optimise parameters using separate populations.

A simple strategy that could be used to reduce this effect is to periodically exchange genetic material between the fittest individuals in each population. This information can be used to ‘update’ the values of the non-evolving parameters so that each individual in each population is defined by one parameter that is optimised by that population and parameters optimised by and copied from the best individuals in the other populations. However, a potentially more successful strategy is to identify interdependent non-separable parameters and to combine non-separable parameters into sets that are each optimised by one population.

C.2.2 Automatic problem decomposition

Although it is theoretically possible to manually decompose a problem that has non-separable parameters into sets of non-separable parameters and to use one population to optimise each set, it is often unclear to a human operator which parameters are non-separable.[4,9] Thus manual problem decomposition is likely to result in the wrong parameters being combined into one set. Additionally the manual decomposition of a problem at the start of the optimisation process means that parameters are assigned to specific sets and cannot be reassigned into different sets if the interdependence between parameters changes during the optimisation. It is therefore more efficient to make the decomposition process computer controlled. Making the decomposition process automatic, means that if the interdependence between parameters changes during optimisation the decomposition can be adjusted so that parameters that evolve and become non-separable can be optimised by one population.[4,9]

In the implementation of cooperative coevolutionary DE developed by Tang *et al* [4], problem decomposition is achieved by initially decomposing a problem defined by N parameters into N populations and using each population to optimise one specific parameter. Each population is also assigned a population-specific ‘separability’ parameter. Initially each population is assigned a separability parameter with a randomly generated value in the range zero and one. During a ‘cycle’ of generations each of the populations is used to optimise the assigned parameter. At the end of the ‘cycle’ parameter values are copied from individuals in each population to construct possible solutions in a process known as ‘collaboration.’ During ‘collaboration’ the separability parameters are themselves optimised. As each population is only defined by one separability parameter the extra computational demand required to optimise the separability parameters is insignificant compared to the computational demand required to optimise the actual problem parameters.[4] As the value of a separability parameter increases, the probability that the population will ‘merge’ with other populations increases. When populations merge the problem parameters that are assigned to each of the separate populations are combined into one set and optimised by one population in the next cycle. As a result redundant populations are eliminated. If the ‘merger’ of populations produces fitter individuals the value of the separability parameter is increased, increasing the probability that the particular problem parameters continue to be optimised together. However if the ‘merger’ of populations results in individuals being assigned lower fitness values the value of the separability parameter is decreased and the parameters separated at the end of the cycle and reassigned to separate populations. This allows different sets of parameters to form and decay as the interdependency between parameters changes during optimisation.

C.3 Collaboration

Since the different populations of a CCGOA only optimise one parameter (or set of interdependent parameters), even if each population contains complete individuals that can evolve independently for a ‘cycle’ of generations, it is necessary for individuals in different populations to periodically collaborate and combine parameters to construct the best possible solution. [1-3,9] Depending on the type of problem it is possible that multiple equally optimal solutions can be found, depending on how different combinations of parameters are combined. For example, multiple equally optimal process schedules of an oil refinery [5] or industrial chemical reaction process [11] are possible, where factors such as pressure, temperature, material flow rate, reaction rate and manufacturing cost are considered. Collaboration therefore not only generates the best possible solution, collaboration can identify different combinations of parameters that produce equally optimal solutions. Different collaboration schemes may be more appropriate for different optimisation problems.[1,3,12] It may be more appropriate to optimise a problem that has one definite solution such as a mathematical function or crystal structure using a collaboration scheme that constructs a solution from parameters copied from the best individual in each population.[1,3] However, it may be more appropriate to optimise a problem that has multiple equally optimal solutions using a collaboration scheme that constructs a solution from parameters copied from individuals in each population regardless of fitness value. This increases the probability that multiple equally optimal solutions are evaluated.[3,12]

In other circumstances [1,13] an algorithm decomposes a problem into populations that contain incomplete individuals (individuals that are not defined by a complete set of parameters), which cannot be directly evaluated. Thus to evaluate one individual it is necessary for populations to collaborate constantly to generate a complete solution by combining 'test' parameters copied from incomplete individuals.[1,13] Thus there is a choice of collaboration scheme. One combination of parameters copied from the best individual in each of the populations can be combined with a 'test' parameter, or multiple combinations of parameters copied from different individuals in each population can be combined with a test parameter. In addition, if a test parameter is combined and evaluated with multiple combinations of parameters, the optimisation procedure becomes more complex. A test parameter can be combined with parameters copied from a number of relatively fit individuals in other populations, or a test parameter can be combined with parameters copied from individuals that have been selected at random from the other populations. Furthermore, if a test parameter is combined and evaluated with multiple combinations of parameters the test parameter can be assigned three different fitness values; 'optimistic', 'hedge' and 'pessimistic. [3,12]

The 'optimistic' strategy involves assigning a test parameter the highest fitness value achieved by combining the test parameter with the best combination of parameters. The hedge strategy involves assigning a test parameter the mean fitness value calculated from all the combinations and the pessimistic strategy involves assigning a test parameter the lowest fitness value achieved by the worst combination. Therefore two main aspects of the collaboration process that require careful consideration are; (a) the collaboration scheme and (b) the method of fitness assignment.

C.3.1 Collaboration schemes

Four common collaboration schemes are complete, best, random and mixed.[3,12] Complete collaboration involves evaluating each parameter with every possible combination of parameters. Best or 'elite' collaboration involves evaluating a 'test' parameter with parameters copied from the best individuals. Random collaboration involves evaluating a 'test' parameter with one or more combinations of parameters copied from randomly selected individuals. Mixed collaboration combines the best and random schemes. Although complete collaboration guarantees that the best combination of parameters is evaluated and thus the best possible solution located, complete collaboration is extremely computationally demanding.[10] If a problem is decomposed into J populations each containing *NP* individuals, J^*NP* combinations need to be evaluated to find the best combination. Thus a CCGOA employing the complete collaboration scheme will require significantly more real time to solve a problem than CCGOAs using alternative collaboration schemes. The best, random and mixed collaboration schemes only sample the various possible combinations, however, CCGOAs that use the best, random and mixed collaboration schemes are significantly less computationally demanding.

Potter and De Jong [1] demonstrated that a cooperative coevolutionary GA using the best collaboration scheme could solve functions comprised of entirely separable variables requiring the calculation of fewer fitness evaluations than a traditional type GA. However, when used to solve functions containing non-separable variables the coevolutionary GA required the calculation of more fitness evaluations than the traditional type GA. A second cooperative coevolutionary GA using a mixed collaboration scheme that selected the best and one random combination solved the functions containing non-separable variables requiring the calculation of approximately the same number of fitness evaluations as the traditional GA. However, when used to solve functions comprised of separable variables the coevolutionary GA using mixed collaboration required the calculation of more fitness evaluations than the coevolutionary GA using best collaboration.

Compared with best collaboration, the use of mixed or random collaboration increases the probability that a test parameter is evaluated with a diverse range of multiple combinations of parameters during a single collaboration event. However, since multiple combinations of parameters are combined with each test parameter, a mixed or random collaboration scheme is intrinsically more computationally demanding than the best collaboration scheme. Thus the coevolutionary GA using mixed or random collaboration often calculates more fitness evaluations than the coevolutionary GA using best collaboration when used to solve functions comprised of separable variables. [3] Furthermore, since the genetic diversity in the different populations decreases as individuals converge, the probability that a test parameter is evaluated with a genetically diverse range of combinations of parameters decreases. Thus as a search converges, the advantage of mixed or random collaboration over best collaboration decreases. Therefore unnecessary use of mixed collaboration can in fact decrease the efficiency of the optimisation.

A possible solution to increase the probability that a CCGOA converges requiring the calculation of as few fitness evaluations as possible (without prior knowledge of whether the problem is separable or non-separable), is to use a mixed collaboration scheme but to reduce the number of randomly selected combinations of parameters used to evaluate each test parameter as the search converges.[14] Compared with best collaboration this strategy increases the probability that a test parameter is evaluated with a diverse range of parameters during the initial generations, but compared with mixed collaboration using a fixed number of collaborators, only evaluated with better combinations of parameters when the search is close to convergence.

Mathematical functions comprised of separable variables and functions containing non-separable variables were solved using a cooperative coevolutionary GA employing a variable number mixed collaboration scheme.[14] This scheme selects the best and nine other individuals at random for the first five generations, and then the best and one other random individual until convergence. For comparison the functions were also solved by coevolutionary GAs using fixed number mixed collaboration. When used to solve functions containing non-separable variables, the coevolutionary GA using variable number mixed collaboration converged requiring the calculation of fewer fitness evaluations than the coevolutionary GAs using fixed number mixed collaboration.[14] This demonstrates that as a search converges and the genetic diversity of populations decrease, it is not advantageous to evaluate a test parameter with many randomly selected combinations of parameters. When used to solve functions comprised entirely of separable variables, the GA using variable number mixed collaboration converged requiring the calculation of approximately the same number of fitness evaluations as GAs using fixed number mixed collaboration.[14] Since the results presented in reference 14 do not compare the coevolutionary GA using variable number mixed collaboration with a coevolutionary GA using best collaboration, it is not possible to determine whether variable number mixed collaboration is more efficient than best collaboration when used to solve functions comprised entirely of separable variables. However, these results do demonstrate that variable number mixed collaboration is no less efficient than fixed number mixed collaboration when used to solve functions comprised entirely of separable variables.

Although Panait and Luke [14] suggest that it is potentially more efficient to control the number of randomly selected collaborators in proportion to the genetic diversity in the populations, (‘gradually’ decreasing the number of randomly selected collaborators as a search converges), these initial results demonstrate the increase in efficiency achieved by reducing the number of randomly selected collaborators after an arbitrary number of generations.

C.3.2 Fitness assignment

The use of best collaboration [1,3,5,12] necessarily means that only optimistic fitness assignment is possible. The use of complete collaboration means that all (optimistic, hedge and pessimistic) strategies are possible but since complete collaboration is computationally demanding it is rarely used. When using random or mixed collaboration, it is necessary to decide which fitness assignment strategy optimistic, hedge or pessimistic to use. The hedge fitness assignment is commonly used in competitive coevolution [2,3,5] because an individual must be able to dominate a variety of competitive individuals. Cooperative coevolutionary algorithms using the optimistic strategy have been found to be more efficient (computing fewer fitness evaluations per search) than cooperative algorithms using the hedge or pessimistic strategies regardless of whether the random or mixed collaboration scheme is used. [3,5,14] If a relatively optimal test parameter is combined with different combinations of relatively non-optimal parameters and pessimistic fitness assignment is used to evaluate the combinations, the fitness value of the worst combination is assigned to the test parameter. Thus use of the pessimistic strategy increases the probability that a relatively optimal test parameter is assigned a low fitness value and overlooked. Similarly, if a relatively optimal test parameter is combined with different combinations of relatively non-optimal parameters and the hedge fitness assignment is used, the test parameter is likely to be assigned a relatively low fitness value. Use of pessimistic and hedge strategies is likely to increase the number of fitness evaluations calculated by a search. The use of the hedge and pessimistic fitness assignments or a traditional type global optimisation algorithm (where all parameters are optimised simultaneously by one population) decreases the probability that a relatively optimal test parameter is identified until the optimisation has located many near optimal parameters, increasing the probability that an individual defined by many optimal parameters is evaluated. Conversely, if a relatively optimal test parameter is combined with different combinations of relatively non-optimal parameters and the optimistic fitness assignment is used, the fitness value of the best combination is assigned to the test parameter. Thus use of the optimistic strategy increases the probability that an optimal test parameter is assigned a relatively high fitness value and rapidly identified. Thus use of the optimistic strategy increases the probability that a search quickly completes optimisation calculating as few fitness evaluations as possible.

C.4 Direct space structure solution using hypothetical cooperative coevolutionary differential evolution CCDE

C.4.1 Separability

The structure solution of asymmetric molecules means that not all the parameters defining the position and orientation of a model in the unit cell are separable. Since the orientation of an asymmetric model in the unit cell has a greater impact on the fit between simulated and experimental diffraction patterns than the position of the model, it is unlikely that a population used to optimise a parameter defining the position of the model in complete isolation from parameters defining the orientation of the model will locate the optimal position. In addition, the orientation of an asymmetric model about a certain rotation axis potentially has a smaller influence on the value of an *R* factor assigned to a model in a certain position along the rotation axis than the orientation of the model about other rotation axes. For example, the rotation of models about the *x* axis potentially has a smaller effect on the value of the *R* factors assigned to models at different positions along the *x* axis than the rotation of the models about the *y* and *z* axes. Thus certain structural parameters are likely to have different levels of separability during optimisation. This means that a manual decomposition of a structure solution problem and the assignment of specific structure parameters to specific populations is likely to reduce the probability that an optimal solution is located. Therefore it is necessary to use an automatic decomposition such as that developed by Tang et al [4] that can adjust the problem decomposition during optimisation. However, it is potentially beneficial to initially optimise all structure parameters in one population and to decompose the problem into separate populations after a certain number of generations. Figures 3.2 and 3.3 demonstrate that the value of the mean *R* factor calculated for the whole population decreases relatively rapidly during initial generations as many poor models defined by bad combinations of parameters are frequently replaced by better models. This suggests that it may be more advantageous to optimise all structure parameters using one population until the rate of change in the value of the mean *R* factor decreases (around the 20th generation) when it becomes more advantageous to optimise different parameters using separate populations.

C.4.2 Collaboration scheme

Once separate populations are used to optimise different structure parameters during ‘cycles’ of generations, it will become necessary at the end of each cycle to copy parameters from individuals in each population to evaluate the separately optimised parameters or sets of parameters. Thus a collaboration scheme is needed. The previous work by Potter *et al* [1] and De Jong *et al* [3,12] demonstrates that cooperative coevolutionary GAs using the best collaboration scheme calculate more fitness evaluations than traditional GAs when used to solve functions containing non-separable variables. Since not all the structural parameters are separable the best collaboration scheme is not suitable. The use of a mixed collaboration scheme is more efficient for optimising problems containing non-separable problems and decreases the rate at which the amount of genetic diversity decreases, reducing the probability that a search converges prematurely. However the advantage of mixed collaboration decreases as a search converges and an algorithm using a mixed scheme potentially calculates more fitness evaluations than is necessary.

Using a variable number mixed collaboration scheme [14] can reduce the number of fitness evaluations calculated, and decrease the rate at which the amount of genetic diversity decreases, reducing the probability that a search converges prematurely. Therefore a search using a variable number mixed collaboration scheme is potentially the most efficient scheme for direct space structure solution. However, unlike the cooperative coevolutionary GA developed by Panait and Luke [14] which uses collaboration to evaluate each parameter in each generation, the CCDE would only use collaboration to evaluate parameters at the end of each cycle of generations. Thus during a search most fitness evaluations would be calculated to evaluate models generated during the optimisation cycles and relatively few evaluations would be calculated during collaboration. Thus the number of combinations of parameters selected during each collaboration event would have less impact on the total number of fitness evaluations calculated. It is therefore practical to select a considerable number of combinations of parameters during each collaboration event and only decrease the number when all populations are close to convergence. When many of the combinations of parameters are assigned *R* factors of similar value it indicates that the populations are close to converging and that switching to the best collaboration scheme is likely to decrease the total number of fitness evaluations calculated without significantly increasing the probability of premature convergence.

C.4.3 Cycle length

In the implementation of CCDE developed by Tang *et al* [4], once a parameter or set of parameters are assigned to one population the parameters are optimised for a cycle of 50 generations before collaborating. This arbitrary cycle length is likely to be non-optimal for our direct space structure solution and the appropriate cycle length can only be found through experimentation. However, it is suggested that if one population locates a new best individual that is assigned a significantly lower *R* factor (such as 5%), the cycle should be interrupted so that the parameters of this individual can be evaluated through collaboration and communicated to the other populations.

C.4.4 Fitness assignment

When applied to cooperative coevolutionary algorithms the use of the optimistic fitness assignment increases the probability that an optimal parameter is quickly identified. Thus to decrease the total number of fitness evaluations calculated by a search, a CCDE applied to direct space structure solution should use the optimistic fitness assignment.

References

[1] M. A. Potter and K. A. De Jong. A Cooperative Coevolutionary Approach to Function Optimization. Third Parallel Problem Solving From Nature. Jerusalem Israel. (1994). 249.

[2] C. D. Rosin and R. K. Belew. New Methods for Competitive Coevolution. *J. Evol. Comp*. (1997). **5**. 1.

[3] R. P. Wiegand, W. C. Liles and K. A. De Jong. An Empirical Analysis of Collaboration Methods in Cooperative Coevolutionary Algorithms. *Proc. Gen. Evol. Conf*. Morgan Kaufmann Publishers.

[4] Z. Yang, H. Tang and X. Yao. Large Scale Evolutionary Optimization using Cooperative Coevolution. *Info. Sci*. (2008). **178**. 2985.

[5] L. M. Simao, D. M. Dias and M. A. l. C. Pacheco. Refinery Scheduling Optimization using Genetic Algorithms and Cooperative Coevolution. IEEE Symp. Comp. Intel. Sched. (2007). 151.

[6] K. D. M. Harris and M. Tremayne. Crystal Structure Determination from Powder Diffraction Data. *Chem. Mater*. (1996). **8**. 2554.

[7] K. D. M. Harris, M. Tremayne, P. Lightfoot, and P. G. Bruce. Crystal Structure Determination from Powder Diffraction Data by Monte Carlo Methods. *J. Am. Chem. Soc.* (1994). **116**. 3543**.**

[8] A. P. Englelbrecht. Computational Intelligence: An Introduction. John Wiley and Sons Ltd. Chester. (2002).

[9] M. H. Nguyen, H. A. Abbass and R. I. McKay. Analysis of CCME: Coevolutionary Dynamics, Automatic Problem Decomposition and Regularization. IEEE Trans. Syst. Man and Cybe. *Appl. Rev. C.* (2008). **38**. 100.

[10] C. K. Goh and K. C. Tan. A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization. IEEE Trans. *Evol. Comp*. (2009). **13**. 103.

[11] M. H. Lee, C. Han and K. S. Chang. Dynamic Optimization of a Continuous Polymer Reactor Using a Modified Differential Evolution Algorithm. *Bid. Eng. Chem. Res*. (1999). **38**. 4825.

[12] E. Popovici and K. A. De Jong. A Dynamical Systems Analysis of Collaboration Methods in Cooperative Coevolution. Am. Ass. Arti. Intel. (2005). http://www.aaai.org

[13] V. R. Khare, X. Yao and B. Sendhoff. Credit Assignment Among Neurons in Co-evolving Populations. http://www. cs.bham.ac.uk/research/NC.

[14] L. Panait and S. Luke. Time-dependent Collaboration Schemes for Cooperative Coevolutionary Algorithms. Am. Ass. Arti. Intel. (2005). http://www.aaai.org