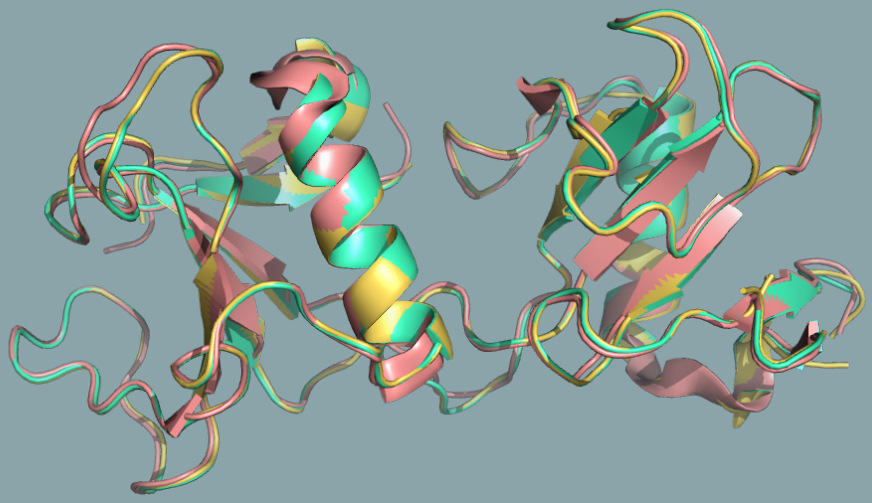
Genetic Algorithm as a Computational Approach for Phase Improvement and Solving Protein Crystal Structures



Sravya Mounika Kantamneni
Department of Chemistry, MIN Faculty, University of Hamburg
October 2020

Genetic Algorithm as a Computational Approach for Phase Improvement and Solving Protein Crystal Structures

Dissertation with the aim of achieving a doctoral degree at the

Faculty of Mathematics, Informatics and Natural Sciences

Department of Chemistry of the University of Hamburg

Submitted by

Sravya Mounika Kantamneni

European Molecular Biology Laboratory, Hamburg

Hamburg

06 October 2020

To my family and my mentor Dr. Sachchidanand

The work presented in this thesis was carried out during the period from June 2016 to May 2020 under external supervision by

Dr. Victor S. Lamzin

at the European Molecular Biology Laboratory (EMBL), Hamburg. University support was provided by

Prof. Dr. Andrew E. Torda

from the research group for Biomolecular Modeling of the Centre for Bioinformatics from the Faculty of Mathematics, Informatics and Natural Sciences at the University of Hamburg.

The members of the Thesis Advisory Committee (TAC), mandatory by EMBL, in addition to Dr. Victor S. Lamzin and Prof. Dr. Andrew E. Torda, also included Dr. Thomas R. Schneider (Macromolecular Crystallography, EMBL, Hamburg), Prof. Dr. Gerard J. Kleywegt (Molecular and Cellular Structure, EMBL-EBI, Hinxton) and Prof. Dr. Richard J. Morris (Computational and Systems Biology, John Innes Centre, Norwich)

1. Evaluator: Prof. Dr. Andrew E. Torda
2. Evaluator: Dr. Thomas R. Schneider

Zusammenfassung

In der Röntgenstrukturanalyse von Makromolekülen wird zur Berechnung der Elektronendichte an einer bestimmten Position (x, y, z) die Fourier-Transformation der Strukturamplituden sowie der Phasenwinkel an dieser Position innerhalb der asymmetrischen Einheit des Kristalles benötigt. Strukturamplituden können mithilfe der Intensitäten des Beugungsmusters berechnet werden, wohingegen im Experiment keine direkten Informationen über die Phasenwinkel gemessen werden. Die fehlende Phasenwinkel-Information kann durch zusätzliche Methoden wie experimentelle Phasenwinkelbestimmung oder Molekularen Ersatz erhalten werden. Die so gewonnenen Anfangsphasen sind nicht immer ausreichend genau, um eine interpretierbare Elektronendichtekarte zu erhalten. Dadurch können zusätzliche Schritte zur Verbesserung der Phasenwinkel nötig werden, wie etwa Dichte-Modifikation und/oder Modell-Verbesserung. Berücksichtigt man die Komplexität der Suche nach korrekten Phasenwinkeln, weisen heuristische globale Optimierungsverfahren basierend auf genetischen Algorithmen (GA) besondere Vorteile auf.

Diese Arbeit befasst sich mit dem Problem der Phasenoptimierung unter Zuhilfenahme genetischer Algorithmen. Die folgenden wesentlichen Punkte wurden untersucht: Die Optimierungsmethode und Optimierungsparameter sowie die Fitnessfunktion, die optimiert werden soll. Die Hauptaufgabe bei der Optimierung eines Algorithmus ist die Entwicklung einer Population oder verallgemeinert, der Arten von Mutations-, Cross-over und Selektionsoperatoren des verwendeten genetischen Algorithmus. Als beste Herangehensweise erwies sich die Verwendung des Karten-Korrelationskoeffizienten für die Fitnessfunktion. Die besten Resultate wurden durch eine Optimierungsstrategie, mit mehreren Rekombinationen und Turnierselektion mit einer Turniergröße größer zwei, erzielt. Um eine Fitnessfunktion zu ermitteln, die unter Realbedingungen eingesetzt werden kann, wurde eine Reihe von Tests durchgeführt. Diese dienten dazu die Möglichkeit der Darstellung von Charakteristika der Elektronendichtekarte und der Kartenkonnektivität als Fitnessfunktion zu ermitteln. Eine Fitnessfunktion mit einer Kombination von Schiefe und Konnektivität, im Falle einer mittleren Auflösung, und Schiefe alleine, in Falle hoher Auflösung, erwiesen sich als annehmbar gute Fitnessfunktionen. Die derzeitige Umsetzung ist noch auf die

Raumgruppe $P2_12_12_1$ beschränkt, aber die Erweiterung des Programmcodes, um auch weitere in Kristallen von Bio-Makromolekülen vorkommende Symmetrien einzuschließen, sollte kein Problem darstellen.

Summary

In macromolecular X-ray crystallography, calculating the electron density at a specific position (*x,y,z*), requires Fourier transformation of the structure factor amplitudes and phases at that position within the asymmetric unit of a crystal. Structure factor amplitudes can be calculated from the intensities of diffraction spots, while phase information is not recorded in the experiment. The lost phases can be recovered using either an additional experiment or molecular replacement. The initial phases obtained by these methods are not always sufficiently accurate to produce an interpretable density map. Additional phase improvement steps using density modification and/or model refinement approaches are required. Given the complexity of the phase space to be searched, heuristic global optimisation techniques based on genetic algorithms (GAs) may have their own advantages.

In this work the phase optimisation problem is addressed using genetic algorithms. The following main issues have been investigated: the optimisation method and parameters, and the fitness function to be optimised. For the optimisation of an algorithm, the important issue is the development of the population, or, more generally, the types of mutation, crossover and selection operators of the genetic algorithm to be used. The best design for the problem was identified by using the map correlation coefficient as a fitness function. The best results were achieved by optimisation using a mixture of multiple crossovers and a tournament selection with size of two. To identify a fitness function that can be used in real cases, a series of tests were performed to assess the applicability of the characteristics of the density map and map connectivity as a fitness function. A fitness function with a combination of skewness and connectivity for medium-resolution test cases, and skewness alone for high-resolution test cases were found to be reasonable best fitness functions. The current implementation is limited to the space group $P2_12_12_1$, but there should be no problem extending the code to handle other symmetries common for crystals of biomacromolecules.

Preface

Parts of this thesis (text and figures) have already been or will be submitted for publication in a peer-reviewed journal and have been presented as posters and oral presentations at conferences and workshops.

Peer-Reviewed Publications

Kantamneni S.M., Lamzin V.S. Phase improvement using genetic algorithms. *Acta Crystallographica D* (in preparation).

Kantamneni S.M., Lamzin V.S. Map characteristics as a function to analyse the quality of electron density maps at different resolutions of the X-ray data. *Acta Crystallographica D* (in preparation).

Oral Presentations at Conference and Meetings

"Automated model building in ARP/wRP", Software Fayre, 24th Congress and General Assembly of the International Union of Crystallography, Hyderabad, India, August 2017.

"Automated model building in ARP/wRP", Satellite meeting, 24th Congress and General Assembly of the International Union of Crystallography, Hyderabad, India, August 2017.

"Automated model building in ARP/wARP", Lunchtime bytes, CCP4 Study Weekend 2018: Multi and Serial Crystal Data Collection and Processing, Nottingham, UK, January 2018.

"Shaping up macromolecules", pedestal presenter, EMBL gala dinner 2018, Heidelberg, Germany October 2018.

"Automated model building in ARP/wARP", ECM32, Vienna, Austria, August 2019.

Oral presentations at Workshops

"Automated model building in ARP/wARP" CAS/CCP4 Workshop and Computational Crystallography School: From data processing to structure refinement, Guangzhou Institute of Biomedicine and Health, Chinese Academy of Sciences, Guangzhou, China, October 2017.

"Automated model building in ARP/wARP" CCP4/BGU Structure Solution Workshop, Ben-Gurion University of the Negev, Beer-Sheba, Israel, February 2018.

"Automated model building in ARP/wARP". CCP4/Spring-8 School and Workshop: From data processing to structure refinement and beyond, Osaka, Japan, October 2018.

"Automated ligand building in ARP/wARP". South-East Asian Crystallographic Overview And Systemic Training (SEA COAST 2020), Bangkok, Thailand, January 2020.

Poster Presentations at Conferences and Meetings

Kantamneni S.M., Lamzin V.S. Recent developments in ARP/wARP. 24th Congress and General Assembly of the International Union of Crystallography, Hyderabad, India, August 2017.

Kantamneni S.M., Lamzin V.S. Recent advances in ARP/wARP. PIER Graduate Week. Hamburg, Germany, October 2017.

Kantamneni S.M., Lamzin V.S. Recent developments in ARP/wARP. EMBL Lab Day, Hamburg, Germany, July 2018.

Kantamneni S.M., Sobolev E. Lamzin V.S. Shaping up macromolecules. ECM32, Vienna, Austria, August 2019.

Contents

Contents

Abbreviations and Notation

General abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
ASCII	American Standard Code for Information Interchange
CCP4	Collaborative Computational Project, Number 4
CPU	Central Processing Unit
EMBL	European Molecular Biology Laboratory
GA	Genetic Algorithm
MCC	Map correlation coefficient
MR	Molecular Replacement
MX	Macromolecular X-ray Crystallography
NCS	Non-Crystallographic Symmetry
PDB	Protein Data Bank
PDBe	Protein Data Bank in Europe
R.M.S.D.	Root Mean Square Deviation or Distance
RNase SA	Ribonuclease from *Streptomyces aureofaciens*
SAD	Single -wavelength Anomalous Dispersion
SAICAR	Saicar synthase from *Saccharomyces cerevisia*
TEMs	Transmission Electron Microscopes
Cryo-EM	Cryogenic Electron Microscopy
NMR	Nuclear Magnetic Resonance

Crystallographic abbreviations

α_{hkl}	Phase angle at location (*h,k,l*) in reciprocal space
F_{calc}	Calculated structure factor amplitude
F_{obs}	Observed structure factor amplitude
F_{hkl}	Structure factor at location (*h,k,l*) in reciprocal space
I_{hkl}	Intensity of a diffracted reflection (h,k,l)

R	Crystallographic R-value
ρ_{xyz}	Electron density at location (*x,y,z*) in real space
V	Volume of the unit cell
$\mathring{A}$	Angstrom (Unit of resolution)
Å^2	Angstrom squared (Unit of Wilson B factor)
°	Degree (Unit of phase error)
σ	Density contour level

Genetic algorithm abbreviations

$P_d(t)$	Loss of diversity at generation t
$P(x)$	Probability of selecting point x
N_{mut}	Number of mutations
t	Size of the tournament
N	Population size
P_c	Crossover probability
P_m	Mutation probability
w	Weighting factor

Introduction

How to determine the structure of the biological molecules?

What is X-ray Crystallography?

What is missing in X-ray data?

What are methods to recover the missing information in X-ray crystallography?

The quest for insights into the structure of biological materials started nearly 380 years ago, when Robert Hooke (Hooke, 1665) and Antoni Van Leuwenhoek (Leeuwenhoek & Hoole, 1800; Lane, 2015) first attempted to use microscopes to study organisms which cannot be seen by the naked eye. By the late 19th century, light microscopy was explored to its theoretical resolution limit by microbiologists (Bracegirdle, 1989). It was then clear that light with a higher order wavelength (~500 nm) cannot be used to provide structural details at the atomic level, as the interatomic distance (~0.2 nm) is several magnitudes lower than that of the wavelength of the light.

The solution to this was found shortly after the discovery of X-rays, which have a shorter wavelength of ~0.01 to 10 nm, by Wilhelm Conrad Röntgen in 1895 (Röntgen, 1896) and based on the Johannes Kepler conjecture on internal hexagonal symmetry of crystalline snowflakes (Kepler, 1966; Hoinkes, 1967). As the X-ray wavelength range aligns with the interatomic distance range in a crystalline molecule, they can be diffracted by electrons in the molecule to obtain a diffraction pattern. This was confirmed by the diffraction experiments conducted by Max Von Laue together with Paul Knipping and Walter Friedrich at LMU in 1912 on Zinc blende and a ZnS crystal (Ewald, 1962).

Based on Laue's experiments, William Henry Bragg in 1913 together with William Lawrence Bragg used X-rays to solve the crystal structure of NaCl and many other inorganic molecules (Bragg, 1913). As X-rays are scattered by electrons, the spots in the diffraction pattern contain the information on the electronic configuration of the molecule. To derive these positions from the diffraction spots, W.L. Bragg developed

mathematical foundations (Bragg, 1929) based on the “Fourier Transforms” (formulated by Jean Baptiste Joseph Fourier in 1822).
In the 1930s, many molecular biologists started to use X-rays to study the structure and function of the biological molecules (Kay, 1996). A few prominent works such as Delbrück’s research on bacteriophages (Delbrück, 1966; Holliday, 2006), Watson and Crick’s discovery of the structure of the DNA double helix (Watson & Crick, 1953), William Astbury’s work on the structure of Keratin and DNA (Astbury & Street, 1932) and Pauling and Corey’s discovery of the structure of alpha helix (Pauling & Corey, 1951, 1953) have proved that X-rays are a powerful form of radiation that can also be used to decipher the structure of biological macromolecules.

Owing to the success of X-ray crystallography (Campbell, 2002; Strandberg *et al.*, 2009; Schwarzenbach, 2012), this work focuses on solving the phase problem currently seen in macromolecular X-ray crystallography and mainly covers protein crystallography. Among various biological macromolecules, proteins play a crucial role in regulating many body functions and are involved in many disease mechanisms. X-rays were helpful in elucidating many protein structures. Nearly 160,000 (as of January 2020) structures of protein molecules are deposited in the PDB (Burley *et al.*, 2019). The first diffraction pattern of a protein, pepsin, was obtained by J. D. Bernal and Dorothy Crowfoot Hodgkin, in 1934 (Bernal & Crowfoot, 1934) which changed the dogma that proteins were “colloids” with random structures (Perutz, 1985). John Kendrew solved the first macromolecular structure of myoglobin at 6 Å resolution (Kendrew *et al.*, 1958) in 1958, followed by the 5.5 Å structure of haemoglobin solved by Perutz in 1959 (Perutz *et al.*, 1960). The first near atomic resolution structure of myoglobin to a resolution of 2 Å was solved by Kendrew and others in 1960 (Kendrew *et al.*, 1960). A brief review of structure determination methods available and their applicability is presented in the following section.

How to Determine the Structure of the Biological Molecules?

An overview on Structure Determination Methods

Over the past few decades, the development of many structure determination methods (Figure C.1) has revolutionised the structure solution process (Campbell, 2002).

Among these methods, a few prominent ones are discussed here. This description does not cover the complete list of methods available, but only includes the most popular ones used either in a complementary manner or in rivalry to X-ray crystallography.

Diffraction method employs various radiation sources such as X-ray, neutron and electron (Figure 1). Among these, the most widely used radiation for the structure determination of macromolecules is X-ray, highlighted in Figure 1.

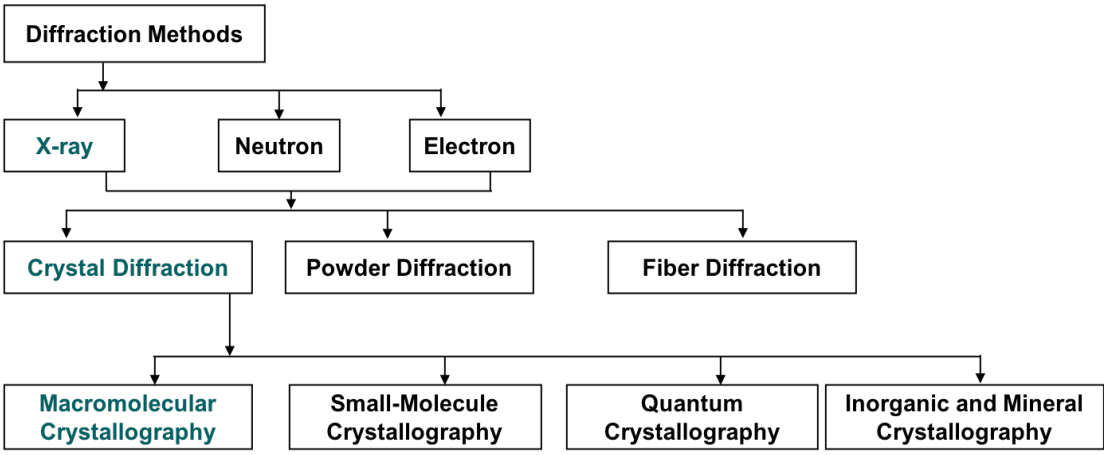


Figure 1 Classification of diffraction methods. Macromolecular crystallography, which is a focus of this work and its classification is highlighted in teal. Powder and fibre diffraction are not covered in this work and hence not classified further.

X-ray Diffraction

X-ray crystallography is the focus of this thesis work, and is hence described both separately in the section “What is X-ray Crystallography?”, and also in the subsequent parts of this chapter.

Neutron and Electron Diffraction

Neutron and electron diffraction methods are analogous to X-ray diffraction. X-ray diffraction is based on the interaction of X-ray with the electron cloud of an atom, neutron diffraction is based on the interaction of neutrons with atomic nuclei, while electron diffraction on the interaction of electrons with the electrostatic field of an atom. In 1927, two groups – Davisson and Gremer, Thomson and Reid carried out the first successful electron diffraction experiments using low-energy and high-energy

electrons respectively (Davisson & Germer, 1927; Thomson & Reid, 1927). It soon gained popularity in inorganic crystallography as it can be used on much smaller crystals compared to X-ray diffraction. It is also possible to automatically determine the 3D structure of molecules using this method. Until a decade ago, it was mainly used for the structure determination of inorganic molecules as they are less affected by the radiation than organic molecules (Warren, 2018). However, in 2013 Tamir Gonen developed microED that could be used to determine the structure of biological macromolecules (Shi *et al.*, 2013). In 2018, Tim Gruene developed a device using transmission electron microscope and a compatible detector for determining the structures of small organic molecules using a beam of electrons from the microscope (Gruene *et al.*, 2018).

Neutron diffraction method can be used to provide complementary information to X-ray diffraction such as the position of hydrogen atoms (which cannot be obtained by X-ray diffraction) and its protonation states. This information can be useful in understanding mechanisms of enzyme catalysis and ligand binding (Liebschner *et al.*, 2019).

Electron Microscopy and Cryogenic Electron Microscopy (Cryo-EM)

Electron microscopy uses transmission electron microscopes (TEMs) in electron diffraction mode to study the structure of macromolecules at the atomic scale. It also allows direct visualisation of molecules, as the reconstructed images of structures can be obtained physically (De Rosier & Klug, 1968). The major disadvantage of this method is radiation damage of the sample due to high-vacuum conditions and intense electron beams, and relatively low-resolution structural details.

To avoid this, samples are cooled to cryogenic temperatures. This method, Cryo-EM, is a combination of three methods: electron tomography, electron single-particle microscopy and electron crystallography, and was developed in the 1970s (Liebschner *et al.*, 2019). With the advances in the past four decades, this has become a powerful tool to investigate the structures of large proteins, nuclei acids, and complexes of these *de novo* (Callaway, 2015). In this method, biological samples are frozen directly from the solution thus revealing the structural details in close-to-native state. Currently, it is

possible to achieve near atomic-resolution (<2Å) with this method (Banerjee *et al.*, 2016; Bartesaghi *et al.*, 2015; Merk *et al.*, 2016; Mitra, 2019).

Nuclear Magnetic Resonance (NMR)

Developed in 1946, Edward Purcell from Harvard and Felix Bloch from Stanford, NMR uses the nuclei of a sample which are excited using radio waves to produce nuclear magnetic resonance (Purcell *et al.*, 1946; Bloch, 1946). This resulting resonance is detected on sensitive radio receivers. Each molecule produces a characteristic resonance frequency based on its intramolecular composition. Thus, it provides details on the electron structure of the molecule and its functional groups. This is another powerful structure determination tool and does not require molecules to form crystals or require heavy atom derivatives, and there is no need for molecules to be bound to the microscopic grid. This can be used to study molecules that have flexible regions (Wüthrich, 1998). Molecules can be studied in their solution (a state near to physiological environments) to display their natural dynamics (Allerhand *et al.*, 1971) but can also be studied in the solid state.

NMR provides information on the composition of functional groups in the molecule, adjacent atoms (Spin-Spin coupling constants), and molecular dynamics (Wüthrich & Wagner, 1975). It is relatively good at identifying weak interactions between proteins and bound small molecules. This is the only method available to study the structures of disordered, denatured or partly folded proteins at atomic resolution (Fersht, 2008; Baum *et al.*, 1989). However, these studies are generally limited by the protein size; determination of structure of macromolecules larger than 50kDa is not possible by using NMR (Pervushin *et al.*, 1997).

With the advances in these structure determination methods, the structural information of more than 160,000 molecules are available as of January 2020. Among these, 89% of structures are solved using X-ray crystallography (Figure 2) and this is the most powerful method available to solve protein structures at atomic resolution.

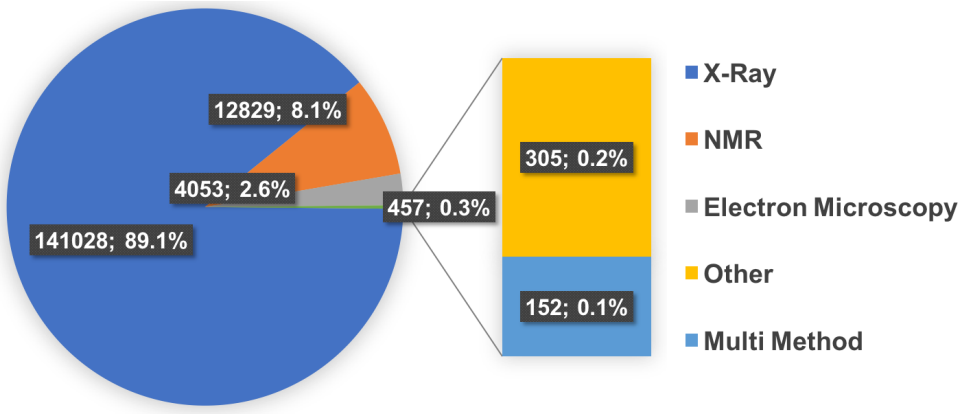


Figure 2 PDB statistics on the number of structures solved by various structure determination methods (Burley *et al.*, 2019).

What is X-ray Crystallography?

Principles and Concepts

As described earlier in this chapter, X-rays have a wavelength of the same order of magnitude as that of the interatomic distances, in a crystal and so when passed through crystalline molecules a diffraction pattern is produced. From the amplitudes and the phases of reflections, using Fourier summations (equation 2), the “electron-density” map of molecules can be computed. From this electron density map, a model of the structure can be predicted. The complete process of structure solution using X-ray crystallography is outlined in Figure 3.

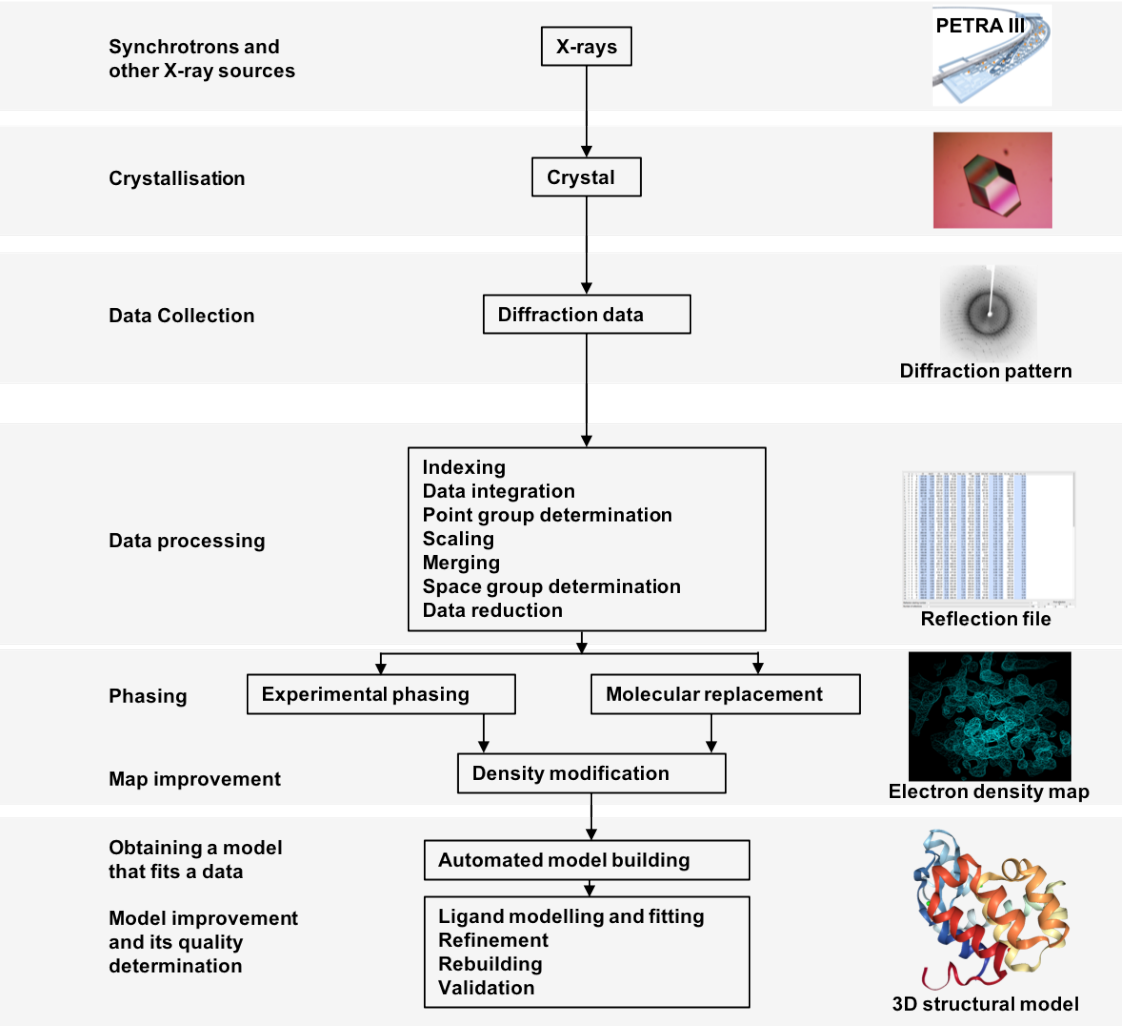


Figure 3 Steps in the structure solution process (Liebschner *et al.*, 2019). A crystal of lysozyme (PDB ID: 253L), a diffraction pattern, an electron density map and a model are shown in the right side.

What is Missing in X-ray Data?

The phase problem

After the diffraction experiment, the intensity $I(hkl)$ of each reflection or diffraction spot (hkl) is calculated and corrected by applying various correction factors such as the Lorentz, polarisation and absorption corrections (Drenth, 1999). The structure factor amplitude $|F(hkl)|$ of a reflection (hkl) can then be computed from the intensity of the reflection $I(hkl)$ by using the equation (1).

$$I(hkl) = |F(hkl)|^2 \quad (1)$$

This structure factor amplitude, together with the phase angle $\varphi(hkl)$ of a reflection (hkl), are Fourier transformed using equation (2) to produce the electron density at coordinates x, y, z of the unit cell.

$$\rho(xyz) = \frac{1}{V}\sum_{\substack{hkl \\ -\infty}}^{+\infty} |F(hkl)| \exp[-2\pi i(hx + ky + lz) + i\varphi(hkl)] \quad (2)$$

The structure factor amplitude can be computed from the reflection intensity but the phase angle information is not recorded in the experiment. Finding the phases is called the "phase problem". It is explained visually in comparison with optical microscopy in Figure 4.

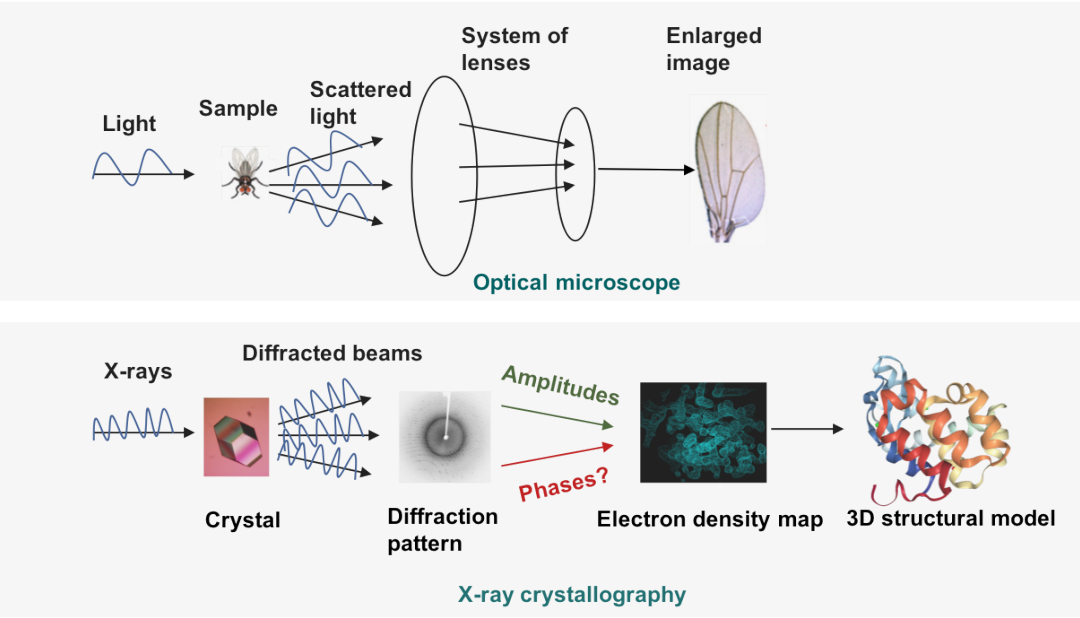


Figure 4 An illustration of the phase problem. Unlike in optical microscopy, no lenses are available to converge diffracted X-rays and produce the enlarged image. The diffraction pattern obtained after the diffraction experiment contains the information on the amplitudes (highlighted in green) of the diffracted rays, but the phase angle (highlighted in red) information is lost in the experiment. This is the so-called "phase problem" (xtal.iqfr.csic.es, 2020).

How Do We Recover the Missing Phase Values?

Phasing methods

Although there is no direct relationship between structure factor amplitudes and phases, phases can be recovered from some prior knowledge of electron density and the structure to be determined. Various phasing methods have been developed exploiting this prior knowledge (Table 1) and successfully applied to solving many crystallographic structures.

Table 1 Overview of phasing methods (Taylor, 2003).

Method	Prior Knowledge
Direct methods	Atomicity of macromolecules and positivity of electron density
Molecular replacement	Homology model
Isomorphous replacement	Substructure of heavy atoms
Anomalous scattering	Substructure of anomalous atom

Patterson Method

The Patterson method is based on the principle that, although phases are required to determine the position of peaks in the electron density which gives atomic positions, the structure factor magnitudes alone are sufficient to get the information on the relative positions of atoms in the structure.

The Patterson function, equation 3, is used to calculate the Patterson map from squared structure factor amplitudes with the phases of all reflections set to zero (Patterson, 1934).

$$P(uvw) = \frac{1}{V}\sum_{\substack{hkl \\ -\infty}}^{+\infty} |F(hkl)|^2 \cos[2\pi(hu + kv + lw)] \qquad (3)$$

where $u\ v\ w$ are $x\ y\ z$ coordinates in the Patterson cell.

Each peak in this Patterson map corresponds to the relative position vectors between a pair of atoms in the structure. With an increase in the number of atoms N, the number

of interatomic vectors $N(N-1)$ to be solved increases, making this method difficult to apply to large molecules with more than 20-50 atoms (Taylor, 2003). This method is usually used to solve the structure of small molecules when phases are not available. For macromolecules this is used in combination with other methods to derive the substructure of a macromolecule.

Direct Method

The two types of prior knowledge used in this method are non-negativity of the electron density map and the atomicity of the macromolecule. Non-negativity or positivity indicates that the electron density function in a crystal is positive everywhere. This property gives rise to statistical properties such as inequalities in the structure factors which can be used to restrict the values of the phases to a few possibilities (Harker & Kasper, 1948; Karle & Hauptman, 1950). The property of atomicity indicates that the electron density is at maximum on the position of an atom but is lower between the atomic positions.

The process of obtaining unknown phases φ from the known magnitudes $|E|$ of the normalised structure factors (the amplitudes of the point atoms at rest) is based on the concepts of positivity and atomicity and involves exploiting the relationships between these normalised structure factors and incorporating suitable recipes for origin fixing and enantiomorph selection (Hauptman, 1991). Direct methods employ probabilistic approaches such as structure invariants and semi-invariants to achieve this. The structure invariants (Hauptman, 1991) are the linear combination of phases calculated independent of the origin. In other words,

$$\psi_3 = \varphi_H + \varphi_K + \varphi_L \qquad (4)$$

(Hauptman, 1991)

ψ_3, which is a linear combination of $\varphi_H, \varphi_K, \varphi_L$ is a structure invariant (triplet), if $H + K + L = 0$.

Triplet structure invariance together with the tangent formula, equation 5, which is used to refine and extend the phases from starting known or “presumed to be known” phases, constitute the combination of the fundamental principle and the

neighbourhood principle of the direct methods. Most implementations of the direct methods are developed based on this combination (Hauptman, 1991).

$$tan\varphi_H = \frac{\langle E_K E_{H-K} \sin(\varphi_K + \varphi_{H-K})\rangle_K}{\langle E_K E_{H-K} \cos(\varphi_K + \varphi_{H-K})\rangle_K} \quad (5)$$

(Hauptman, 1991)

This method is seriously limited by the resolution of the X-ray data. With a decrease in the resolution to worse than 1.2 Å (Morris & Bricogne, 2003), the applicability of this method is reduced. While this is often used to phase small molecules up to ~1000 atoms, for macromolecules it is used in combination with other phasing methods.

Isomorphous Replacement Method

This is the phasing method used to obtain the first structures of macromolecules, myoglobin and haemoglobin by Kendrew and Perutz respectively (Strandberg *et al.*, 2009; Kirk, 2014). In this method, the crystals of the macromolecule are soaked in a heavy-atom solution to form an isomorphous (same unit cell and orientation) heavy atom derivative. As the heavy atom produces measurable intensity changes, the difference $\Delta|F|_{iso}$ in the structure factor amplitude of the native crystal $|F_P|$ and the structure factor amplitude of the derivative crystal $|F_{PH}|$ can be computed using the following equation.

$$\Delta|F|_{iso} = |F_{PH}| - |F_P| \quad (6)$$

(Drenth, 1999; Taylor, 2003)

From this difference, using a system of equations, the approximate position of the heavy atoms can be deduced using Patterson and direct methods. After refinement, the heavy atom amplitude $|F_H|$ and the phases φ_H are computed. Using the cosine rule, from equation 7, the phases of the native protein φ_P are estimated.

$$\varphi_P = \varphi_H + cos^{-1}[({F^2}_{PH} - {F^2}_P - {F^2}_H)/2F_P F_H] \quad (7)$$

(Taylor, 2003)

The errors in the computation of heavy-atom positions, structure factors, and errors due to non-isomorphism are some of the limitations of this method.

Anomalous Scattering Method

The principle of the anomalous scattering is similar to the isomorphous replacement except instead of heavy atoms, anomalous scatterers (atoms that scatter X-ray anomalously with a change in amplitude and phase at their absorption edges) are used. Anomalous scattering of the atoms violates Friedel's law (intensities of reflections hkl and $\bar{h}\bar{k}\bar{l}$ are equal if the crystal is centrosymmetric or if no resonant scattering is present). This results in the anomalous or Bijvoet difference $\Delta|F_{ano}|$ which can be computed using equation 8. This is used to identify the position of the anomalous scattering atoms.

$$\Delta|F_{ano}| = \{|F_{PH}(+)| - |F_{PH}(-)|\}\frac{f'}{2f''} \quad (8)$$

(Taylor, 2003; Drenth, 1999)

where $|F_{PH}(+)|$ is the amplitude of reflection (hkl) and $|F_{PH}(-)|$ is the amplitude of the reflection $(\bar{h}\bar{k}\bar{l})$. hkl and $\bar{h}\bar{k}\bar{l}$ are Bijvoet or Friedel pairs. The Bijvoet difference $\Delta|F_{ano}|$ is scaled up with f'/f'' where f' is the dispersion term and f'' is the absorption term of the atomic scattering factor.

This approach overcomes the limitations related to the non-isomorphism of the method isomorphous replacement. However, the changes in the amplitudes of the anomalous scatterers are generally small. This necessitates the accurate measurement of intensities. The radiation damage to the crystals is another limitation of this method.

Molecular Replacement Method

This method was developed by Michael Rossmann and David Blow (Rossmann & Blow, 1962) and can be used when a homologous structure is available for the structure to be solved. For this the sequence identity of the homology model $\boldsymbol{x_{model}}$ (known molecule or model), and structure to be determined $\boldsymbol{x_{unknown}}$ (unknown molecule) should be usually > 25% and the r.m.s. deviation should be < 2.0 Å between the αC atoms of the known and the unknown molecule (Taylor, 2003).

In molecular replacement, the phasing is performed principally in three steps. In the first step, the known molecule is rotated $\mathbf{R}$ in three dimensions. And the orientation in which the calculated structure factors of the known molecule gives best agreement with the observed structure factors of the unknown molecule is selected. In the next step, the model is placed at every position of the unit cell and the model is translated $\boldsymbol{t}$ to the position that gives the best agreement between structure factors. In the final step, the phases from the model and the weighted structure factors of the unknown molecule are used to compute the electron density map of the unknown molecule. The complete process, equation 9, is illustrated in Figure 5.

$$\boldsymbol{x}_{unknown} = \mathbf{R}\,\boldsymbol{x}_{model} + \boldsymbol{t} \tag{9}$$

(Taylor, 2003)

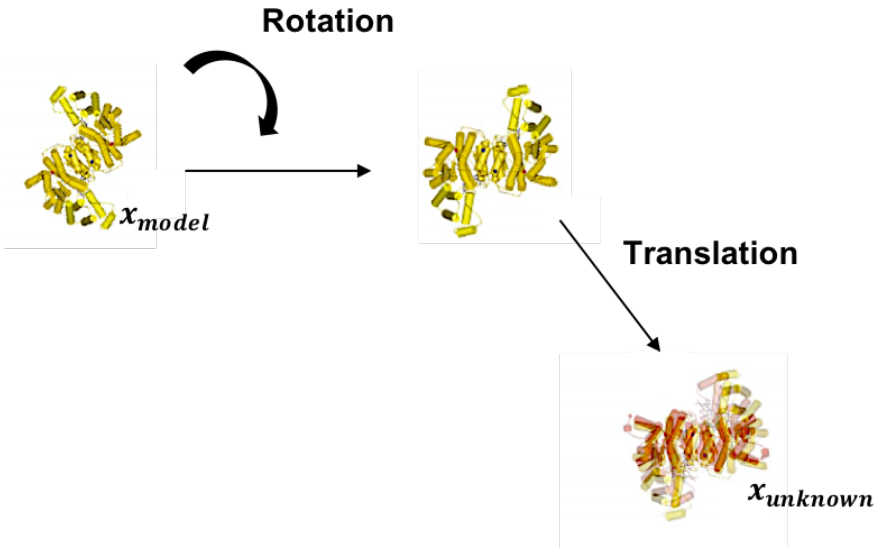


Figure 5 Molecular Replacement (Taylor, 2003). The model (shown in yellow colour) is rotated to get the desired orientation and then translated to the position of the unknown molecule (shown in red colour).

The major limitation of this method is error in the phase computation due to model bias. Model bias is explained visually in Figure 9.

The initial phases obtained by these methods may not always be accurate due to a variety of reasons. These include errors in the computation of structure factors, errors in the calculation of atomic positions of heavy atoms or anomalous scatterers, low

resolution of the X-ray data due to flexibility of proteins, non-availability of a synchrotron, model bias and lack of experienced crystallographers. Therefore, these phases are rarely sufficient to interpret the electron-density map accurately, and therefore need further improvement and refinement.

How to Improve the Correctness of Phase Values?

Phase improvement methods

Various phase improvement methods have been developed to improve the initial phases. These are mainly sub-divided into two categories:

1. Density Modification
2. Automated model building and refinement

Density Modification

Density modification methods focuses on improving the electron-density map by exploiting the prior knowledge available in describing the desired features expected in these maps. These methods are further sub-divided into:

1. Classical density modification
2. Statistical density modification

Classical Density Modification

The classical density modification works by iteratively cycling between real and reciprocal space. The process starts with the inverse Fourier transformation of an electron-density map to obtain improved phases, followed by combining these phases with the experimental phases to compute a new map. This procedure (Figure 6) is then iterated until reaching convergence (Cowtan, 2010).

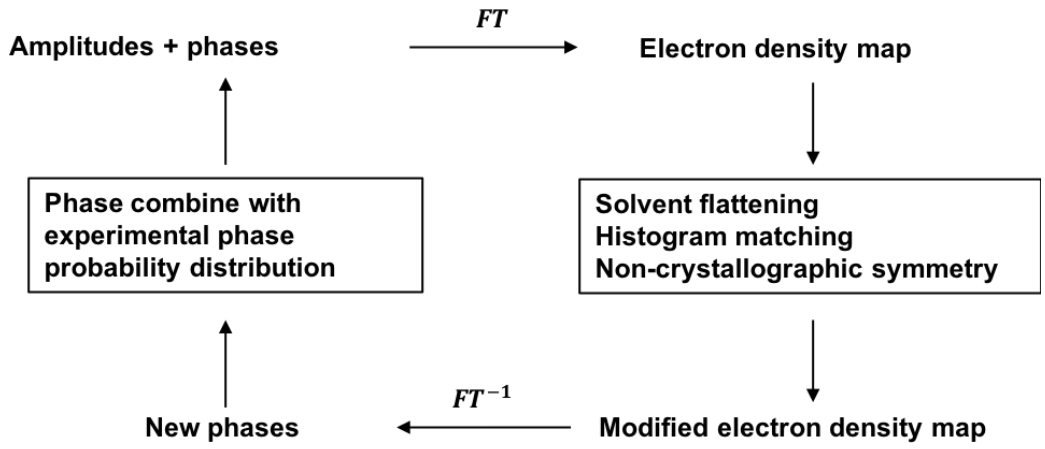


Figure 6 The classical density modification cycle.

The three most popularly used methods based on different prior structural knowledge used are:

Solvent Flattening Solvent features are modified based on the assumption that the variation of electron density of the solvent region is low (Figure 7a). Hence, the phase combination that gives more flattened solvent map is considered correct and used in the computation of the modified electron density map (Wang, 1985).

Histogram Matching The electron density histogram of a good map has a characteristic shape. A map with any deviation from this shape is considered as a badly phased map (Figure 7b). This method works by rescaling this badly phased map to make it look closer to the histogram of the well phased map (Zhang & Main, 1990).

Non-Crystallographic Symmetry (NCS) Averaging If a crystal contains several copies of an identical or nearly identical molecule in the asymmetric unit, the electron densities of these molecules are averaged to get better signal-to-noise ratio and to improve the phases by imposing better restraints (Rossmann & Blow, 1963) (Figure 7c).

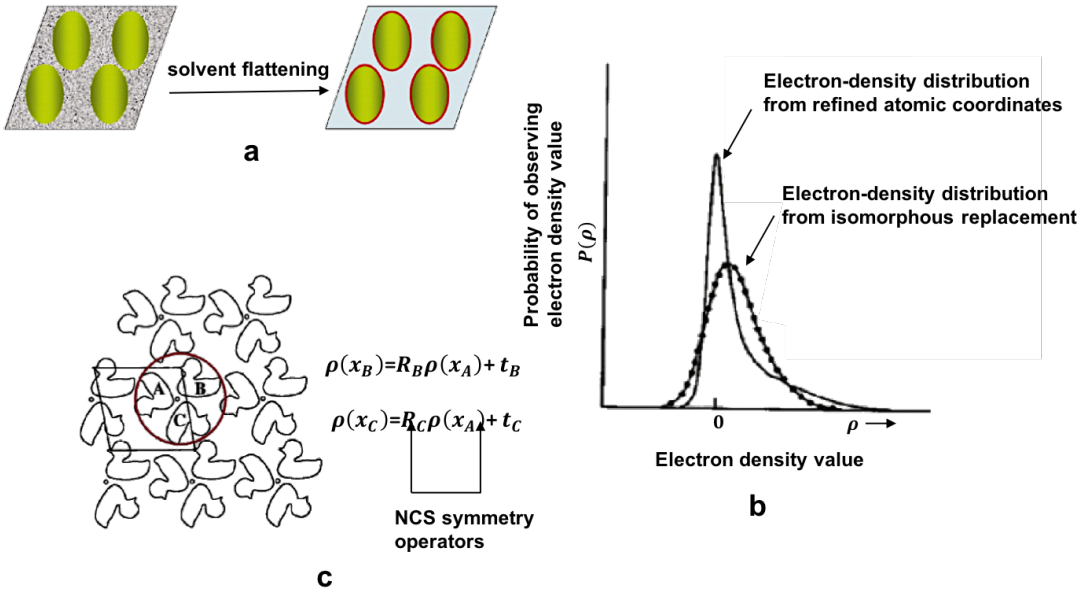


Figure 7 Classical density modification methods (Taylor, 2003). Figure 7a shows solvent flattening methods in which the solvent region (black background) is flattened (light blue background) after defining the border (shown in red) between the solvent and the protein (green ovals). Figure 7b shows a histogram of electron density obtained after isomorphous replacement (dotted black curve) overlapping the histogram of the well phased map (solid black curve). Figure 7c shows several copies of ducks and the NCS operators (equations on right) relating the data of duck B and C to A. Ducks A, B and C are encircled in red in left.

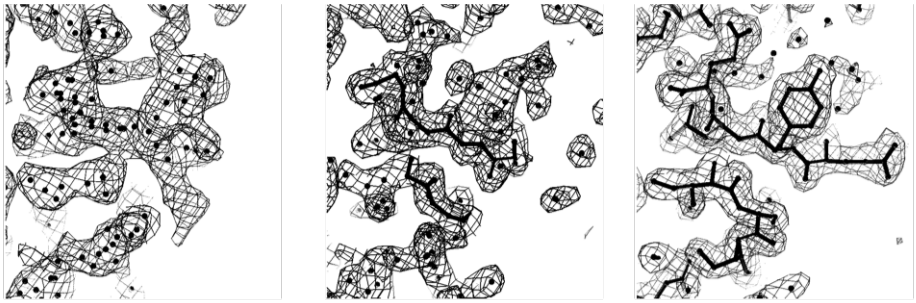
Statistical Density Modification

The classical density modification methods might introduce some additional errors in the phase computation due to the bias introduced by the prior knowledge used. This can be overcome by using statistical density modification, which uses additional information based on the probability distributions and provides a very weak link between this information and the initial phases in order to avoid bias. However, the major disadvantage with this method is the heavy computational overhead (Cowtan, 2010).

Automated Model Building and Refinement

Due to many computational or experimental errors usually associated with the low-resolution data, the phases may need further improvement and refinement even after

using density modification methods (Figure 8). Therefore, further phase improvement is performed as a part of automated model building and refinement.



Solvent flattened map **Intermediate map** **Map after automated model building and refinement**

a **b** **c**

Figure 8 Map improvement after density modification. Figure 8a shows the solvent flattened map with less interpretable density which has improved in the intermediate map (Figure 8b), showing interpretable backbone density. The final map (Figure 8c) shows well defined backbone and side chain density after using automated model building and refinement. Maps generated using ARP/wARP software after model building and refinement of protein Leishmanolysin (PDB ID:1LML).

The Significance of the Accuracy of Phases

The poorly phased electron-density obtained from low-resolution data, with associated errors may not be improved even after using phasing and phase improvement methods, which may lead to incorrect interpretation of the model (Figure 9). Further phase optimisation may require a robust algorithm that samples possible phase values and employs an efficient mathematical machinery to choose correct phases.

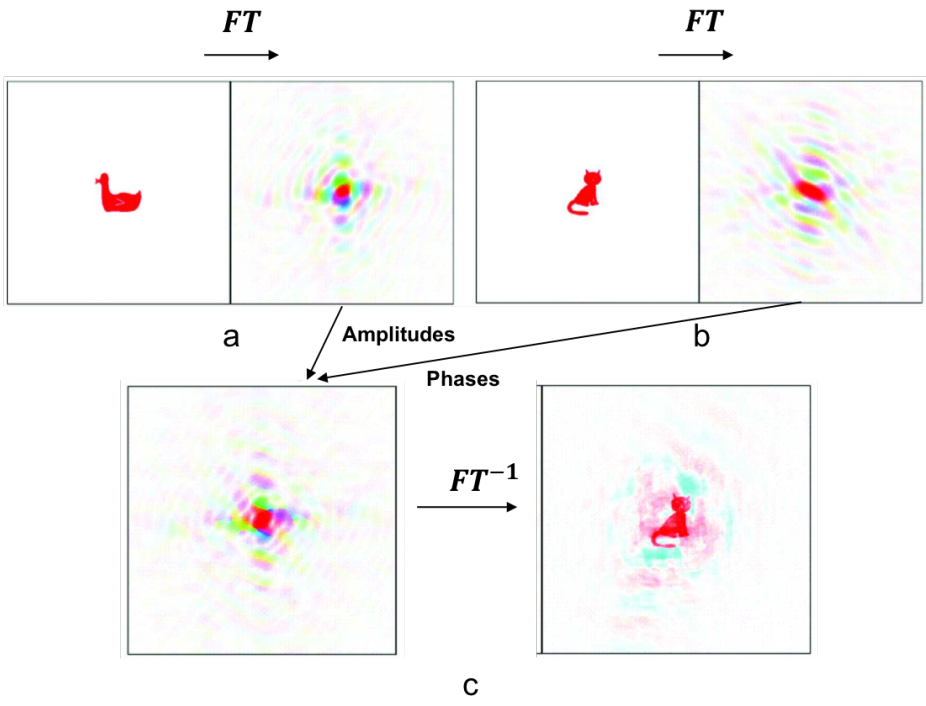


Figure 9 The effect of phases on the correctness of the structure (Taylor, 2003). Combining the amplitudes from the duck (shown in right of Figure 9a) and phases from the cat (shown in right of Figure 9b) results in a hybrid diffraction pattern (shown in left of Figure 9c) with more features from the cat (shown in right of Figure 9c) after inverse Fourier transform.

The Phase Optimisation Problem

The sampling and optimisation of phase space is difficult, as it is highly multimodal (with various local minima) and multidimensional (proportional to the number of reflections). The use of a random walk for the optimisation of such objective function may be convenient but it often converges to different solutions depending on what is defined as random, while a gradient-based method may take unacceptably long to identify the correct solution (Figure 10). More on the sampling and optimisation methods is described in chapter 2.

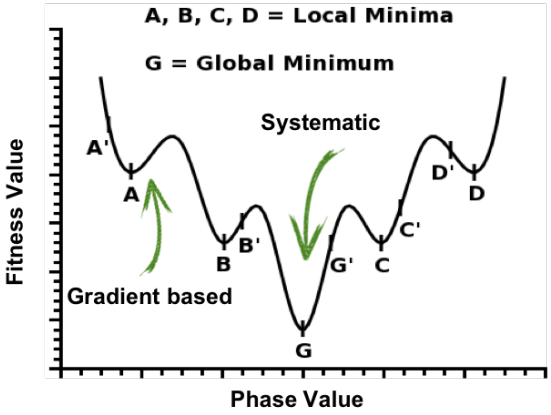


Figure 10 Phase error landscape. The figure showing various local minima (A, B, C and D) and global minimum G. Most gradient based methods may get trapped in a local minimum while systematic methods are costly in terms of computational time required to the global minimum.

Genetic algorithms, a heuristic optimisation technique, have a certain advantage in such situations because of its 'jumpy behaviour', allowing efficient sampling of the entire search space.

Apart from the complexity of the phase space, another factor limiting the efficient sampling of the phases in real space is the unavailability of the reliable objective function that could relate to the relationship between phase error and characteristics of electron density map and provide a useful estimate on the quality of phases. There were some studies performed to identify such an objective function and showed that the higher order moments, skewness and kurtosis, can be used to monitor the quality of the electron density map (Cochran, 1955; Podjarny & Yonath, 1977; Lunin, 1993). Further details on these moments are described in the "Fitness function" section of chapter 3.

This thesis is an attempt to address these issues: phase improvement given an initially high phase error (usually associated with low-resolution data) using genetic algorithms with a minor focus on providing insights towards identifying a reliable objective function. Therefore, a literature survey on the nature of genetic algorithms, their behaviour, implementation, and performance is presented in the next chapter.

Chapter 2

Introduction to Genetic Algorithm

What are genetic algorithms?
Why use genetic algorithms?
When to use genetic algorithms?
How do genetic algorithms work?
Can we use genetic algorithms for phase optimisation in crystallography?

In the 1940s, early computer scientists envisioned (Turing, 1950; Neumann, 2017; Heims, 1980) developing computer programs that have life-like abilities; programs that can self-replicate and have an adaptive capability to understand and control their environments. These biologically motivated computing activities developed into fields such as neural networks, machine learning and evolutionary computation. Among these, evolutionary computations (Bäck *et al.*, 1991; Zitzler & Thiele, 1999) that are inspired by genetic variation and selection are far more appealing in solving many computational problems for a variety of reasons.

Why use algorithms inspired from evolution? The programs based on evolutionary computation are *adaptive* and *innovative* in nature; an ability to adjust to a changing environment without compromising in performance while producing something new. They can also accommodate computational parallelism allowing efficient use of modern computing facilities to solve higher order search problems. Many complex biological optimisation problems - for example finding a correct conformation of a small molecule in drug discovery - need such an intelligent system.

Genetic algorithms (GAs), invented by John Holland in the 1960s (Forrest & Mitchell, 2016) belong to this class of evolutionary algorithms. Holland, his students, and colleagues from the University of Michigan have extensively studied the phenomenon of nature's adaption and provided ways to imbibe this intelligence into computer programs. His book "*Adaption in Natural and Artificial systems*" (Holland, 1975) with theoretical foundations on GAs, is a pioneering work in the field of evolutionary computation. Over the past few decades, GAs have evolved as an interdisciplinary field sharing vague boundaries with other evolution-related approaches such as

evolutionary strategies and programming, using overlapping terminology and concepts.

Genetic and Evolutionary Terminology

With more than eight decades of research, GAs and evolutionary computations have emerged into an independent field acquiring distinct terminology generic to their class of computations. These terminologies are extensively used in this work. Hence, it is helpful to define them at this point.

GAs are inspired from natural selection and mimic many processes related to biological reproduction and propagation. In living organisms, *genes* are made of DNA - a blueprint of an organism. Each gene is responsible for a specific *trait*, for example, hair colour of an individual. Different possibilities of a *trait* (feature) for a gene are called *alleles* of that specific gene - for example, black and white are possible alleles for the trait hair colour. Many genes collectively form a *chromosome* and the position of a gene on this is known as the *locus*. The complete collection of genetic material - all chromosomes together - is a called a *genome*. A particular set of genes in a genome is referred as the *genotype.* A genotype results in a phenotype – physical and mental characteristics of an organism such as height, hair colour and intelligence.

A chromosome in most sexually reproducing organisms, for example humans, are arranged in pairs, a state called *diploid*, while some organisms have unpaired chromosomes, a state called *haploid.* After haploid or diploid reproduction, an *offspring* produced from its parents collectively forms a *population* together with its parents. The *fitness* of this offspring is defined in terms of its *viability* (an ability to live and reproduce) and/or *fertility* (the number of offspring it produces). A pictorial depiction of differences in DNA, genes, chromosomes are shown in Figure 11.

Most of the biological terms used in GAs have similar or equivalent meaning to their original use in biology. GAs do not work at the DNA level i.e., most representations only adapt concepts such as genes, chromosomes, and their corresponding phenotypic expression. A *solution* to the problem is called a *member* or an *individual* at a phenotypic level. All members at any given point collectively form a *population*. A

chromosome in a GA is a genotypic representation of candidate solution for a search problem and mostly haploid in nature. Each gene *encodes* a specific element of the solution. Often genes are represented as single bits, where an allele can be either 0 or 1. Offspring are produced by recombining haploid chromosomes using crossover events. Mutations are used to change a specific gene at a selected locus. An intelligent strategy, encoded as a mathematical objective function known as the *fitness function* is used to identity the fitness of an individual. GAs usually works at a genotypic level and often do not have a notion of phenotypic level. In simpler implantations of GA, the *fitness* of an individual usually refers to its viability while some robust implementations of GA consider both fertility and viability of an individual.

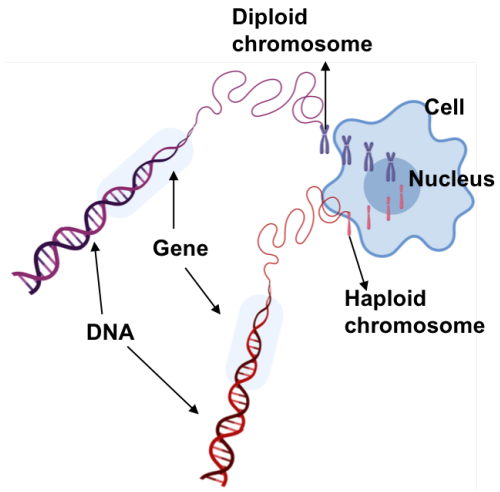


Figure 11 Diploid and haploid chromosomes, genes and DNA. Cell and nucleus in the figure are shown as entities without any genetic typing i.e., neither haploid nor diploid.

What Type of Problems is GA Best Suited to Solve?

Search Spaces and Fitness Landscapes

The suitability of the GA for a specific problem can be defined based on the nature of a *search space* and the *fitness landscape* of that problem. These concepts and the related terminology are described here.

In computer science, the concept of searching has three overlapping meanings: search for stored data, search for paths to goals and search for solutions (Mitchell 1998). In

GA, the term search often refers to the search for a correct solution in a pool of possible solutions ("search space") for a given problem. A search space contains all candidate solutions to a problem with some positional notion in terms of "distance" between them.

Another important concept commonly used in GA is "fitness landscape", a term derived from population genetics which can be defined as a representation of all possible genotypes and their fitness values (Wright, 1931). These landscapes, just as any physical landscapes with characteristic "hills", "peaks" and "valleys", can be visualised as a pattern of crests, troughs and flats formed by the collection of genotypes and their fitness values. Graphically, this can be imagined as a $l + 1$ dimensional plot in which each genotype is a point in l dimensions while $l + 1^{\text{th}}$ dimension corresponds to the fitness value of these genotypes (Mitchell 1998). The process of evolution can be seen as a movement of the population along these landscapes while the adaption can be seen as a process that drives the population towards a "local peak" ("local optimum"). A local optimum is a point in the landscape where any slight deviation from it results in reduction of the population fitness. This is not necessarily the highest point ("global optimum"). Depending on the context, an adaption can be a process of movement towards a local minimum or a trait acquired during the process that helps the organism to survive in its environment. If it is not specified as a trait, the word "adaption" in this work means a process.

The distribution pattern of the population can be discussed in terms of what is called *diversity*. The diversity of the population at any point is: the distance from the starting point and distance between themselves. Diversity from the starting point can be seen as the distance travelled by the population in a landscape and shows that the population is evolving. Diversity within the members of the population can give an idea on how close the population is to convergence (Figure 12a). The population is said to have "converged", if the members are similar to each other genotypically. If they have converged to the global optimum, the process is seen as convergence in the right direction (Figure 12b). The convergence before reaching the global optimum is called "premature convergence" (Figure 12c). Other applied terms are described wherever appropriate.

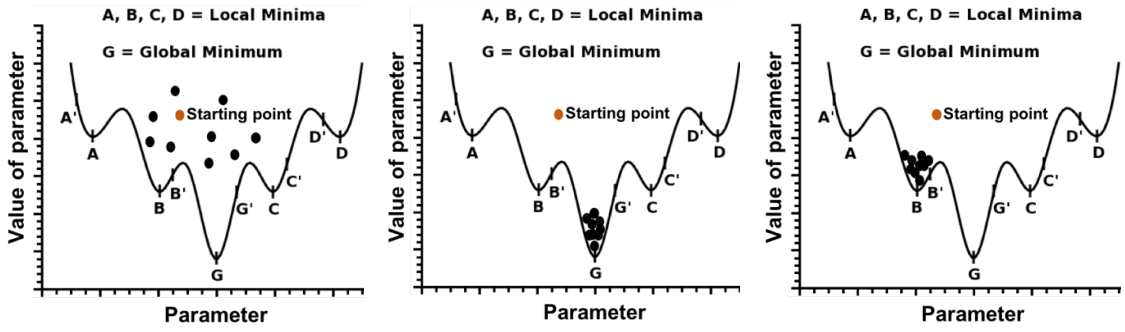


Figure 12 Schematic representation of the fitness landscape. (a) Initial population members (black dots) with high diversity, close to the starting point (red dot) and far from convergence. (b) Population converged to a global minimum, G. Members (black dots) have much less diversity, far from the starting point (red dot) and converged to correct minimum. (c) Premature convergence. Members (black dots) have very less diversity, far from the starting point (red dot) but converged to local minimum, B.

What type of problems are GAs best suited to solve? GAs are used when the search space is very large, not well defined, not smooth and multimodal (with many hills), and/or if the fitness function is noisy and in the type of the search problems where the information is not stored explicitly but the possible solutions has to be created on the fly as the process proceeds. If the space is smooth or unimodal, gradient-based methods are much more efficient than GA. If the space is well understood, domain specific heuristics can be applied. Most implementations of GA aim at finding good enough solutions in a relatively short time ("statisficer") using optimal computational resources, rather than finding the global optimum.

The expectations based on these cannot be considered as check markers to define the efficiency of the GAs, as its performance is also largely dependence on the choice of different parameters and finding the correct balance in their implementation. The best balance of the parameters can be seen as an interplay between "exploration" and "exploitation". The search of new and useful adaptions (traits) is exploration, while use and propagation of these adaptions (traits) in a population is exploitation. In the Mitchell (1998) formulation of exploration and exploitation balance, the system has to keep trying out new possibilities, but it also has to continually incorporate and use past experience as a guide for future behaviour. Otherwise it might "overadapt" or become

inflexible when facing novelty. Understanding how GAs process intrinsically and extrinsically and gaining knowledge on how to achieve the exploration and exploitation balance might be helpful in obtaining the desired goal of optimisation (global optimisation or satisficer) with GA.

How do GAs Intrinsically Work?

A Schema Theory

The behaviour of GAs can be very complex. It might be hard to understand in what type of problems they perform well. Developing GAs without establishing a way to learn how they work intrinsically might lead to a poor design where even a random mutation-based search can perform better than them (Jong, 1993). Holland (1975) introduced a framework and the term *schema* for describing the behaviour of GAs.

A schema in this theory is a constellation of genes that are responsible for a specific adaption (trait). Computationally, any schema is a template that has fixed positions and "don't cares". For example, in a chromosome with bit values, the schema H = 1***1 stands as a template for all 5-bit strings that start and end with 1 and the bits between these two bits can be of any value (hence called don't cares and represented as *'s). Goldberg (1989*c*) called this template H a *hyperplane.* The strings 10011 and 11001 that fit to this template H are called *instances*. The *order* of any schema is equivalent to the number of the fixed (in the example above, number of ones) positions. Any short, low-order schemas with a fitness above the average are called *building blocks*.

The schema theory or building block hypothesis states that, during recombination, a good genetic algorithm combines good building blocks using crossover to form better solutions (Holland, 1975; Goldberg, 1989*c*; White, 2014). In the selection process, GAs calculate the fitness of the population explicitly based on the average fitness of many schema calculated implicitly, which are in turn are calculated implicitly based on the average fitness of all instances possible to these schemas (Figure 13). This property is called "implicit parallelism" by Holland. Implicit calculation of instances and schemas takes the same computational time without needing any additional storage, making GAs more robust than many other optimization algorithms. The performance of selection, crossover and mutation are often described based on this theory. Most of

the works on this theory were presented based on the static assumption called “Static Building Block Hypothesis (SBHH)” that states: a GA will converge on actual winners of each short, low order partition competition but not on the schema with the best fitness (Grefenstette, 1993). Assuming this static nature, schema theory is widely criticised by many researchers (Peck & Dhawan, 1993; Mason, 1993). However, Mitchell *et al.*, (1993) proposed more dynamic approach on schema processing in GA, proving the relevance of the schema theorem. More on this is described in the section, “What Makes a Problem Hard for a GA?” of this chapter.

In Holland’s view, selection searches for the schemas with estimated above average fitness, crossover brings together high fitness building blocks to form a chromosome with increased fitness, and mutation works as an “insurance policy” in preventing the loss of genetic diversity at any locus (Holland, 1975).

Figure 13 Illustration of different levels of processing genetic information in GA. All these steps are processed in parallel, a property called implicit parallelism of GA.

What Makes a Problem Hard for a GA?

A Deceptive Fitness Function and Hitchhiking by Crossover

When using GAs, one might conclude that they do not work, or that other random search methods perform better. The following theoretical foundations provides a way to understand why most GA implementations fail in achieving what other random search methods can do. The primary reason for this could be the use of deceptive fitness function. As described earlier in this chapter, a “fitness function” is a mathematical function used to calculate the fitness of an individual. According to Bethke (1980), GAs cannot find an optimum of fitness function if the low-order

partitions have incorrect information about high order partitions. For example, a schema is considered a winner when all its defining bits (fixed positions) are ones (11….111) except for the schema of length l, where a schema with all zeros (00….000) is considered a winner. In this case, GAs cannot find this higher-order partition with all zeros as every low order partition picks misleading instances as a winner. Fitness functions that propagate such behaviour are termed "deceptive" (Deb & Goldberg, 1993; Whitley, 1991). Using "Walsh Transforms" (Goldberg, 1989*b*,*a*) similar to Fourier transforms, Bethke, (1980) presented different designs of fitness functions with varying degrees of deceptiveness.

The deceptiveness of fitness functions decides whether GAs can be used as function optimisers (finding the global optimum) or as satisficers (finding a good enough solution). With a highly deceptive fitness function, it is impossible to reach the global optimum. But Grefenstette (1993)'s work proved that the deceptiveness, which is based on the SBBH, is not necessarily a culprit that creates trouble for GAs and may not be capable of such. More dynamic hypothesis based on the "Royal Road Functions" experiment performed by Mitchell *et al.*, (1992, 1993) to study the principle of building blocks in an idealised form, showed that difficulty for a GA comes from "Hitchhiking" or "spurious correlation" of schemas. "Hitchhikers" are the bits that are not part of the desired schema but tag (or propagate) along with the schema by being next to it on the string (Schaffer *et al.*, 1991; Schraudolph & Belew, 1992). The types of the crossover that propagates hitchhikers are discussed in the section "Parameters of GA" of this chapter.

How to Implement a GA?

Components of a GA

The common implementation of a GA (Figure 14) is as follows:

1. Map the starting candidate solution to a problem on a chromosome. Randomly mutate the chromosome of length l to generate n members.
2. Compute the fitness $f(x)$ of each chromosome x in the population. Select the desired number of offspring from the population using fitness function. These selected population members are considered parents.

3. Randomly select a pair of parents for crossover. Recombine the selected pairs of parents with a probability of p_c until the desired number of offspring are produced.
4. Mutate the offspring with a probability p_m and place the mutated chromosomes in the population.
5. Replace the current population with the newly generated population.
6. Go to step 2.
7. Stop if the population has a desired fitness (average/max) but otherwise continue.

Each iteration from step 1 to step 5 is called a *generation*. The complete set of generations required to obtain the desired solution is collectively called as a *run*.

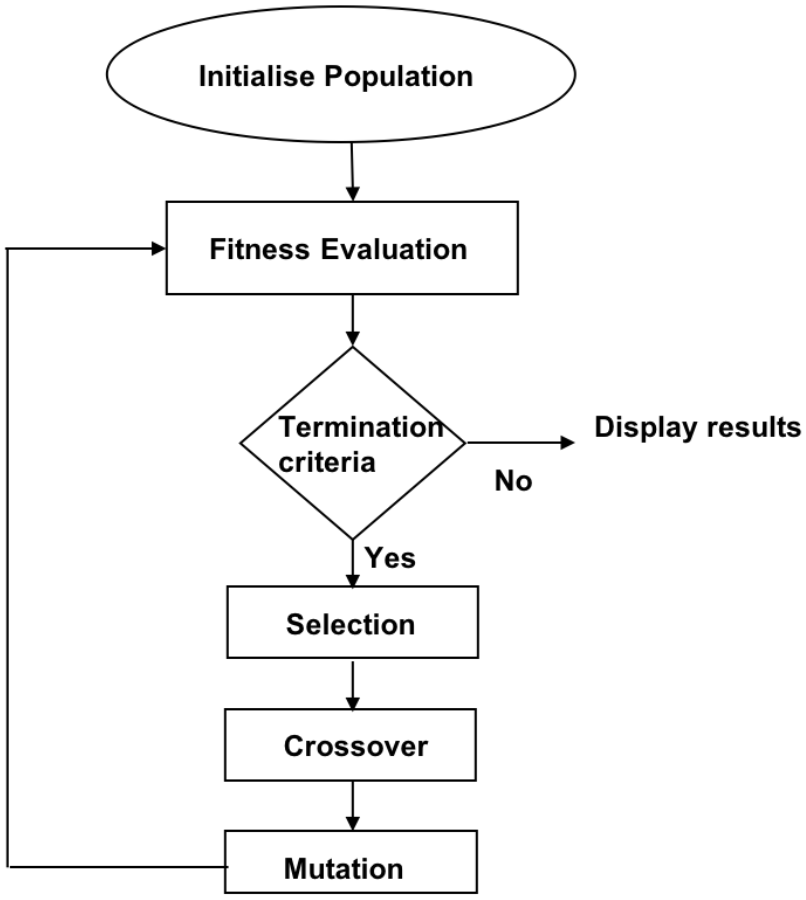


Figure 14 The outline of the implementation of Genetic Algorithm.

Parameters of GA

Genetic Operators

In GAs, every step in Figure 14 is achieved by mimicking a specific biological phenomenon involved in evolution. The term “operator” in a GA is used to describe such a phenomenon. While we have many such operators (phenomena) that can be studied, most applications use a few popular ones. The original implementation of a GA by Holland used four operators: selection, crossover, mutation, and inversion (Holland, 1975). Many recent versions of GAs exclude inversion, as it does not improve the performance notably (Goldberg, 1989*c*). GAs process a large number of schemas intrinsically, while how and what type of schemas will gain priority and be allowed to propagate is dictated by the genetic operators such as selection, crossover and mutation. A brief overview of the different steps of GAs is mentioned in Figure 13 and is discussed below. This includes a description on various operators.

Step 1 Initialisation

This step includes mapping or encoding the starting point of the search problem onto a chromosome, and generating a set of chromosomes (population) by mutating few of its genes. As mentioned earlier, chromosomes in GAs are usually haploids. The genes on these haploid chromosomes can contain any value; real or bits. Different encoding methods (Kumar, 2013) that can be used are explained below:

Binary Encoding In binary encoding, each chromosome is a string of bits with value either 0 or 1 (Figure 15).

Bits	1	0	1	1	1	0	1	1	1

Figure 15 A chromosome showing binary encoding with bits or genes having a value of either 1 or 0.

This is common and the very first encoding type used in early GA implementations. It has limited applications as it can only be used in the problems where a gene value represents either presence or absence of a certain parameter.

Permutation Encoding It is commonly used in ordering problems (e.g. Travelling salesman problem), where the task is to find the minimum distance given all the cities

and distance between them (Figure 16). A gene value in this encoding contains the order or a step number information of the problem.

Order	8	6	3	9	4	2	5	7	1

Figure 16 A chromosome showing permutation encoding with chromosome presenting the order of cities a salesman will visit.

Value Encoding In this encoding, a gene value takes values such as real numbers, characters connected to the problem (Figure 17). This encoding is good for special problems where alleles of genes cannot be represented as bits, simple ranks or orders.

Real	45.67	67.46	90.59	83.20	24.39
Character	A	D	F	A	F
Parameter	north	south	west	east	south

Figure 17 Figure showing three chromosomes with real, character and parameter encoding.

Tree Encoding This encoding is used in programs or expressions that are evolving during optimisation such as determining a function from a set of given values. Every chromosome here is a tree of objects. These objects usually are functions or commands in a programming language (Figure 18). A programming language called LISP is used for this type of encoding.

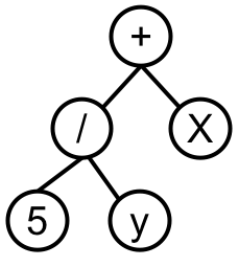


Figure 18 A chromosome with tree encoding of set of functions.

The next step after encoding a solution on a chromosome is to generate a set of chromosomes by mutating its genes. The initial set of chromosomes generated act as

parents for reproduction after succeeding in the selection process. Generating initial populations with enough diversity ensure that populations converge to the global optimum instead of a local optimum (Hillis, 1990). The effect of diversity in the initial population can be understood from the visualization in Figure 19. In Figure 19a, the initial populations are very close to each other in space (low diversity) and have a higher chance of converging to a local minimum, B, in a relatively short time compared to Figure 19b. In Figure 19b, the initial population are well distributed in the search space (high diversity). An optimal design of GA with good fitness function might drive towards the global minimum (point G in Figure 19a/19b). But convergence in this case takes more time compared to Figure 19a.

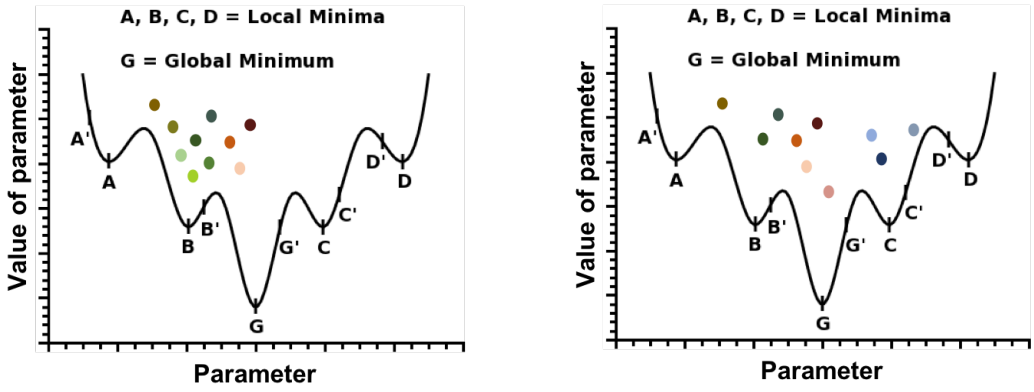


Figure 19 Visualisation on the effect of diversity of initial population on convergence. The colour and the tint of points indicate diversity and spatial distance between them respectively. In Figure 19a, points have less diversity within them and crowded near local minimum. This initial population might lead to premature convergence. In Figure 19b, points are diverse enough with representations from multiple minima. In this case, a good algorithm might drive towards the global optimum.

Step 2 Selection

Selection, based on the Darwin's theory of evolution, ensures that the fittest individuals (survivors) are taken to the next generation and participate in further recombination. In the selection process, the population members are scored based on the chosen fitness function, and those with better scores have higher probability to *survive*. Selection modifies the current fitness distribution of a population into a new distribution. A very strict selection may result in insufficient diversity of the population, which is needed for

further evolution. On the contrary, weak selection may reduce the convergence speed (Mitchell, 1998). The performance of different selection operators can be evaluated (Blickle & Thiele, 1996) in terms of the following:

A. reproduction rate, the ratio of the members of the population with a certain fitness value after and before selection,
B. loss of diversity, the proportion of members of a population removed during the selection phase.
C. selection intensity, the expected average fitness value of the population when the selection method is applied.
D. selection variance, the normalised expected variance of the population's fitness distribution after selection method is applied.

Different selection operators commonly used are as follows:

Proportional Selection This is a very first method proposed by Holland (1975). The concept of this method is similar to rotating a roulette wheel. The fitness of the population is mapped on a scale similar to a roulette wheel. And this wheel is rotated N times to get N survivors. This means that the higher the fitness of an individual, the greater probability of getting selected (Figure 20).

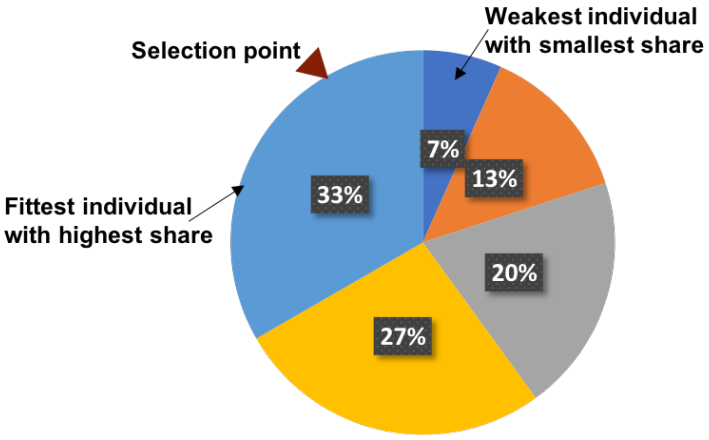


Figure 20 Fitness proportional or roulette wheel selection. A member in the light blue zone (33%) has a high probability of getting selected compared to others. This wheel is rotated with a fixed selection point for N times to get N survivors.

A serious disadvantage of this method is its strong dependence on the scaling of the fitness function and its non-applicability for population with members having zero fitness. This selection has very high *bias* and too low selection intensity. A bias in terms

of selection functions can be defined as a deviation between the expected reproduction rate and the algorithmic sampling frequency (Blickle & Thiele, 1996).

Stochastic Universal Sampling (SUS) This method (Baker, 1987) is similar to proportional selection except that it spins the wheel once and selects N members at an N evenly spaced point from the starting pointer of this spin (Figure 21).

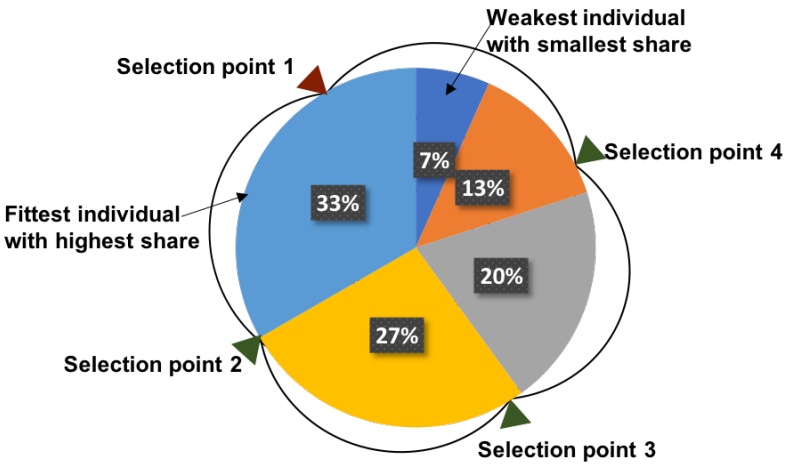


Figure 21 SUS showing members mapped on a roulette wheel with percentage equivalent to their fitness respectively. The marker in brown is the member at a random starting point. After selecting this point, the next points (markers shown in green) are selected at an evenly spaced interval.

The problem of the bias with proportionate selection can be overcome by using SUS. This selection stated as optimal sampling algorithm (Blickle & Thiele, 1996) also has minimal spread of the range of possible fitness values.

Linear Ranking Selection For this selection, the members of the population are sorted based on their fitness value. Each member is assigned with a rank between 1 and N where 1 indicates worst fitness and N indicated best fitness (Baker, 1985; Grefenstette & Baker, 1989; Whitley, 1989). The selection probability is assigned linearly to members based on their ranks and no member gets a similar rank having similar probability of selection even if they have same fitness value.

When the fitness function is noisy, incorrect selection might lead to premature convergence. Ranking helps to overcome this problem. However, this is not the best method for problems where it is important to know that one individual is much fitter

than its competing neighbour. When the fitness variance is high, ranking reduces the selection pressure by not giving the largest share to a small proportion of fitter individuals. When the fitness variance is low, the members i and $i+1$ have the same chance irrespective of their high or low absolute fitness differences.

Exponential Ranking Selection This is similar to linear ranking except that the probabilities of the ranks of the members are weighted exponentially (Blickle & Thiele, 1996).

Truncation Selection In truncation selection (Crow & Kimura, 1970; Mühlenbein & Schlierkamp-Voosen, 1993) the fraction of best performing members of a population above the threshold T are all selected as having same selection probability and the rest are discarded.

Tournament Selection This selection (Blickle & Thiele, 1995; Goldberg & Deb, 1991) works (Figure 22) as follows:

1. Choose t number of individuals randomly from the population
2. Copy the best individual from this group of t members to survivors
3. Repeat N times to get N survivors

This selection method is better than ranking and fitness proportionate selection, as these require more computational time than tournament. It has a similar performance to rank selection in terms of selection pressure. Another advantage of this method is its feasibility in allowing parallel implementation.

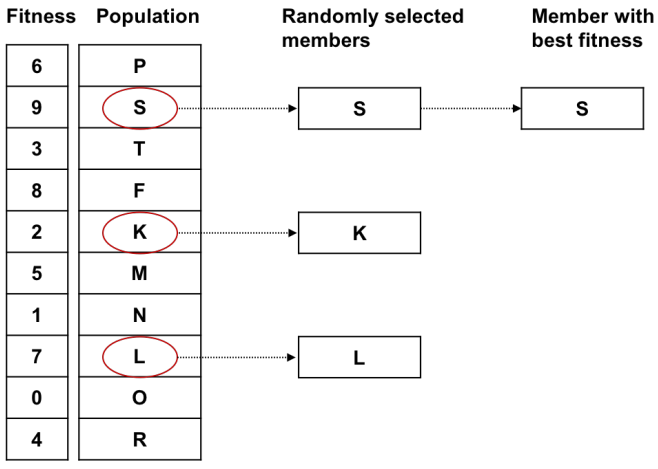


Figure 22 Visualisation of different steps in a tournament selection operator.

Elitism This is often used as complementary to other selection methods. The idea is to pass on certain percentage of best individuals at every generation (De Jong, 1975) to avoid the loss of these individuals due to selection pressure.

Blickle & Thiele (1996) extensively studied the behaviour of tournament, truncation, linear ranking and exponential ranking selections in terms of reproduction rate, loss of diversity, selection intensity and selection variance. The loss of diversity was found to be very high in truncation, moderate in tournament, but very low in exponential selection. Selection variance reported to be very low in truncation compared to tournament. Exponential ranking had the highest selection variance. These studies also showed complementary performance of a binary tournament compared to a linear ranking selection, in having a small fraction of members with worse fitness.

Step 3 Crossover

A crossover (also called recombination) is the process of combining parental chromosomes to produce offspring (or children). A crossover has the ability to combine high fitness-low order schemas to form high fitness-higher order schemas, "constructive power" (Thierens & Goldberg, 1993; Spears, 1993) as described in the Building Block Hypothesis (Goldberg, 1989*c*). A crossover is also studied for its schema "destructive" power (Sastry & Goldberg, 2002; Blickle & Thiele, 1996). These studies provided detailed descriptions of different crossover operators and their roles as a mixer, schema disruptor and innovator that are analogous to the concepts of constructive and destructive power of a crossover. These roles of the crossover are important in the selection of the appropriate crossover type.

The performance of a crossover can be further evaluated based on the following properties (Eshelman *et al.*, 1989):

1. *positional bias,* the dependence of the probability that a set of genes will be transmitted together depending on the relative positions of those genes on the chromosome. This preferential treatment might prevent some high fitness-low order schemas in generating new higher order schemas with high fitness.

2. *distributional bias,* the number of genes transmitted during a crossover and the probability that these genes are more likely transmitted than others. The higher the distributional bias, the higher the diversity and schema (related genes) disruption rate.

In addition to these, at a genotypic level the nature of crossover can be studied in terms of *recombinative bias* and *schema bias* (Sastry & Goldberg, 2002; Senaratna, 2005).

Table 2 Comparison of performance of different crossover operators (Sastry & Goldberg, 2002).

Crossover operator	Positional bias	Distributional bias
One point	high	low
Two or k- point	medium	medium
Uniform	low	high

Various crossover operators available for different applications are presented below:

Single-Point Crossover In a single-point crossover, a random *locus* (crossover point) is selected and genes after this locus to the end of the chromosome in the chosen direction are exchanged for a crossover between two parents (Figure 23). This crossover cannot mix all possible schemas ("positional bias"), e.g., it cannot combine 11***1 and ***1*1 to give 1**1*1, and treats end points preferentially ("endpoint effect"). Moreover, it tends to propagate hitchhikers.

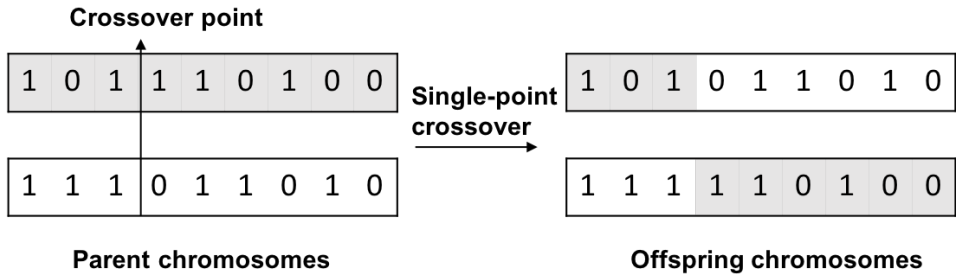


Figure 23 Single-point crossover. Figure showing an exchange of genes between two parent chromosomes at crossover point resulting in two offspring with new genetic makeup.

Two-Point Crossover In a two-point crossover, two *loci* (crossover points) are selected randomly and the genes between these two loci in the parent chromosomes are exchanged to form two offspring (Figure 24). This crossover has less disruptive power and can mix more schemas than a single-point crossover.

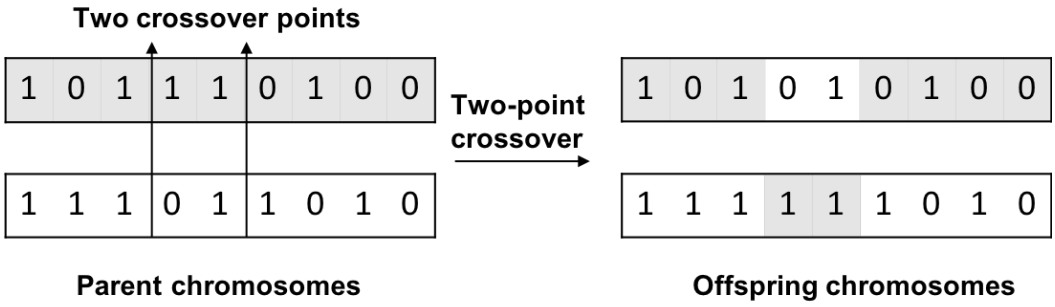


Figure 24 Two-point crossover. Figure showing an exchange of genes located between two selected crossover points in parent chromosomes, resulting in two offspring with new genetic makeup.

Uniform Crossover In this crossover, each gene is swapped between two parents with a certain crossover probability, p_c (Figure 25). This is highly disruptive and prevents propagation of co-adapted alleles but it has no positional bias. This is considered superior compared to other operators (Spears & Jong, 1991).

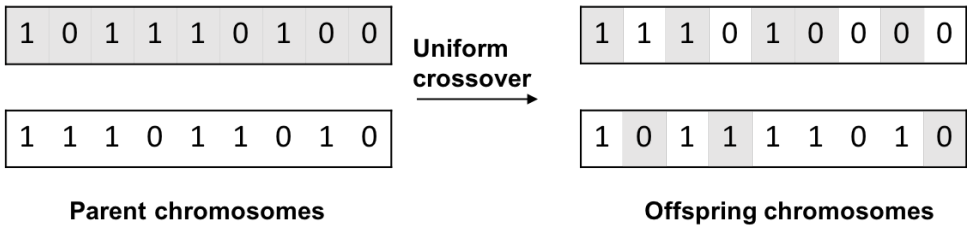


Figure 25 Uniform crossover with gene swapping between parents resulting in two offspring.

Step 4 Mutation

Mutation introduces relatively heavy changes, compared to a crossover, at a certain locus by changing the value of the corresponding gene randomly or systematically.

This allows a leap from one point of the landscape to another point and is thus helpful in avoiding ending up at a local minimum. At a genetic level, mutation acts as an insurance policy against the fixation of gene value to a certain preferred allele at any locus, and thus promotes genetic variation. Different mutation operators are as follows:

Flip Bit Mutation In this mutation, randomly selected genes are flipped i.e., if the gene value is 1, it is changed to 0 and vice versa (Figure 26). This is used in GAs that work with binary encoded chromosomes.

Figure 26 Flip bit mutation in which highlighted gene value is flipped to 0 from 1.

Boundary Mutation A selected gene's value is mutated to either upper or lower bound randomly (Figure 27). This is used in GAs that work with integer or float values.

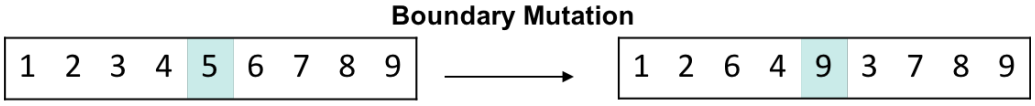


Figure 27 Boundary mutation in which highlighted gene value is changed to upper bound of values in the same chromosome.

Swap Mutation In this mutation, two randomly selected gene values are interchanged (Figure 28). This is commonly used in GAs that work with permutation-based encodings.

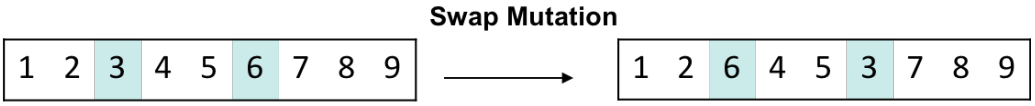


Figure 28 Swap mutation in which highlighted gene values are interchanged in a chromosome.

Scramble Mutation A selected set of genes is shuffled randomly in this mutation (Figure 29). This mutation is also suited for permutation-based encodings.

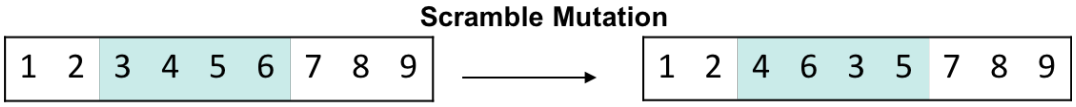


Figure 29 Scramble mutation in which highlighted genes values are shuffled randomly.

Inversion Mutation In this mutation, a selected subset of gene values is inverted (Figure 30).

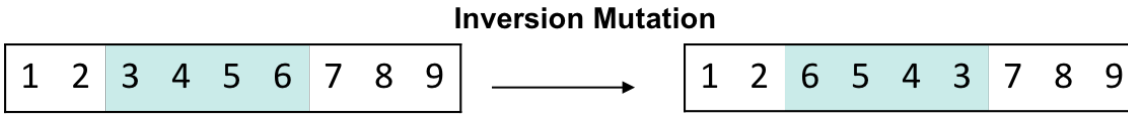


Figure 30 Inversion mutation in which highlighted gene values are inverted.

Evolutionary strategies work only with mutations and GAs work with both crossover and mutation. This sparked the argument that mutation is superior to crossover and resulted in "Crossover-Mutation Debate". However, Senaratna (2005)'s work on this debate emphasises that there is no absolute winner between these two. The author presented the usefulness of both crossover and mutation based on the mathematical frameworks and models established to study the constructive and destructive power of these operators.

Other than selection, crossover and mutation, the fourth element of the GA introduced (Holland, 1975; Goldberg, 1989*c*) is inversion, as mentioned above. Inversion works by selecting two loci and reordering the genes between these loci. This operator showed some success in GAs applied to the ordering problems (Parsons *et al.*, 1995). But this is not used in the recent implementations.

Steady-state and Generational GA

A Focus on Search Techniques

As discussed in the very beginning of this chapter, GAs belong to the class of evolutionary algorithms. GAs can be further sub-categorised as parallel and sequential (Figure 31). Parallel GAs are not used in this work, hence not described here. Sequential algorithms are further classified as "generational" and "steady-state". The Generational and Steady-state essentially differs in terms of "generation gap" which is

the fraction of new individuals at each generation (Jong & Sarma, 1993; Syswerda, 1991). Most implementations of GAs use a generational approach, where offspring produced after crossover and mutation completely replace their parents and form a new population at every generation. In steady-state, only a small fraction of parents with worse fitness are replaced with offspring produced after crossover and mutation. Steady-state GAs are used in systems where continual learning and remembering what was learnt is important. This thesis work focuses on generational GAs.

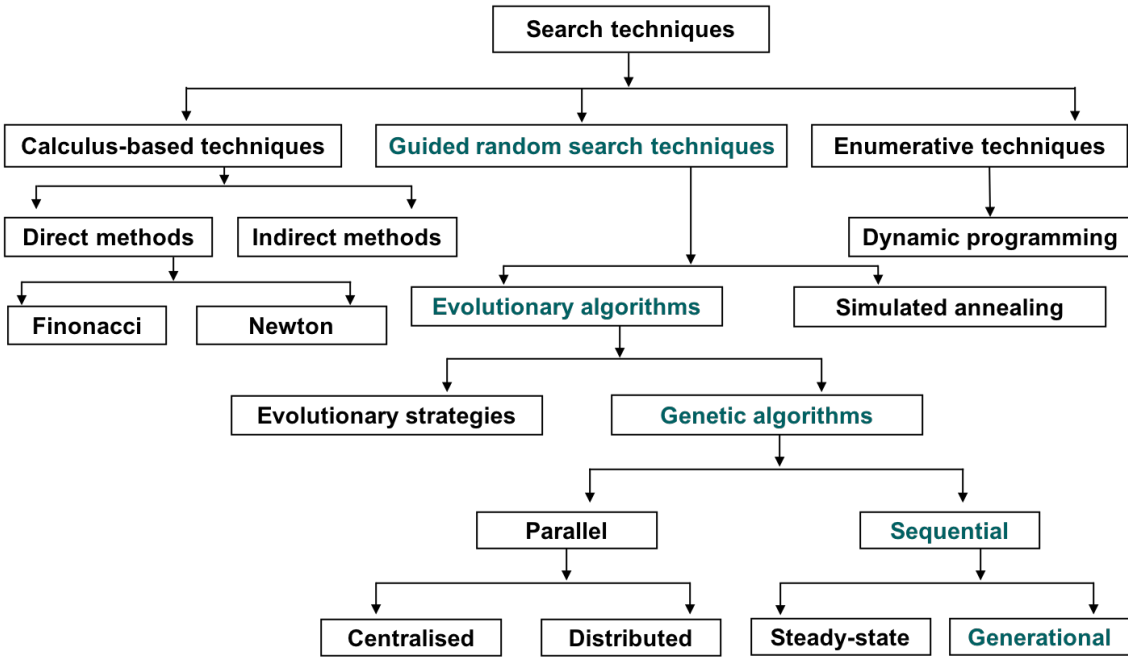


Figure 31 Classification of search techniques. Generational GA used in this work and its class is highlighted in teal.

What are the Best Parameter Settings in GA?

On-line and off-line Performance

The parameters such as crossover, mutation, and selection types and rates, are usually dependent. This makes the optimisation of a single parameter sequentially ineffective. A great deal of research was conducted to find the optimal parameters and was reported effective in the literature on the selected data set. Few such studies are discussed here. In these studies, the parameter's combinations are evaluated in terms of on-line and off-line performance of the GA. The "on-line" performance at generation t is the average fitness of all members that have been evaluated over t generations.

The “off-line” performance at generation *t* is the average value, over *t* generations, of the best fitness observed at each generation (Blickle & Thiele, 1996). Some of the “best” parameter settings that either improve on-line or off-line performance of GA, reported by different research groups after conducting various experiments are discussed here.

De Jong (1975) study on a set of test functions to improve on-line and off-line performance of GAs presented the best parameters as: a population size of 50-100 individuals, single point crossover with a crossover rate of ~0.6 per pair of parents and a mutation rate of 0.001 per gene. These settings were widely used in the GA community (Blickle & Thiele, 1996). A group of researchers (Bramlette, 1991; Grefenstette, 1986) used GA (“meta-level GA”) to optimise the parameters of another GA with De Jong’s test set. Each individual in this meta-level GA encodes the parameters: population size, generation gap, crossover rate, mutation rate, a scaling window, and an elitist or non-elitist selection strategy. The function of on-line or off-line performance of an individual based these encoded parameters was taken as the fitness of that individual. From on-line performance, the fittest individual’s encoded parameters are: population size 30, crossover rate 0.95, and mutation rate 0.01. However, this meta-level GA did not find a parameter set that gives better off-line performance than De Jong’s parameters. Schaffer *et al.*, (1989) did similar experiments on a small set of numerical optimisation problems. The best parameters according to this study are: population size 20 - 30, crossover rate 0.75 - 0.95 and mutation rate 0.005 - 0.01. It is clear from these studies that a small population size is better than a large population size in terms of on-line performance, contradicting studies that voted for large population size (Goldberg, 1989*d*; Alvarez, 2002). In view of variety of problems types in different applications, it is unlikely these parameters produce similar performance and hence cannot be taken as global recommendations. Based on a popular school of thought, the promising results can be expected when parameters adapt in real-time during search process (“self-adaption”). Davis (1989) study on self-adaption of operator rates provides useful insights on this approach.

Can We Use GAs for Phase Optimisation in Crystallography?

Applications of GAs in Crystallography

GAs have been used for solving many biological optimisation problems - for example finding a correct conformation of a small molecule in drug discovery - in the past few years. They have been used to solve many search problems related to crystallography as well. Most successful applications were found in powder diffraction studies (Harris *et al.*, 1998; Hanson *et al.*, 2005; Yakimov *et al.*, 2008, 2009). As a global optimisation method, GA was sometimes used in combination with local search methods to reach a global optimum. Some prominent examples are discussed below.

In the work of Nishibori *et al.*, (2008), a combination of GAs and Maximum Entropy method were used for *ab initio* structure determination of prednisolone succinate from powder diffraction data. GA was used together with Monte Carlo methods for performing structural analysis of crystalline materials in Immirzi *et al.*, (2008)'s work. This approach proved to be successful in yielding correct solutions when applied to four know molecular structures. Other notable examples include phase retrieval of coherent diffractive images using a GA, iterative phase retrieval algorithms by Truong *et al.*, (2017), automatic on-line beamline optimisation using GA and differential evolution by Xi *et al.*, (2017), refining structure of multidomain proteins and complexes against SAXS with NMR-derived restraints (encoded in a program called *DADIMODO*) using GA and simulated annealing by Evrard *et al.*, (2011).

A GA was also used in macromolecular crystallography to solve problems ranging from merging synchrotron crystallographic data to the identification of conformationally invariant regions in macromolecules. A few examples are presented in Table 3.

Table 3 Examples of application of GA in macromolecular crystallography.

Application	**Elitism**	**Crossover operator**	**Selection operator**	**Mutation**
Automatic beamline optimisation, (Xi *et al.*, 2015)	Yes	Single-point (rate = 0.8)	Roulette wheel	Insertion (rate = 0.05)
Grouping SAD datasets, (Foos & Nanao, 2019)	No	Single- & Two point (rate = 0.6)	Random	Insertion (rate = 0.5)

Low-resolution *ab initio* phasing – gamification, (Jorda *et al.*, 2016)	No	Single-point (rate = 0.3)	Tournament (Size 12) by human players	Insertion (rate = 0.2)
Merging of synchrotron serial crystallographic data, (Zander *et al.*, 2016)	No	Uniform (rate = 0.05)	Tournament (Size 3)	Uniform (rate = 0.05)
Identification of conformationally invariant regions in protein molecules, (Schneider, 2002)	Yes	n/a	Truncation (70%)	

Challenges and Demands

The use of GAs for phase optimisation has not been very well established. Some studies reported success when phase space is limited by the factors such as availability of prior information as in MR (Kissinger *et al.*, 1999; Chang & Lewis, 1997) or by the considerations of symmetry (Miller *et al.*, 1996) when applied to small molecule structures (Kariuki *et al.*, 1997; Nishibori *et al.*, 2008) or searching for a small subset of phase-determining heavy atoms (Chang & Lewis, 1994). Zhou & Su, (2004) reported that GAs are more efficient than simulated annealing for phase optimisation by minimising the least-square residual of Sayre's equation (Sayre, 1952) in centrosymmetric structures. Jorda & Michael, (2014) and Jorda *et al.*, (2016) developed an online game called *CrowdPhase* for *ab initio* phase retrieval in macromolecular crystallography based on a human-powered GA where players select better looking electron density maps (phenotypic expression of phases) manually. It was shown that the players were able to choose phase sets with a phase error of less than 30°.

Uervirojnangkoorn *et al.*, (2013) used a GA to optimise the phases for the 4% of strongest reflections using skewness of the density distribution as a fitness function at a resolution range from 2.6 Å to 3.5 Å. The electron density map with the optimized phases showed improvement in map quality, with increased map correlation from 0.56 to 0.70 in one of the test cases, after the density modification.

In view of these studies, efficient implementation of GAs for phasing in macromolecular crystallography starting from more or less random phases is seen as a challenging task, and is attempted in this work. Based on these earlier works, it is evident that a focus on understanding the behaviour of the algorithm and its parameters is what is lacking and is crucial for its optimal implementation. Moreover, Uervirojnangkoorn *et al.*, (2013) emphasised the importance of fitness function for phase optimisation using GA in macromolecular crystallography.

Scope of This Thesis

Inspired from the success of GAs in simple phase optimisation problems, I intend to study the nature of GAs and their parameters. In contrast to any other applications of GAs in macromolecular crystallography, this work was an attempt to identify a practical set of GA parameters and steps for phase optimisation, rather than an actual application. The major focus of this work is to provide a guiding light in designing the best GA scheme for phase optimisation.

Various designs of GA using different combination of operators and their parameters were tested for their efficiency in phase optimisation and the results are presented here. This work also has a minor emphasis on identifying the best fitness function for a GA which could drive this work towards an actual application.

Following Uervirojnangkoorn *et al.*, (2013) work on map moments, skewness and kurtosis were used as a cost function initially. To improve the performance of these atomicity related functions, a 3-dimensional parameter, map connectivity was introduced in the later experiments. Is it beneficial to use these parameters individually or do they perform better in combination? If they need to be combined, what is the best combination? To investigate these questions, various studies were performed to identify the relative performance of these parameters individually and in combination.

Chapter 3

Methodology and Materials

The problem of optimising and obtaining crystallographic phases is highly multidimensional, with parameters nearly equivalent to the number of reflections times the number of phase possibilities. For example, we can consider a molecule having approximately 150 residues in an asymmetric unit with nearly 10000 reflections. Assuming each reflection has four phase possibilities, sampling these possibilities for 10,000 reflections would require a computational time of 2^{20000} seconds, which is equivalent to around 10^{6000} seconds, while the estimated age of the universe is 4.32×10^{17}s. If this takes more time than the age of universe, sampling 10,000 reflections and considering a phase value of anywhere between 0° to 360° would be a combinatorial explosion. To minimise this computational complexity, the phases of every reflection is rounded to one of four possibilities: 45°/135°/225°/315°. These represent each quadrant in a circle signifying the possibility of a phase value in the range of 0° to 360° (Figure 32). This is called “phase discretisation” in this work.

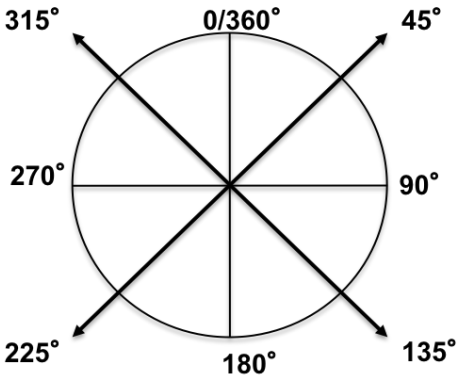


Figure 32 Phase discretisation. Phases are discretised to 45°/135°/225°/315° representing each quadrant of the phase space.

GAs are then used to sample this discretised phase space. The idea is to take advantage of the GA’s jumpy behaviour to sample this restricted phase space within a realistic computational time. The first step in designing an efficient algorithm is to select an appropriate set of test cases with a desired degree of phase error that can provide room for improvement. Two test cases selected for this purpose and their selection criteria are discussed below.

Test cases

Throughout the study, two macromolecular test cases taken from the PDB were used. Most of the studies presented in this report were carried out with the X-ray data of these two proteins truncated to a resolution of 2.5 Å. At 2.5 Å resolution and worse, angle-bonded atoms are no longer resolved and the corresponding density map no longer contains any traces of “atomicity” (Figure 33).

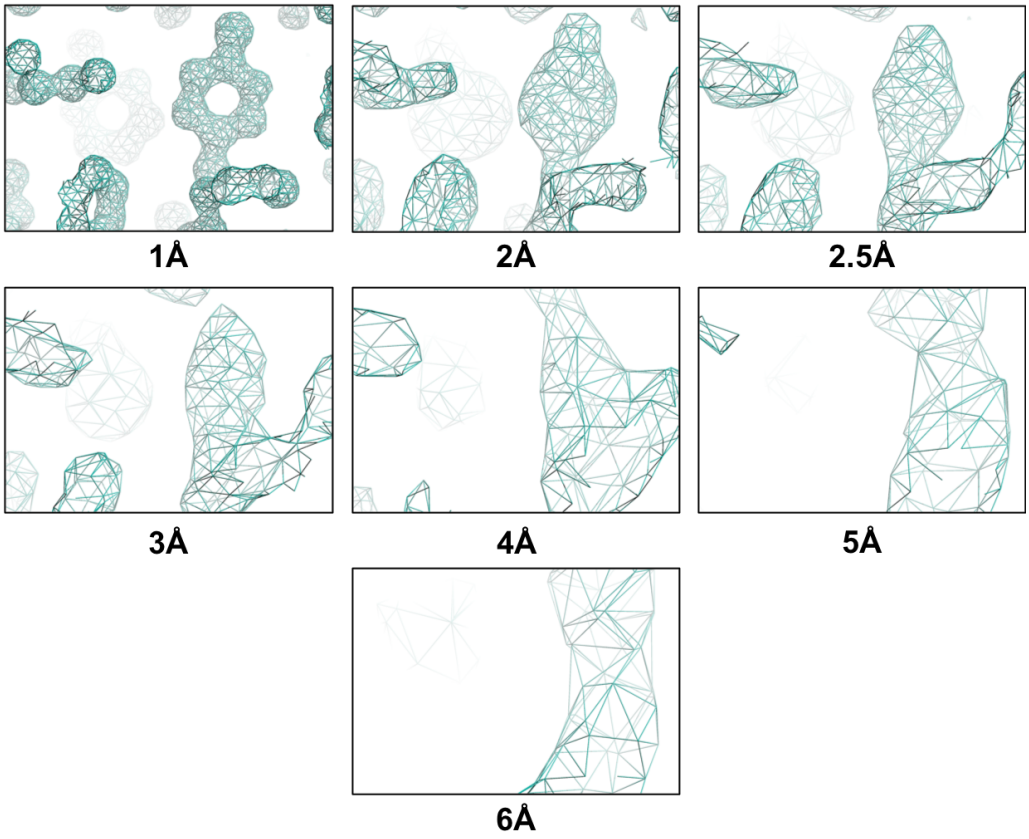


Figure 33 Electron density map at different resolutions. In the map at 1 Å the tyrosine residue of ribonuclease from *Streptomyces aureofaciens* (PDB ID: 1LNI) was well resolved showing atomic details. After 2.5 to 3 Å resolution, the map is more blobby and misleading. These maps are generated using Coot (Emsley & Cowtan, 2004).

Case I: Saicar Synthase from *Saccharomyces cerevisia*

The structure of saicar synthase (Levdikov *et al.*, 1998) was solved using experimental phases and refined using anisotropic atomic displacement parameters at 1.9 Å with an

Rfactor of 0.16 (Figure 34). The crystals belong to space group $P2_12_12_1$ and there is one molecule per asymmetric unit. The X-ray data, with experimental isomorphous replacement phases after density modification using solvent flattening (11859 reflections), was extended to 2.5 Å resolution. The experimental phases to 2.5 Å resolution were taken without their figures of merit (a measure of phase error). This reduced the map correlation coefficient (more on this parameter is discussed in the section "Fitness function" of this chapter) from 0.7867 to 0.7557. To further limit the variation of the phases for acentric reflections to only four possible values, they were rounded to the nearest value of 45°/135°/225°/315°.

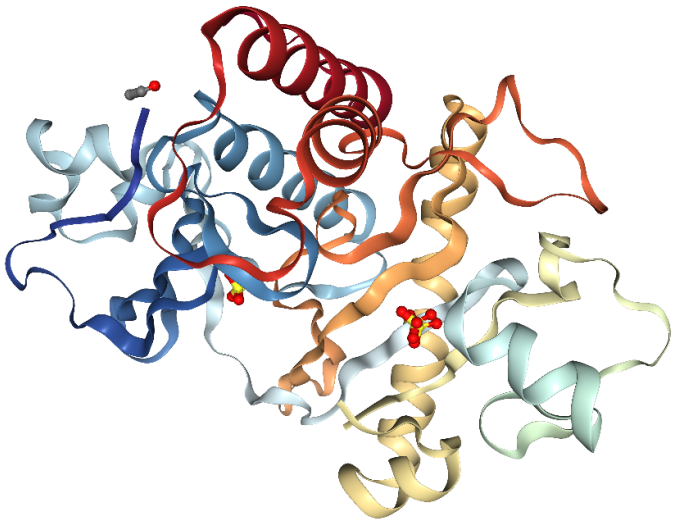


Figure 34 Structure of saicar synthase solved at resolution 1.9Å (PDB ID: 1A48). Figure taken from the PDB (Burley *et al.*, 2019).

The rounded phases were taken as the initial phase set for the development of the method. As we were starting with discretised phases, it would be difficult to reach the "actual solution", a final map with a phase possibility between 0° to 360°. To create a reference for this discretised phase scenario, the phases of a density map at 1.9 Å were rounded to 45°/135°/225°/315°, resulting in a map correlation coefficient (MCC) of 0.9088 to the final map (map at 1.9 Å). This map is considered the "final point" to reach using a GA when starting with discretised phases. The MCC to the final map, the overall phase error and the MCC of the *final point* of the GA are shown in Table 4.

Case II: Ribonuclease from *Streptomyces aureofaciens* (RNase SA)

The structure of RNase SA (Sevick *et al.*, 2002) was refined using atomic displacement parameters at 1.0 Å to an Rfactor of 0.161 (Figure 35). The crystals belong to space group $P2_12_12_1$ and contain two molecules per asymmetric unit. The X-ray dataset was truncated to 2.5 Å resolution to represent the resolution range similar to that of test case I. The Wilson B factor was upweighted by 36 $Å^2$ accordingly. The model was refined using Refmac (Murshudov *et al.*, 1997) against this 2.5 Å data (in order to reduce the high-resolution model bias) that was giving an Rfactor of 0.092. There are 6,866 unique reflections in this data to the selected resolution limit. The phases from the model refined against the 2.5 Å data were subject to an additional uniformly distributed phase error of 50°, as described in the next paragraph.

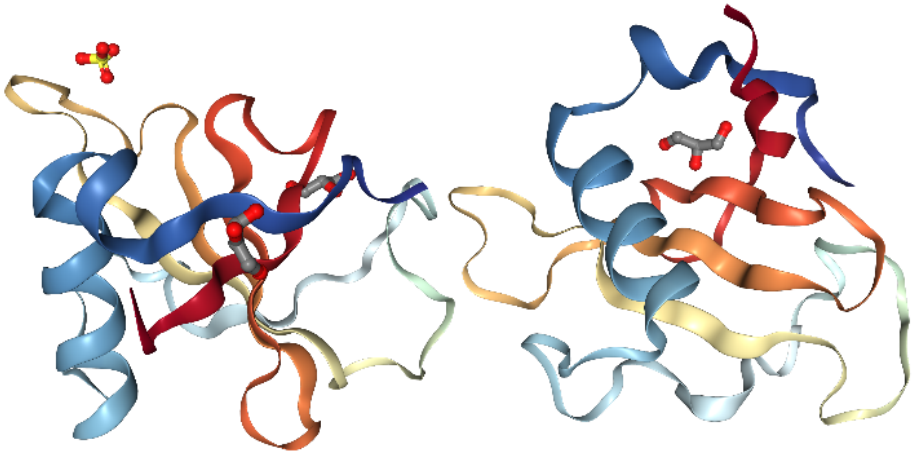


Figure 35 Structure of RNaseSA solved at resolution 1.0Å (PDB ID: 1LNI). Figure taken from the PDB (Burley *et al.*, 2019).

The phases for centric reflections were changed to +180° in 50/180 cases at random, resulting in a mean cosine of the phase error of 0.44. The phases for acentric reflections were changed with an addition of a phase error uniformly distributed within the range +/-100°, resulting in a mean cosine of the phase error of 0.56. These phases of acentric reflections were then rounded to the nearest value of 45°/135°/225°/315 degrees introducing an additional but small phase error of about 3°.

The rounded phases were taken as the initial phase set for the development of the method. The *final point* to be reached here is a map at 2.5 Å with phases rounded to

45°/135°/225°/315° while having an MCC of 0.8898 to the final map (map at 2.5 Å without artificially introduced phase error). The MCC of the final map, the overall phase error and the MCC of the *final point* for the GA were as shown in Table 4.

Table 4 The characteristics of the initial density maps and their corresponding phases.

Test case	Resolution (Å)	Acentric/ centric reflections	MCC	Phase error (degrees)	Mean cosine of the phase error	MCC of the *final point*
Saicar synthase	2.46	10261/1598	0.6873	51.27	0.518	0.9088
RNase SA	2.50	5700/1166	0.4944	52.76	0.505	0.8898

The general formulation of the phase optimisation problem is shown in Figure 36. The task is to start the map with discretised noisy phases (Figure 36b) and reach a map with discretised final phases or correct phases (Figure 36c) using GA.

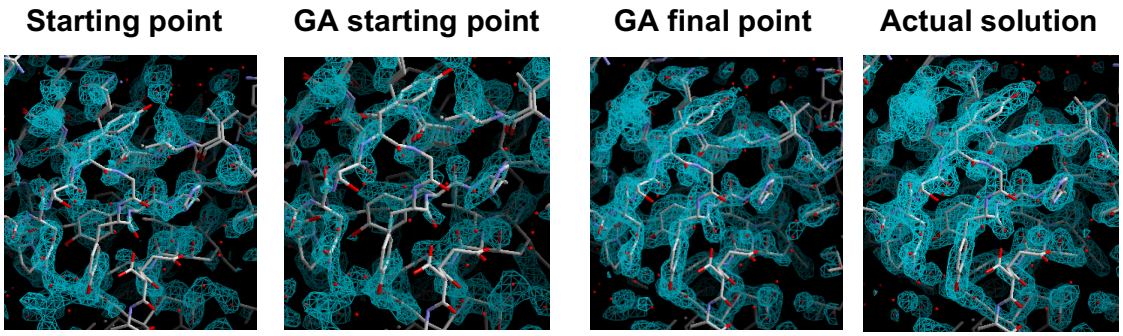


Figure 36 Formulation of the phase optimisation task. The noisy electron density map (Figure 36a) is the starting point whereas the noisy rounded map (Figure 36b) is the starting point for GA. The improved map after phase optimisation using GA is expected to have density features similar to the GA *final point* (Figure 36c) which is closely resembles the actual solution (Figure 36d). Electron density maps are generated using RNase SA data in ArpNavigator (Langer *et al.*, 2013).

The assumption is that as the discretised final map has well defined density with features very much similar to the final map with correct phases, this is good enough to interpret the density accurately.

Implementation of GA

Initialisation of the Population

The noisy discretised phase set was mapped on to a “chromosome”, which was represented as a string of integers. Each “gene” in the chromosome has four allele possibilities: 45°/135°/225°/315˚ for acentric reflections and one of the two values for centric reflections (0°/180˚ or 90°/270˚). This initial chromosome is hereafter called the *first parent* (Figure 37).

45	225	135	45	0	315	135	270	45	315

Figure 37 Haploid chromosome representation of noisy discretised phase set, the *first parent*.

As the phases of crystallographic reflections are independent from each other, the reflections for the first parent have been sorted in an arbitrary order. The order of the reflections was kept fixed for all members of the population throughout the GA.

Generation of First Parents

The phase variability

Approximately 2000 *second parents* were generated from the first parent. The difference between the first parent and the second parents is hereafter called *phase variability*. The five different values of phase variability investigated in this work were: 0.5°, 1°, 2°, 4° and 8°. The variability was achieved by introducing an average phase error equivalent to these values of the phase variability between the first parent and each of the second parents. For example, to obtain second parents with 1° phase variability, a phase for each centric reflection was changed from its value in the first parent by an addition of +180° with a probability of 1/180. Similarly, a phase for each acentric reflection was changed with an addition of either +90°, +180° or +270° with a

probability of 1/360. The average acentric reflections having changed phases compared to the phases of the first parent (simply called “distance to the first parent” hereafter, and this definition covers phase changes of both centric and acentric reflections) can be computed by using the following equation (10). For example, this average for different phase variabilities when the number of reflections equal to 6000 is given in Table B.1.

$$Average\ phase\ changes = \frac{\sum_{i=1}^{n} {}^{N_{refl}}/_{360}\ (3\epsilon)}{n} \tag{10}$$

where, n is population size, N_{refl} is the number of reflections and ϵ is the phase variability.

Population and Generation

Each generation starts by recombining parents using crossover. The population size expanded from ~2000 to 20,000 after crossover. This population was then subjected to selection based on the fitness function. The surviving population were passed as parents to the next generation.

Crossover

For recombining the phase sets, the one-point crossover and the uniform crossover operators were used.

Parameter Selection Rationale

As the structure factor amplitudes and phases of the X-ray reflections are almost independent from each other, both one-point crossover (with high positional bias) and uniform crossover (with high distributional bias) can be used.

One-point Crossover For a selected pair of parents, a crossover point (the reflection number) was taken at random. The phases from each of the two parents before the crossover point were passed over to the two children. The phases after the crossover

point were swapped so that the phases from one parent were passed over to the second child while the phases from another parent were passed to the first child.

Two variants of the one-point crossover were used in this work. In the first variant, for each randomly selected pair of parents, four crossover points are generated using a random number generator. Each crossover resulted in two children, giving 8 children in total by crossing over at 4 points (Figure 38a). In the second variant, for each randomly selected parent, four partners were selected at random. Each pair of parents generated two children, giving 8 children in total after recombining with four different partners (Figure 38b).

Uniform Crossover For a selected pair of parents, the phases which were the same in both parents were passed over to the two children. When the phases in the parents differed, they were swapped with a probability of 0.5 and then passed over to the children.

Two variants of the uniform crossover were developed: In the first approach, the randomly selected parent was crossed over with four partners selected at random. Each pair of parents produces two children, totalling 8 children. In the second approach, eight partners were chosen for every randomly selected parent. Each pair produces two children. The first child was retained and the second child was discarded. After crossover, the generated children together with their parents (passed as elite members) expands the population size to 19440.

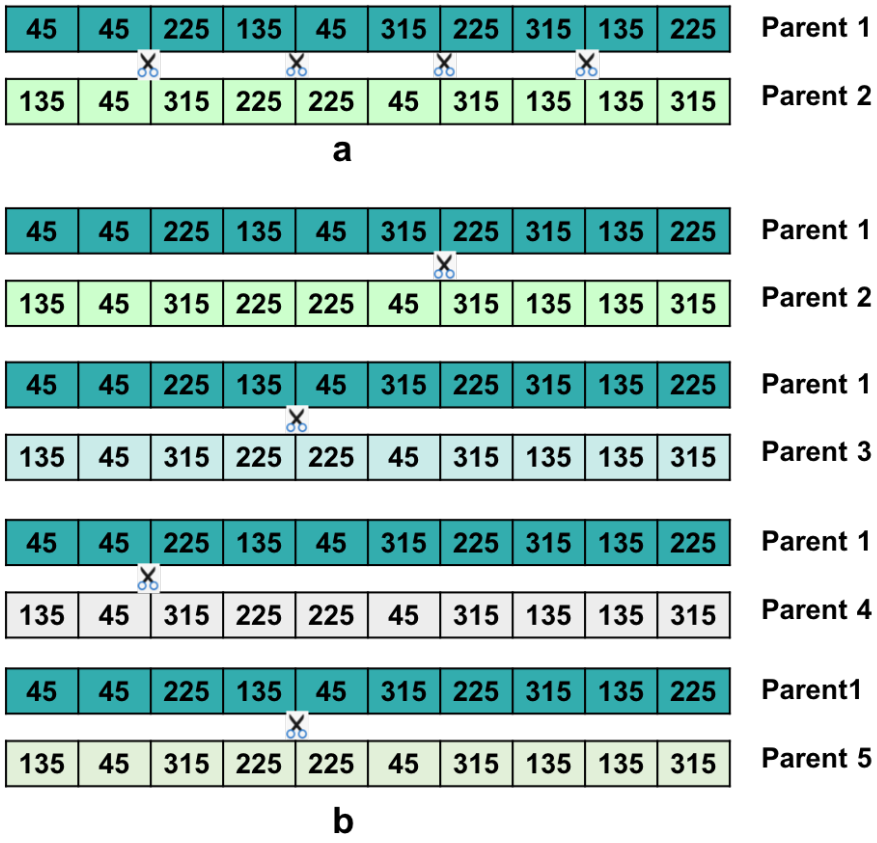


Figure 38 Two variants of the one-point crossover. Figure 38a showing parent1 being crossed over at four different points with parent 2. Figure 38b showing parent1 being crossed over with four different parents (Parent 2, 3, 4 and 5).

Selection

The SUS and the tournament selection operators were used in this work.

Parameter selection rationale

In the studies performed by Blicke and Thiele, tournament, linear and exponential ranking selections were given as the best selection methods. As we planned to investigate different sizes of tournament including binary, we skipped linear ranking which was shown to have identical performance (Blickle & Thiele, 1996). When deciding between tournament and exponential, tournament was selected as it allows

parallelisation which can be helpful in working with computationally complex problems like phase optimisation.

SUS, which has less selection bias, was also tested for its performance in the phase optimisation.

Stochastic Universal Sampling

In the first step, the phase sets in the population are sorted according to their fitness value. These sorted phase sets were mapped on to a chromosome. Here, each gene encodes a phase set and its fitness value. Starting at a random point on this chromosome, P pointers were generated at regular intervals (Figure 39). The distance between the pointers or the width of the interval was F/N where N is the number of offspring to be generated and F is the total fitness of the population.

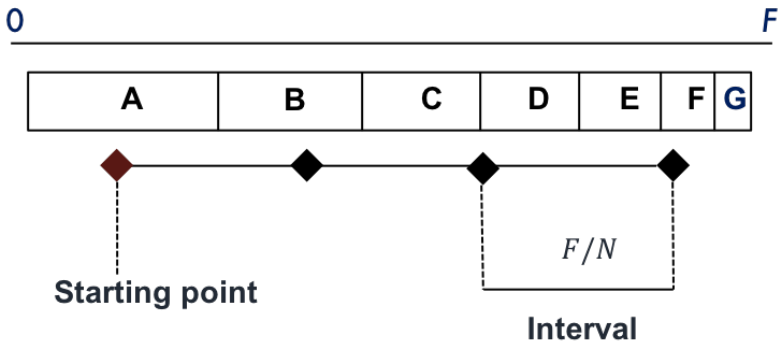


Figure 39 Implementation of SUS. Genes A to G correspond to the phase sets and their fitness values after sorting in decreasing order based on their fitness value. Genes are selected at the regular intervals (Black diamond shaped markers) from the random starting point (Brown diamond shaped marker).

Tournament Selection

Tournament selection was also used in this work. The sizes of the tournament operator (t) studied ranged from 2 to 9. The tournament sizes of 2 and 9 were extensively studied in GA protocols called “designs” in this work. The effect of tournament size on the diversity of the population can be described in simple statistical terms. If the population members are sorted by their fitness value and ranked at an interval between

0 and 1, with 0 being the member with the best and 1 being the member with the worst fitness value, the probability of selecting a point x within this interval using different tournament sizes t can be expressed as follows:

$$p(x) = t(1-x)^{t-1} \tag{11}$$

The probability of selecting a point x for tournament size 9 is a parabola of the eighth order starting from 9 at $x = 0$ and dropping to 0 at $x = 1$. The probability distribution for tournament size 2 is triangular, with a straight line starting from 2 at $x = 0$ and falling to 0 at $x = 1$ (Figure 40).

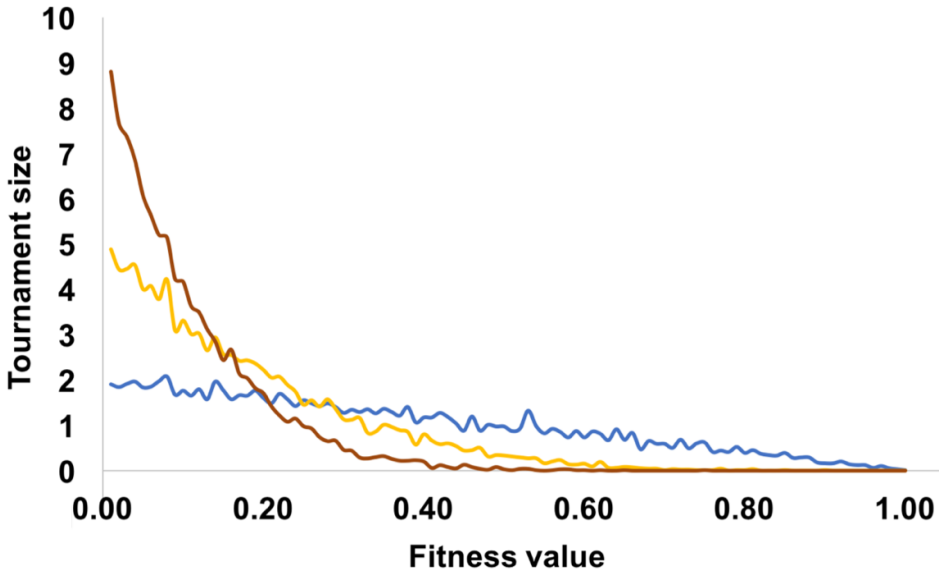


Figure 40 The selection intensity for the tournament of size 9 (shown in brown), 5 (shown in yellow) and 2 (shown in blue). In this X-axis, 0 represents best fitness while 1 represents worst fitness.

In this selection, for example, $N/2$ of the population members (where N is the size of the population) were selected using a size 2 tournament operator. The remaining $N/2$ of the population members were ignored. The survivors selected in the current work

have two different functional roles: to participate in the reproduction for producing next generation children and to move on to the next generation as *elite* members (Table 5).

Table 5 The survivors, children, and *elite* fractions for tournament size 2 and 9.

Tournament size	Survivors	*Elite*	Children
9	$N/9$	$1/9$	$8/9$
2	$N/2$	$1/4$	$3/4$

Mutation

In this work, a concept similar to the flip bit is applied for the integer-based genes. Two variants of this concept were designed: non-targeted and targeted. In the non-targeted, the genes for mutation were randomly selected while in the targeted, the genes for mutation were selected based on the degeneracy statistics (explained in the "Directed mutations" section of this chapter). The phase values of these selected genes were then changed uniformly to either to 45˚/135˚/225˚/315˚ for acentric reflections and to 90˚/270˚ or 0˚/180˚ for centric reflections. The number of genes selected was defined by the mutation rate.

Design Rationale

A major concern in using the non-targeted random mutations is the possibility of a large jump in the search space. The jump depth and length were determined by mutation rate and number of generations exposed to mutations respectively. To understand this effect, relatively high mutation rates of 1, 2, 4, 8, 100, 200, 300 were introduced into every generation. The drastic drop in the MCC showed that introducing mutations into every generation with high mutation rates can be detrimental (Figure C.2). This might drive the system to jump to an incorrect region of the phase space. To avoid this, mutations were only introduced in some selected generations. These were the generations in later developmental stages where growth became stagnant (identified by non-linear growth in the fitness value). The mutation rate used was three times lower that the distance of the population to the first parent at these generations.

Two types of non-targeted mutations were designed: static and dynamic. The common step for these mutations was to start with identifying the generation at which a nonlinear growth in the MCC was first observed. For simplicity this is called “non-linear growth generation”.

Static Mutations From the non-linear growth generation, mutations with a constant mutation rate were applied for ~20-30 generations and then turned off completely for the subsequent generations.

Dynamic Mutations At the non-linear growth generation, mutations with a decreasing mutation rate (decrement of 0.01 per generation) were applied until it reached zero. No mutations were introduced in the subsequent generations.

Directed Mutations Directed mutations are performed based on what is called in this work the statistics of “reflection degeneracy”:

- If a reflection has the occurrence of all the four possible phase values (45˚/135˚/225˚/315˚) at least once in a population at a given generation, this reflection is considered a “non-degenerate” reflection at that generation.
- If a reflection has the occurrence of three of the four possible phase values at least once per population in a given generation, this reflection is considered a “slightly degenerate” reflection at that generation.
- If a reflection has the occurrence of two of the four possible phase values at least once per population in a given generation, this reflection is considered a “moderately degenerate “reflection at that generation.
- If a reflection’s phase value is the same throughout the population in a given generation, this reflection is considered a “completely degenerate” reflection in that generation.

The four scenarios described above can be seen as four schemas that were monitored to understand the behaviour of GAs in phase optimisation. Direct mutations are introduced in expectation that it is more beneficial to mutate a completely degenerate reflection than to mutate a non-degenerate reflection to prevent the loss of diversity.

Fitness function

The fitness functions used in this work were: MCC, moment of the density histogram (skewness and kurtosis), and map connectivity.

Map Correlation Coefficient

The MCC is a linear correlation coefficient between the map in question and the map with phases from the refined model. The MCC was computed in reciprocal space following (Lunin & Woolfson, 1993) as implemented in the ARP/wARP module ph_rms. Although the MCC cannot be used as a realistic fitness function, it was employed in this work to benchmark the various designs of the GA that were studied.

Moments of Density Distribution

The studies performed by Cochran (1955), Podjarny & Yonath (1977) and Lunin (1993) indicated that certain properties of the electron density map can be expressed as statistical moments of density distribution. Furthermore, Podjarny & Yonath proposed that skewness, which is related to the third moment of the histogram, can be used to identify the quality of the electron density map. Petersen performed an extensive study on eleven 3D moment invariants and one higher-order chiral invariant of local regions of electron density map, (Peterson 2013). This study further proved the usefulness of the skewness and kurtosis in analysing the quality of electron density distribution. Skewness was also used as a fitness function for the optimisation of experimental phases using a genetic algorithm by Uervirojnangkoorn *et al*., (2013).

Based on these studies, in this work we evaluated the use of skewness (measure of symmetry) and kurtosis (measure of peakedness) related to the third and the fourth moment of the histogram respectively. The moments were used both individually and in combination. The first two moments of the histogram, mean/median/mode (measure of location) and standard deviation (measure of spread), do not use phases in their computation, equations 12 and 13. Therefore these moments cannot be used to evaluate the phase quality. The skewness and kurtosis which have phase components in their calculations can be computed using the equations 14 and 15 respectively.

$$m_1 = \frac{F_{000}}{V_{cell}} \qquad (12)$$

$$m_2 = \frac{\sum_s F_s^2}{(V_{cell})^2} \quad (13)$$

$$m_3 = \frac{\sum_{s_1+s_2+s_3=0} F_{s_1} F_{s_2} F_{s_3} \exp\left[i\left(\varphi_{s_1} + \varphi_{s_2} + \varphi_{s_3}\right)\right]}{(V_{cell})^3} \quad (14)$$

$$m_4 = \frac{\sum_{s_1+s_2+s_3+s_4=0} F_{s_1} F_{s_2} F_{s_3} F_{s_4} \exp\left[i\left(\varphi_{s_1} + \varphi_{s_2} + \varphi_{s_3} + \varphi_{s_4}\right)\right]}{(V_{cell})^4} \quad (15)$$

(Lunin, 1993)

where m_1, m_2, $m_3$and m_4 are mean, standard deviation, skewness and kurtosis respectively, F_s is structure factor amplitude φ_s is the phase of a reflection s, V_{cell} is volume of the cell, F_{000} is the structure factor amplitude of the reflection 000.

The equations 12 to 15 can be used to compute the moments in reciprocal space. However, in this work the moments are computed in real space (from the electron density map) using the equation 16.

$$m_k = (1/V_{cell}) \int_V \rho(r)^k \, dV_r, \quad k = 0, 1, \ldots . \quad (16)$$

(Lunin, 1993)

where $m_k$is the moment of order k, V_{cell} is volume of the cell $\rho(r)$ is electron density at position r. These are implemented in the ARP/wARP module Histogram.

Map Connectivity

The higher order one-dimensional moments of the density histogram, which are dependent on the resolution of the data, may not be an effective metric. Hence, the use of 3-dimensional information such as map connectivity obtained by generating skeletons from the electron density map was also included as a component of the fitness function. The computation of connectivity is discussed in the section "Connectivity" of chapter 5.

Parameters for Monitoring the Performance

The overall performance of the entire GA design was monitored by three parameters:

- improvement in the MCC values of the population.
- the increment in the **distance** of the phase set expressed as the average number of reflections with changed phases compared to the first parent.

- the **divergence** of the phase set expressed as the average number reflections with changed phases compared to the other members of the population.
- number of residues built by ARP/wARP, an automated model building software.

Termination criteria

The algorithm was designed to terminate at a generation where the MCC of the population was approximately equal to the MCC of the *final point*.

Designs of GA

The first parent (with noisy spared phases) was taken as the starting point. From the first parent, using five different protocols with phase variabilities of 0.5°, 1°, 2°, 4° and 8°, five sets of first-generation parents were produced. Three different designs were developed using different parameters to execute the subsequent steps in a GA (Figure 41).

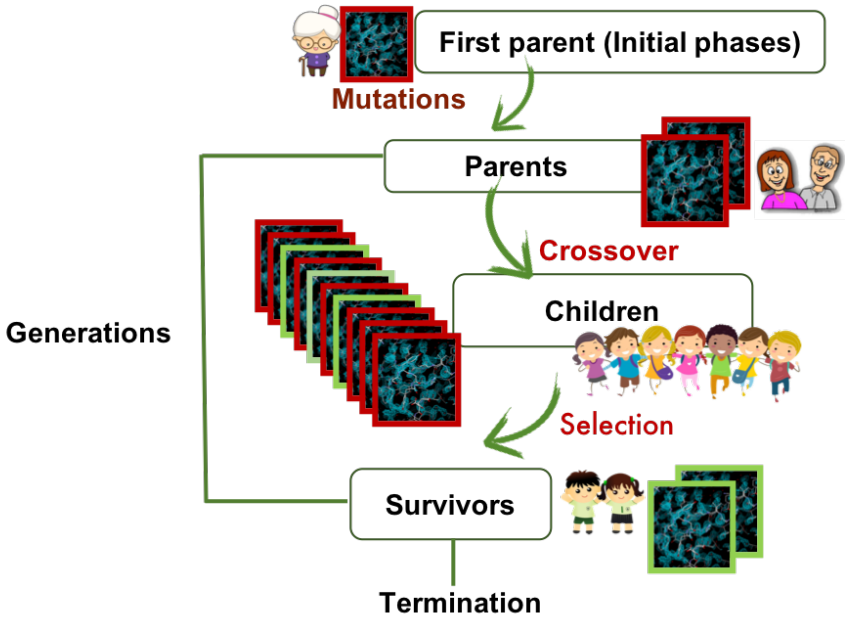


Figure 41 The overview of a GA design.

GA Design 1

In the first generation, each parent was subjected to crossover with the first parent using a one-point crossover (first variant described in the section "one-point crossover" of this chapter) to produce children. The second and subsequent generations use either the first or second variant (discussed in the section "one-point crossover" of this chapter) of the one-point crossover to produce children. The selection was performed using either SUS or tournament selection. The size of the tournament was set to 9. The survivors after selection participated in a crossover to produce children for the next generation and were included as *elite* members in the population together with their children. No mutations were applied. This process was continued until convergence (average MCC of the population ≅ MCC of the GA *final point*) was achieved (Table 6).

Table 6 Parameters used in GA design 1.

Parameter	**Design 1**	
	1a	**1b**
Crossover	One-point V1*	One-point V2**
Selection	Tournament with size 9 / SUS	Tournament with size 9
Mutation	None	None
Population composition	$1/9$ *elite* (All parents) $8/9$ children	$1/9$ *elite* (All parents) $8/9$ children

* First variant of one-point crossover **second variant of one-point crossover (discussed in the section "one-point crossover" of this chapter)

GA Design 2

The first generation was created similarly to the first GA design. The second and subsequent generations use uniform crossover variant 1 (refer to section "Crossover" in this chapter for further details) to produce children. The tournament operator was used to select survivors. The size of the tournament was set to 9 and 2 in design 2a and 2b respectively. The population from the second generation in the design 2a was composed of $1/9$ survivors that became elite members and $8/9$ children. The population from the second generation in the design 2b was composed of $1/4$ survivors that became elite members and $3/4$ children. No mutations were applied (Table 7).

Table 7 Parameters used in GA design 2.

Parameter	Design 2	
	2a	2b
Crossover	One-point V2** for the first generation and uniform for other generations	One-point V2** for the first generation and uniform for other generations
Selection	Tournament with size 9	Tournament with size 2
Mutation	None	
Population composition	$1/9$ *elite* (All parents) $8/9$ children	$1/4$ *elite* (All parents) $3/4$ children

**Second variant of one-point crossover (discussed in the section "one-point crossover" of this chapter)

GA Design 3

In this design, the first generation was obtained similarly to that of GA design 1 and 2 with the exception that uniform crossover was used for producing children by mating the second parents with the first parent. The population was then subjected to selection using a tournament of size 2.

Table 8 Parameters used in GA design 3.

Parameter	Design 3			
	3a	3b	3c	3d
Crossover	Uniform crossover for all generations	Uniform crossover for all generations	Uniform crossover for all generations	Uniform crossover for all generations
Selection	Tournament with size 2	Tournament with size 2	Tournament with size 2	Tournament with size 2
Mutation	None	Static	Dynamic	Directed
Population composition	$1/10$ *elite* (best parents) $9/10$ children	$1/10$ *elite* (best parents) $9/10$ children	$1/10$ *elite* (best parents) $9/10$ children	$1/10$ *elite* (best parents) $9/10$ children

The first pair of survivors (parents of the next generation) selected were recombined using uniform crossover to produce two children. Among these two children, only the

first child passed to the next generation. This selection of a pair of parents and passing of a child to the next generation was continued until 90% of the desired population size ($0.9 \times N$) was generated. The remaining 10% of the next generation was then filled with the best performing parents, selected based on their fitness value (Table 8).

Computational Resources

The algorithm was developed in three different programming languages: Fortran 77 for mathematical computations, shell scripting for integrating Fortran programs and other executables, with python 2.7 scripts used to analyse various performance parameters of the GA. This developed package can run on Mac and Unix, or Unix-like operating systems.

The package uses some of the CCP4 (Winn *et al.*, 2011; Ten Eyck, 1973; Read & Schierbeek, 1988) and ARP/wARP libraries (Lamzin & Wilson, 1993) for tasks such as computation of electron density map (module “fft” of CCP4), skewness and kurtosis of the density distribution (module “histogram” of ARP/wARP), MCC and mean cosine of the phase error (module “ph_rms” of ARP/wARP).

The computations of GA were parallelised using EMBL’s cluster computing facilities, “sistina” and “hyde”. Each generation took 90 to 120 minutes of computational time of which ~30% was used for ASCII to binary file conversions (~20000 files per generation) and ~20% for computing electron density maps from reflection data.

Optimisation of GA for Phase Improvement

To identify a combination of GA parameters that are ideal for phase optimisation problem, an artificial fitness function, MCC, was used. This chapter presents the comparative performance of all three designs of GA using this fitness function. To enable comparison, the scale of all plots in this chapter are kept same.

Premature Convergence

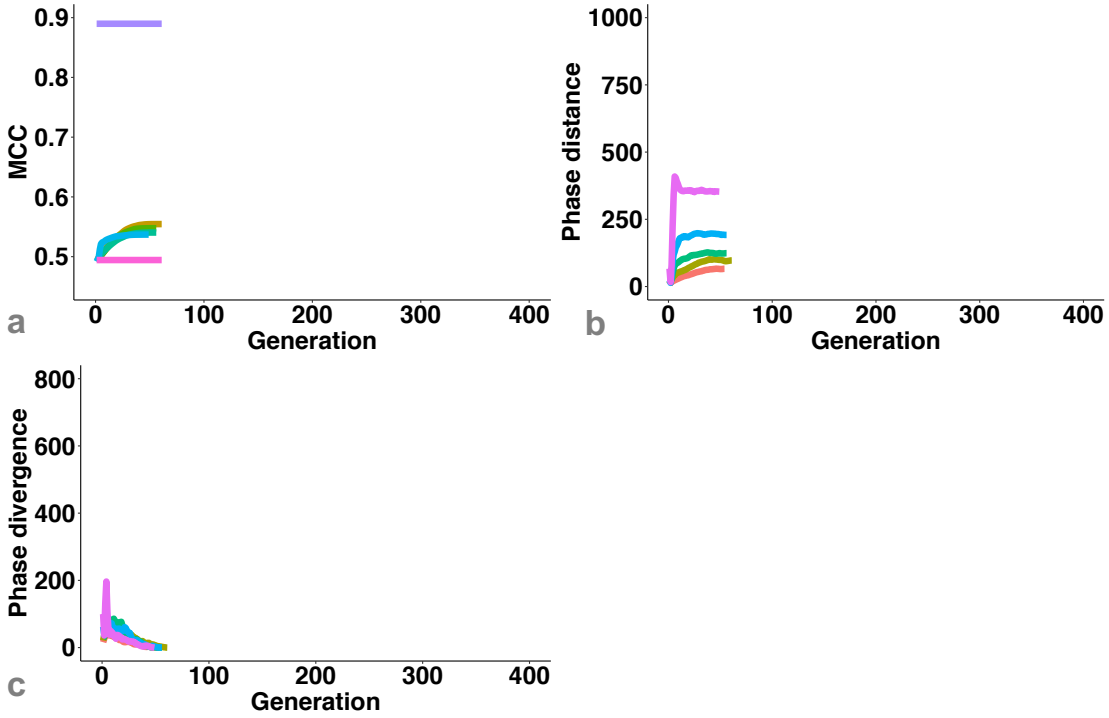
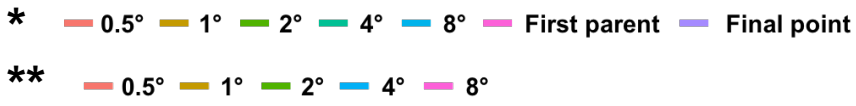


Figure 42 GA design 1 with MCC as a fitness function in the test case II. The growth of MCC (Figure 42a), phase distance (Figure 42b) and phase divergence (Figure 42c) for all phase variabilities are plotted as a function of generation. * colour legend for Figure 42a ** colour legend for Figures 42b, 42c.



The distribution of the population produced by GA design 1 showed *premature convergence* (refer to section "What Type of Problems is GA Best Suited to Solve?" of

chapter 2). The MCC reached a steady-state well below the maximum MCC in test case II (Figure 42a). The evolution of the system stagnated (Figure 42b) with no diversity among population, as can be seen from the "phase divergence" plot (Figure 42c).

A noticeable dependence on the phase variability can be seen from Figure 42b. The higher the phase variability, the longer distance the population travelled from the first parent. However, populations with initial phase variability greater than 1° have converged faster and showed comparatively smaller improvement in the MCC.

The *premature convergence* may be due the insufficient diversity among the population produced by the crossover or due to the high selection intensity imposed by the selection operator. To investigate this, different crossover and selection sizes were studied.

Crossover

To amplify the diversity among the population, the one-point crossover was replaced with uniform-crossover in GA design 2. Swapping nearly every gene produced a population with very high variance. This population converged slower (Figure 43c) and showed nearly three-fold (45.2%) improvement in the MCC compared to the one-point crossover (15.2%) in the test case II with 1° phase variability at generation 400 (Figure 43a).

To further improve the diversity needed to reach the *final point*, selection intensity was reduced by using tournament size 2, and uniform crossover variant 1 was replaced with variant 2. The effect of tournament size is discussed in the section "Selection" of this chapter. With uniform crossover variant 2, only two children per pair of parents were produced and the second child was discarded to prevent the accumulation of closely related members in terms of their genetic information. This was useful in avoiding the movement of the population to an incorrect local minimum (Figure 43b) in the early stages due to accumulation of similar genetic copies in the population (Figure 43c). Thus approximately 77.3% improvement in MCC was achieved by using GA design 3 in the test case II with 1° phase variability at generation 300 (Figure 43a).

The improvement in the MCC, the distance travelled in the phase space away from the *GA starting point* (*distance to the first parent*) and the growth of the diversity among the population using three different designs (GA design 1, GA design 2a, GA design 3a) with different crossover operators are presented in Figure 43. This comparative illustration clearly depicts the effect of loss of diversity on *premature convergence* and the critical role of crossover operator type in maintaining diversity.

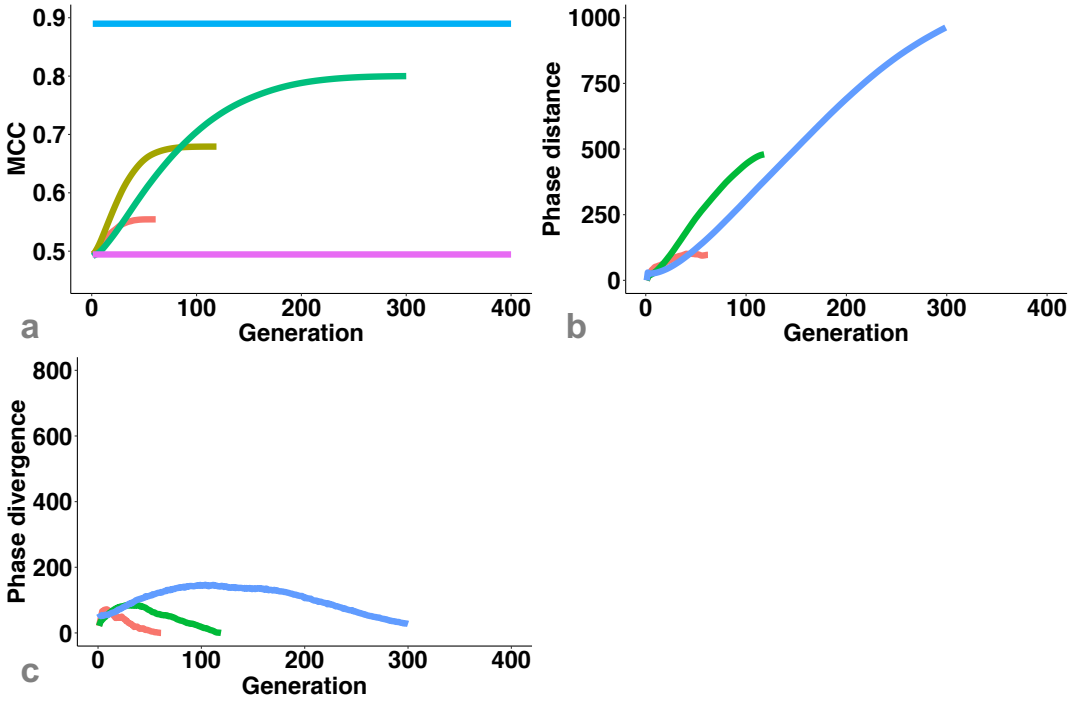


Figure 43 Comparison of GA design 1, GA design 2a and GA design 3a to show the effect of crossover. The growth of MCC (Figure 43a), phase distance (Figure 43b) and phase divergence (Figure 43c) for phase variability of 1° in the test case II are plotted as a function of generation. * colour legend for Figure 43a ** colour legend for Figures 43b, 43c.

Selection

The importance of the selection intensity and the role of tournament size in controlling the selection intensity and thereby controlling the diversity are shown in Figure 44. The

improvement of MCC using GA design 2a and 2b are shown in Figure 44a. With tournament size 9 having a parabolic probability distribution (refer to section "Tournament Selection" of chapter 3) that selected far worse individuals (Figure 40), the diversity in the population (Figure 44c) decreased earlier resulting in *premature convergence* (Figure 44a and 44b).

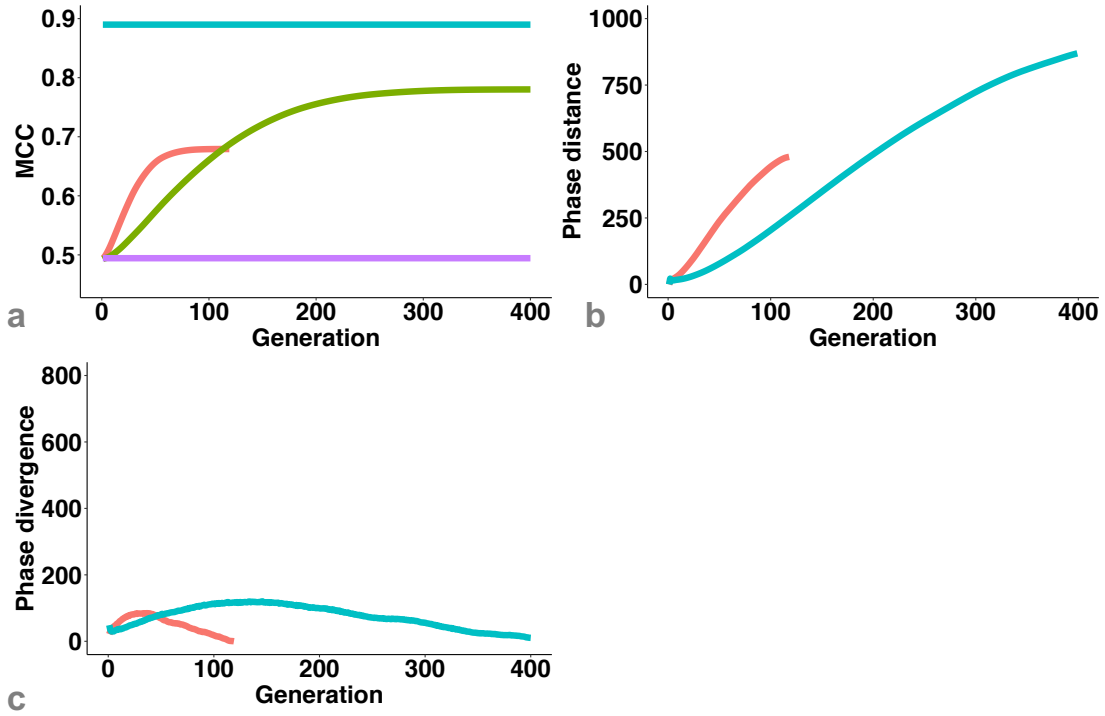


Figure 44 Comparison of GA design 2a and GA design 2b to show the effect of tournament size. The improvement of MCC (Figure 44a), phase distance (Figure 44b) and phase divergence (Figure 44c) for phase variability of 1° in the test case II are plotted as a function of generation. * colour legend for Figure 44a ** colour legend for Figures 44b, 44c.

* Design 1 Design 2b First parent Final point

** Design 1 Design 2b

With the size 2 tournament having triangular probability distribution (refer to section "Tournament Selection" of chapter 3), a comparatively higher percentage of poorly performing members with a variety of genetic composition were retained. This allowed enrichment of genetic information for further evolution (Figure 44b). With the fitness function, MCC, driving towards the correct minimum in phase space, the improvement

in the MCC increased from 45.2% using GA design 2a to 72.3% using GA design 2b. This was achieved in the test case II with 1° phase variability at generation 400 (Figure 44a).

Mutation

To further improve the map quality, different types of static, dynamic and directed mutations were introduced in the best design identified: GA design 3. A characteristic (decrease in MCC when mutations were applied, a period of no change in MCC after mutations were turned off, followed by recovery period showing improvement in MCC, Zoomed-out in Figure 45a) deviation in the improvement of MCC due to mutations can be seen between generation 150 to 200. However, the pattern of this characteristic deviation in the MCC was found be different in static and dynamic mutations.

In static mutations, the longer *jump length* (the number of generations in which the deviation from the linear growth in MCC was observed, shown as a horizontal black line in Figure 45a) was observed due to a constant rate of mutations introduced over selected number of generations (Table B.2) compared to dynamic mutations. This resulted in a slower recovery time for static mutations than dynamic mutations (Figure 45a). The *jump depth* (the amount of change in MCC, shown as vertical black line in Figure 45a) was almost similar with static and dynamic mutations (Figure 45) as the same mutation rate was introduced in both the cases (Table B.2). In test case II using GA design 3, the improvement in MCC from 0.4944 to 0.8293 (84.7%) with static mutations and to 0.8326 (85.5%) with dynamic mutations respectively was observed at generation 400 for the population with 1° phase variability (Figure 45a). A comparatively higher improvement in MCC with dynamic mutations reflects the positive effect of a smaller mutation rate at late developmental stages. This can be seen as more beneficial than no mutations or mutations with high mutation rate (Figure 45). A heavy mutation rate in the early developmental stages was found be detrimental for the improvement of optimisation problem (Figure C.2). However, a smaller mutation load at the early developmental stages needs to be investigated.

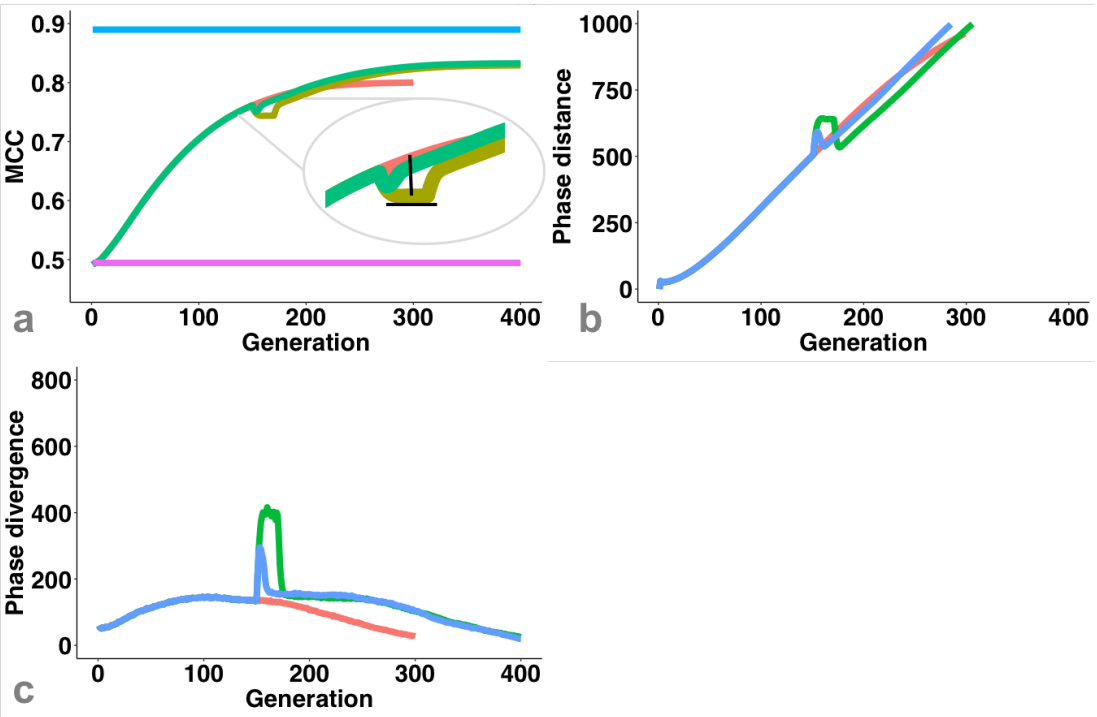


Figure 45 Comparison of GA design 3a, GA design 3b and GA design 3c to show the effect of mutations. The growth of MCC (Figure 45a), phase distance (Figure 45b) and phase divergence (Figure 45c) for phase variability of 1° in the test case II are plotted as a function of generation. The *jump depth* (vertical black line) and *jump length* (horizontal black line) in the improvement of MCC due to mutations are highlighted in Figure 45a. * colour legend for Figure 45a ** colour legend for Figures 45b, 45c.

Nextgen: Directed Mutation

In GAs, various factors influence the permutation probability of genes. Certain reflections are more permuted than other reflections. This can be due to selective preference exerted by the fitness function towards certain gene values. For example, the possibility of a higher preference to the reflection's phase value of "0" when skewness is used as a fitness function. This can also be due to the use of non-uniformly distributed random numbers for implementing various parameters. In this work, a uniform randomiser was used and no selective preference to a specific phase value

due to fitness function (e.g. skewness) was observed (Table B.3). Yet, the non-uniform permutation frequency of different reflections was observed in the population generated from test case II using GA design 3 (Figure 46).

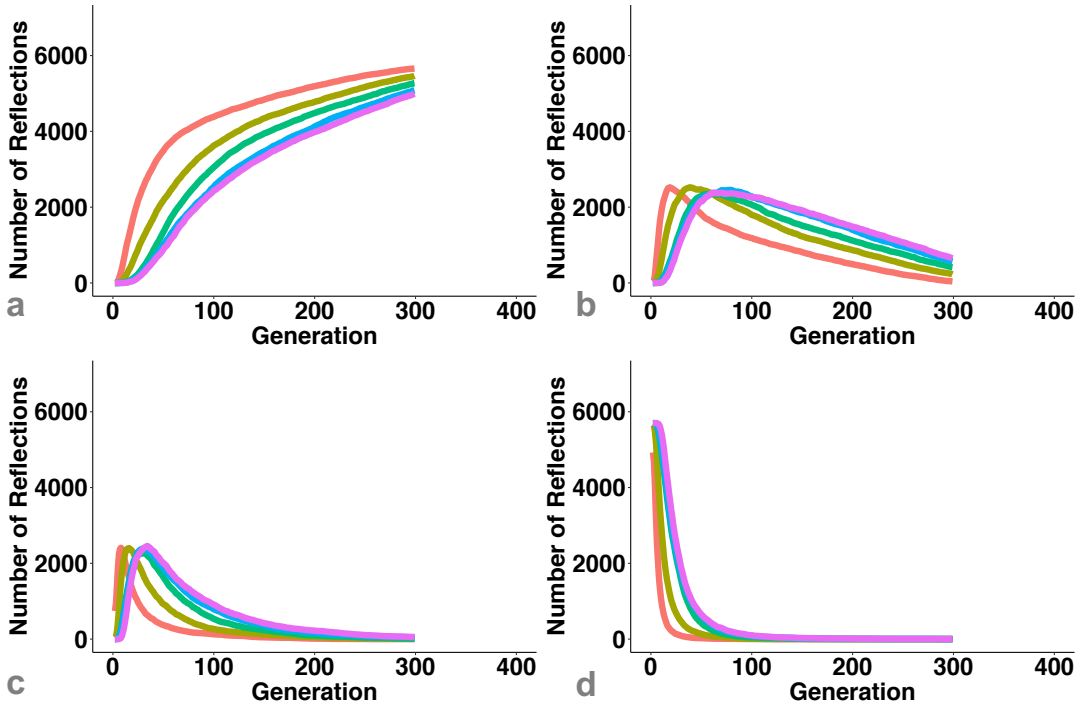


Figure 46 Reflections statistics for the test case II using GA design 3. The growth of "completely degenerate" (Figure 46a), "moderately degenerate" (Figure 46b), "slightly degenerate" (Figure 46c), and "non-degenerate" (Figure 46d), all phase variabilities are plotted as a function of generation.

This behaviour can be described as a process of convergence. When a system finds the best value for a gene/reflection, an ideal system would promote that value over generations without a change until it finds a better one. However, this can also happen in the presence of hitchhikers: an adjacent gene that has reached convergence and stayed constant could cause the hitchhiker gene to also stay constant without reaching convergence as it is promoted along with the adjacent gene. This can also lead to *premature convergence*.

To understand this behaviour, the individual reflections were categorised as "completely degenerate", "moderately degenerate", "slightly degenerate", and "non-

degenerate" (for definitions refer to section "Directed mutations" of chapter 3) and their occurrences over 300 generations were computed for test case II in GA design 3 (Figure 46). In this analysis, a constant increase in the "completely degenerate" reflections over 300 generations (Figure 46a) while a drastic decrease in the "non-degenerate reflections" in less than 100 generations (Figure 46d) was observed.

To prevent this early decrease in the frequency of different types of reflections studied, these reflections were mutated by applying high mutation rate to "completely degenerate" reflections and low or no mutation rate to other reflections. The expectation was to shift/delay the peak in Figures 46b and 46c and fall-off time in Figure 46d for few further generations.

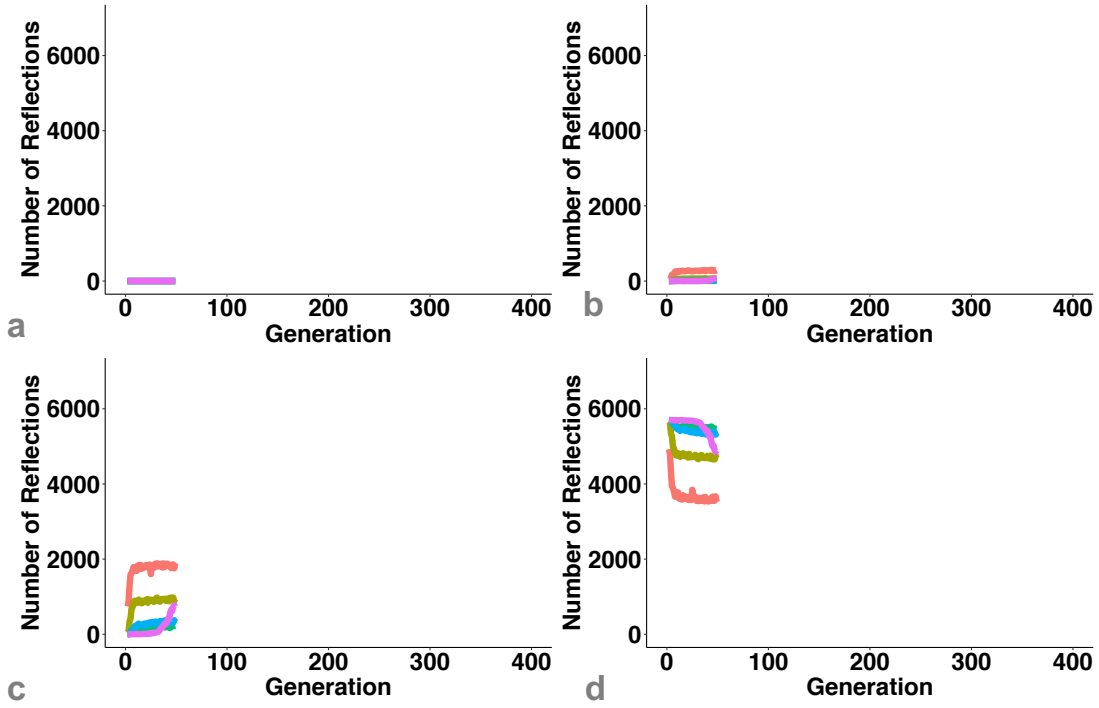


Figure 47 Reflections statistics for the test case II using GA design 3d. The growth of "completely degenerate" (Figure 47a), "moderately degenerate" (Figure 47b), "slightly degenerate" (Figure 47c), and "non-degenerate" (Figure 47d) for all phase variabilities are plotted as a function of generation.

This was achieved by implementing a decremented (mutation rate was decremented by 0.01 per generation until it reaches zero) directed mutations. The mutation rate of

0.015 (1/3 of the phase divergence of the population with a 4° phase variability (Table B.2) was applied non-uniformly to four different types of reflections. A non-linear growth of the completely degenerate reflections (Figure 47a), delay in the peak formation for moderately and slightly degenerate reflections (Figure 47b and 47c) and delay in the fall-off time for non-degenerated reflections was observed (Figure 47d).

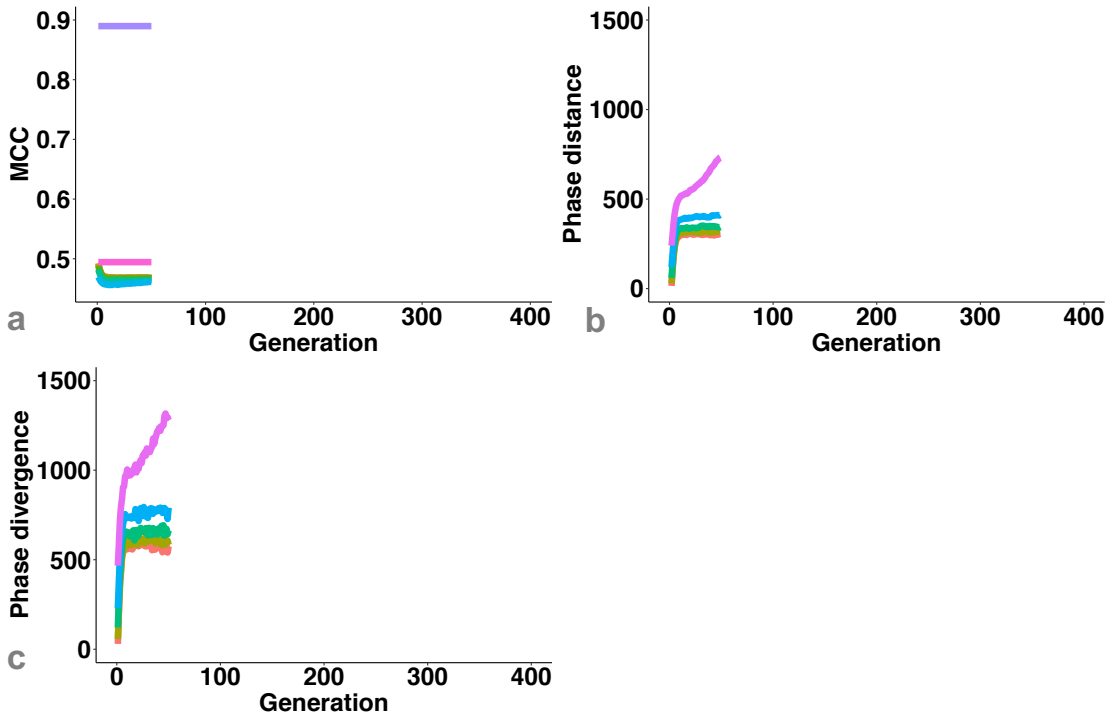


Figure 48 The effect of directed mutations. The growth of MCC (Figure 48a), phase distance (Figure 48b) and phase divergence (Figure 48c) for all phase variabilities in the test case II using GA design 3d are plotted as a function of generation. * colour legend for Figure 48a ** colour legend for Figures 48b, 48c.

* 0.5° 1° 2° 4° 8° First parent Final point

** 0.5° 1° 2° 4° 8°

However, this introduced a very high diversity within the population. The increase in the cluster size (phase divergence, Figure 48c) and shift in the cluster locus (phase distance, Figure 48b) compared to GA design 3a (Figure 48d and Figure 48d respectively) was observed. The improvement in the MCC, for all phase variabilities was found to be less than the first parent (Figure 48a). This could be due to the use of a high mutation rate (mutation rate of 0.015).

To identify the appropriate mutation rate that introduces the required diversity without spreading out the population too much in the phase space, different mutation rates of 0.0005, 0.001, 0.002, 0.004 were tested. These mutations rates were kept as low as possible. The goal was to first identify the mutation rate that does not disturb the system too much, and then arrive at the best mutation rate by adding small increments.

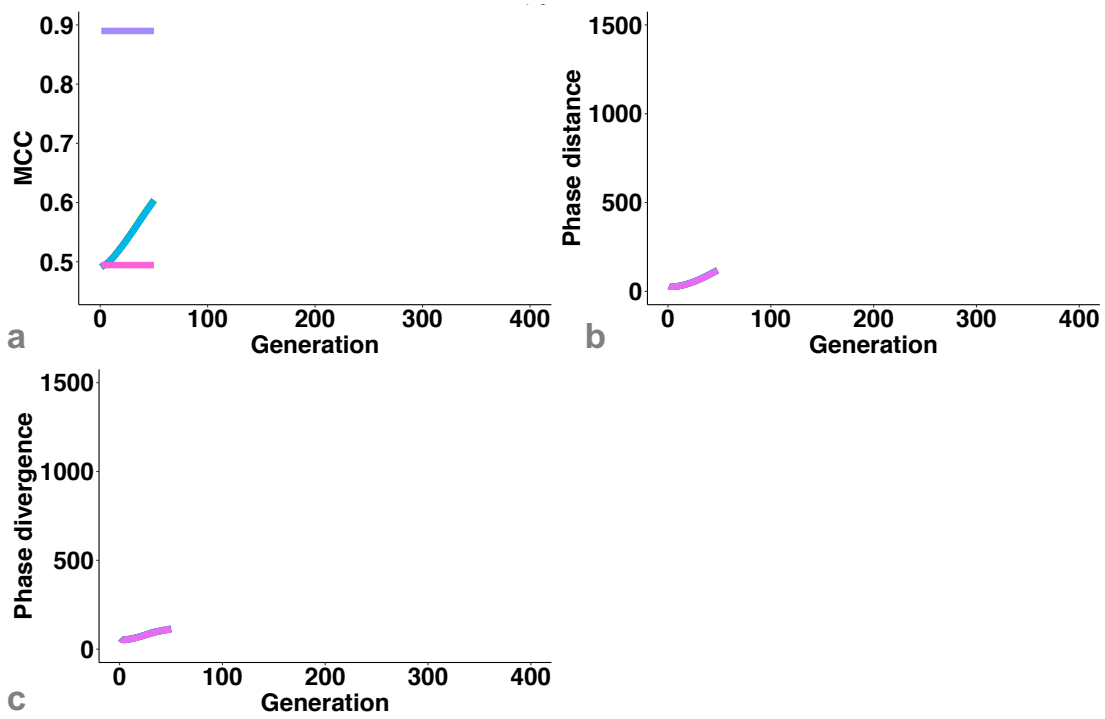


Figure 49 The study of base mutation rate. The growth of MCC (Figure 49a), phase distance (Figure 49b) and phase divergence (Figure 49c) for phase variability of 1° in the test case II using GA design 3d are plotted as a function of generation. * colour legend for Figure 49a ** colour legend for Figures 49b, 49c.

Therefore, the four mutation rates identified were tested in the test case II using GA design 3d with 1° phase variability. The growth pattern of MCC, phase distance, and phase divergence was similar to the GA run using GA design 3a without mutations (Figure 49). The reflection distribution pattern was also found to be similar (Figure 50). Using 0.004 as the base, different mutations rates needs to be identified by adding

small increments to it. The further fine tuning of the parameters should be pursued in the future.

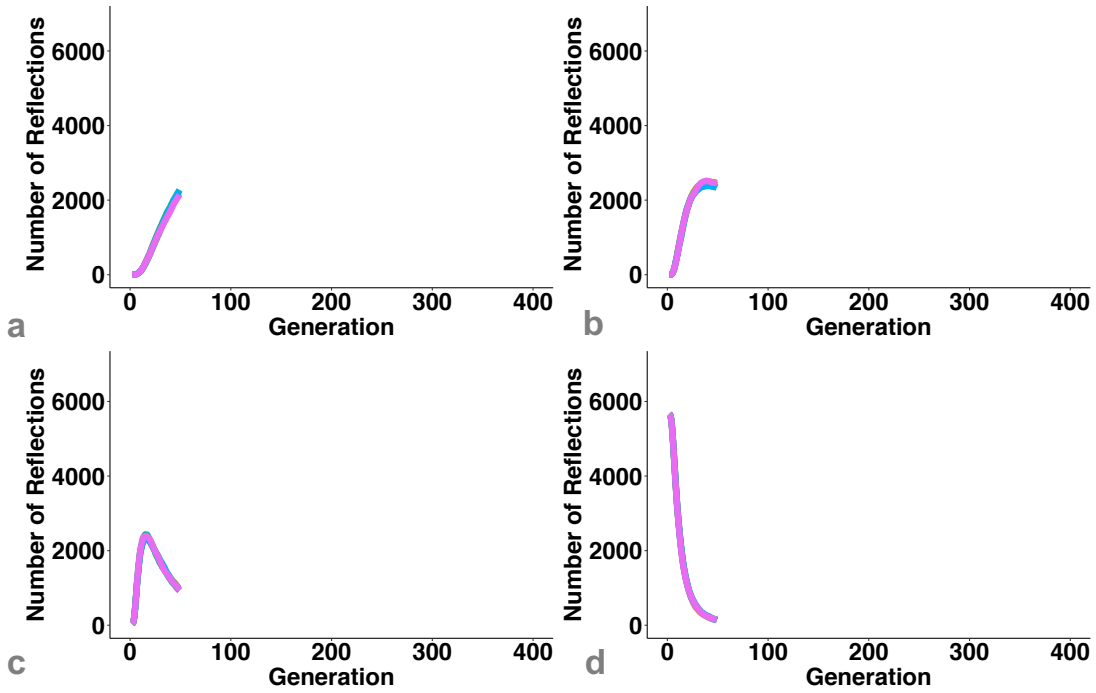


Figure 50 Reflections statistics for the test case II using GA design 3d. The growth of "completely degenerate" (Figure 50a), "moderately degenerate" (Figure 50b), "slightly degenerate" (Figure 50c), and "non-degenerate" (Figure 50d), for phase variability 1° are plotted as a function of generation.

Diversity in Starting Population

The features in the growth pattern of all phase variabilities in all GA designs were found to be related (Figures 51, 52 and 53). For example, these features in the MCC growth curves can be characterised as: a stem with nearly straight line showing overlapping growth pattern and diversification of the growth after inflection point that can be seen as branches. This is called "trajectory deflection" in this work. This pattern was found to be triggered by the drop in diversity (Figure 53). In designs 1 and 2a, due to comparatively low diversity this *trajectory deflection* started in less than 50 generations (Figure 51a and 51b respectively). While in designs 2b, 3a, 3b and 3c, the *trajectory deflection* started after 100 - 150 generations (Figure 51c, 51d, 51e and 51f

respectively). Among the five different phase variabilities, the phase variability of 1° showed greatest improvement in the MCC in GA design 1 and 2a. For GA designs 2b and 3 (a, b and c), the phase variability of 8° and 4° proved to be good starting points (Figure 51). The sudden jumps in the Figures 51e, 52e, 53e, and 51f, 52f, 53f are due static and dynamic mutations respectively.

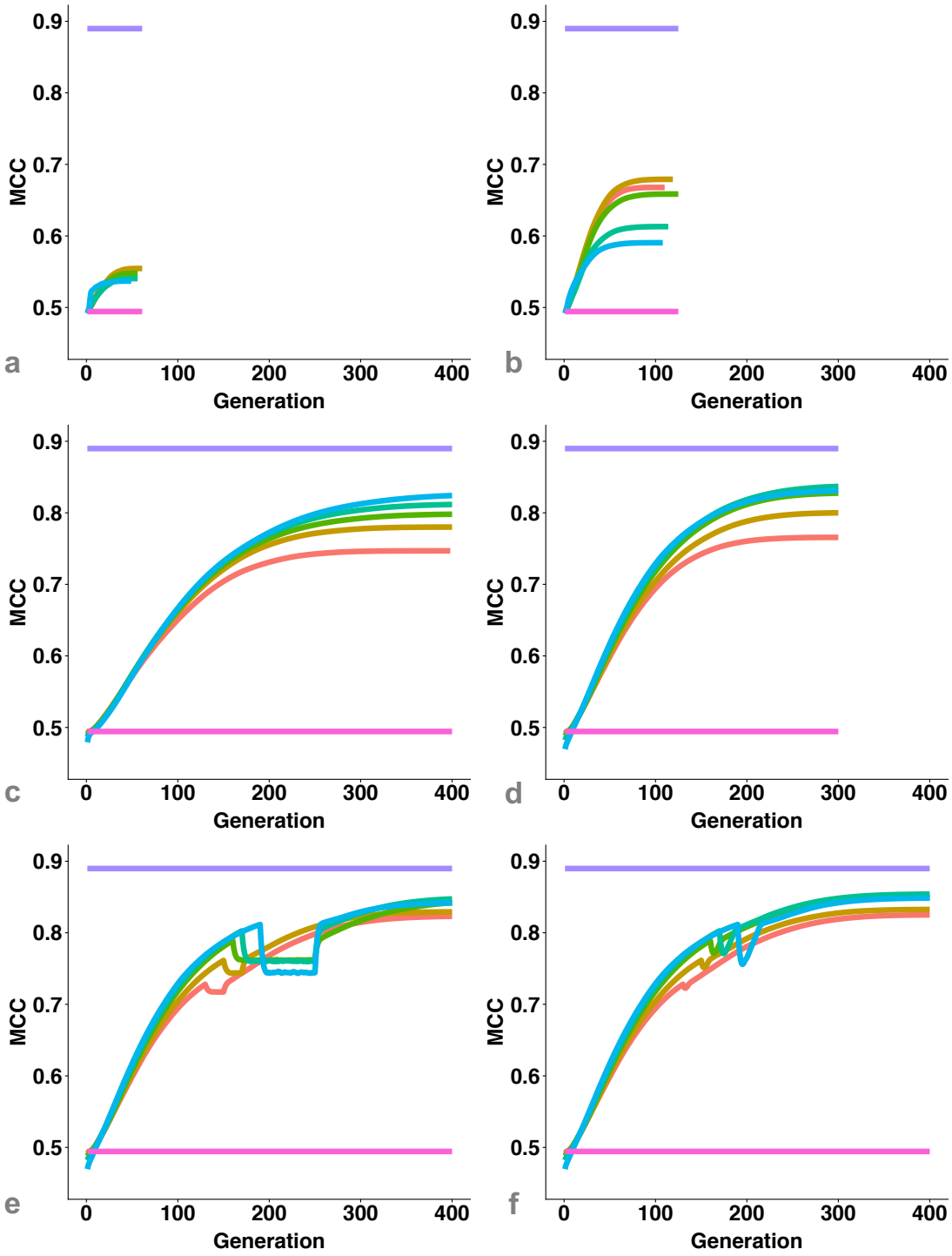


Figure 51 The growth of MCC in GA design 1 (Figure 51a), GA design 2a (Figure 51b), GA design 2b (Figure 51c), GA design 3a (Figure 51d), GA design 3b (Figure 51e), GA design 3c (Figure 51f) for all the phase variabilities in the test case II.

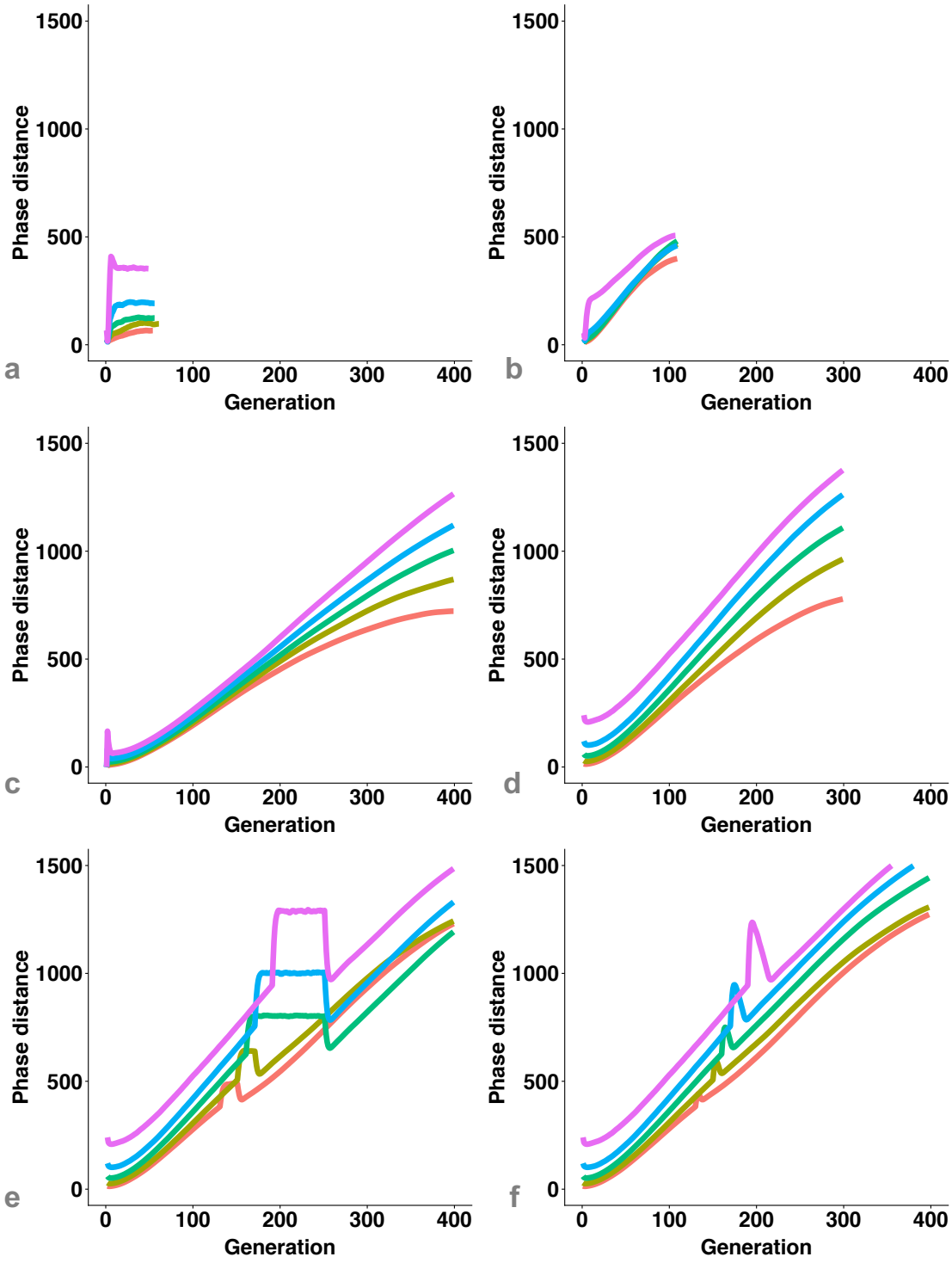


Figure 52 The growth of phase distance in GA design 1 (Figure 52a), GA design 2a (Figure 52b), GA design 2b (Figure 52c), GA design 3a (Figure 52d), GA design 3b (Figure 52e), GA design 3c (Figure 52f) for all the phase variabilities in the test case II.

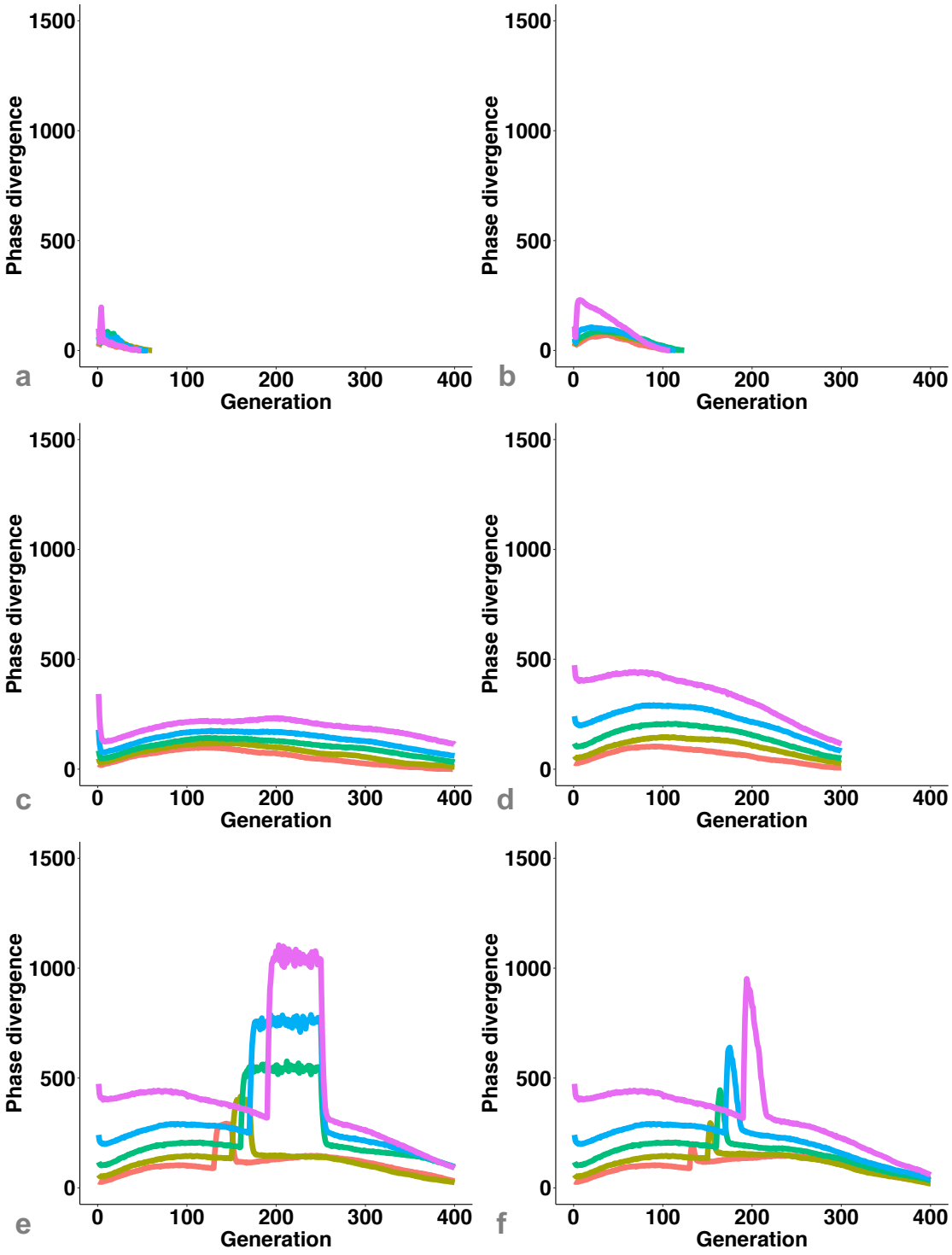


Figure 53 The growth of phase divergence in GA design 1 (Figure 53a), GA design 2a (Figure 53b), GA design 2b (Figure 53c), GA design 3a (Figure 53d), GA design 3b (Figure 53e), GA design 3c (Figure 53f) for all the phase variabilities in the test case II.

In test case I, the stem and *trajectory deflection* pattern of growth in MCC was found to be less noticeable than test case II. Here in design 1, the *trajectory deflection* observed in less than 25 generations (Figure 54b). The *trajectory deflection* for design 2a, 2b and 3 was observed in less than 50, 120 and 150 generations (Figure 54b, 54c and 54d, respectively). In this test case, GA design 3a is referred to as GA design 3 for simplicity.

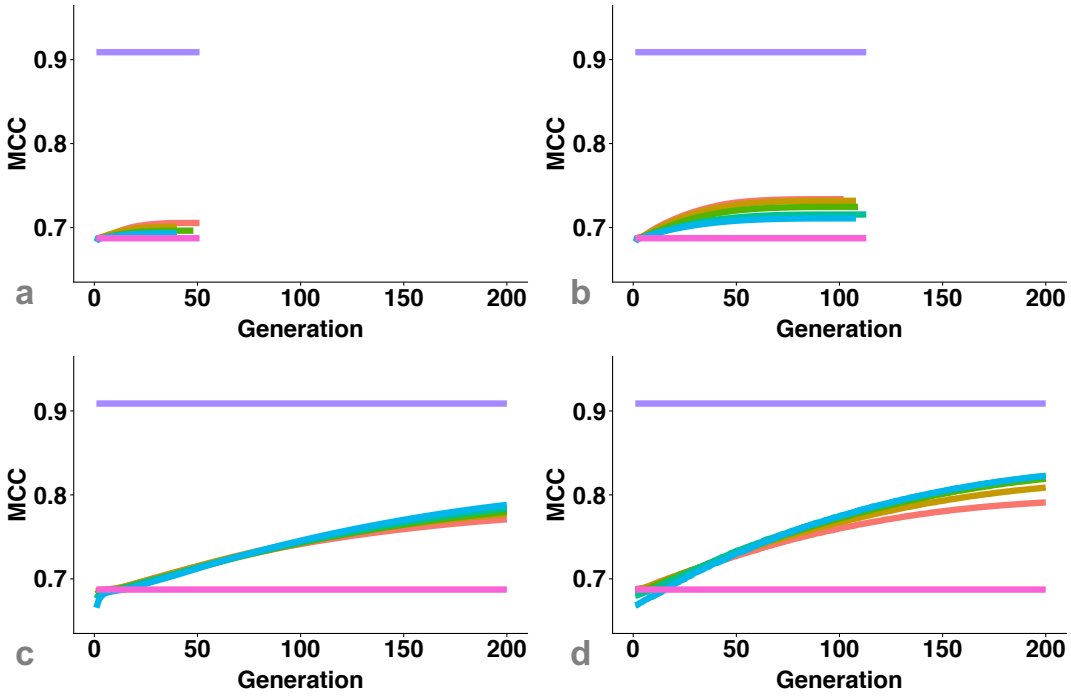


Figure 54 The growth of MCC in GA design 1 (Figure 54a), GA design 2a (Figure 54b), GA design 2b (Figure 54c), GA design 3 (Figure 54d) for all the phase variabilities in the test case I.

Similar to test case II, in this test case I, phase variabilities: 0.5° and 1° showed the best performance in GA design 1 and 2a while phase variabilities: 8° and 4° in GA designs 2b and 3 (Figure 54).

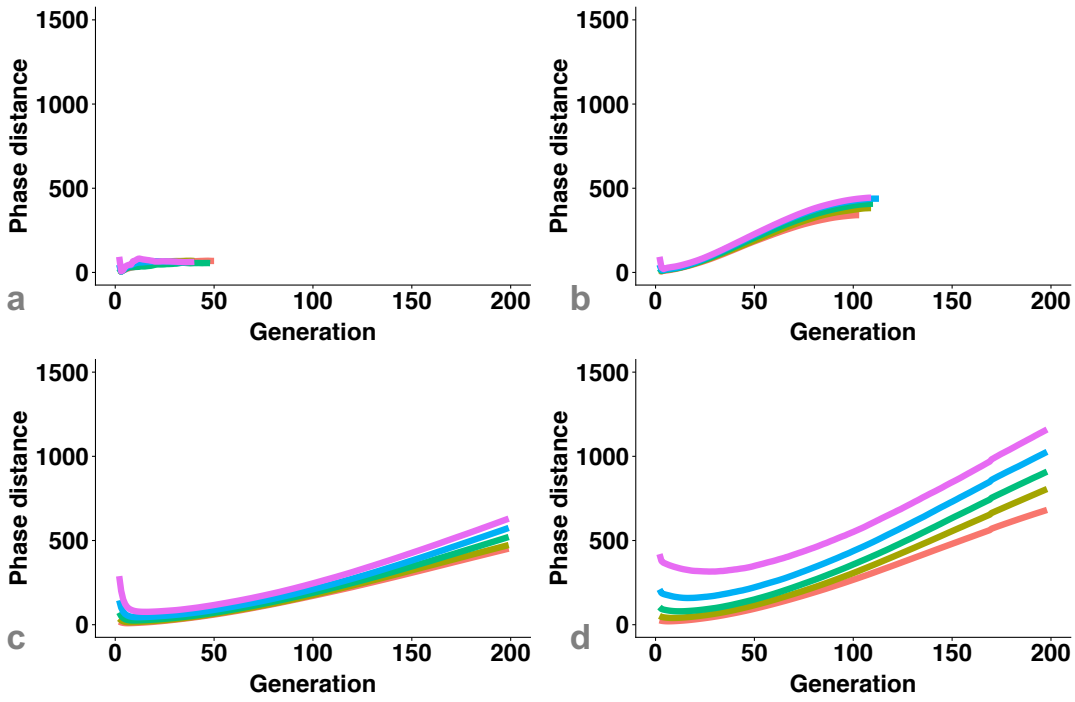


Figure 55 The growth of phase distance in GA design 1 (Figure 55a), GA design 2a (Figure 55b), GA design 2b (Figure 55c), GA design 3 (Figure 55d) for all the phase variabilities in the test case I.

The relatable features in the growth patterns of the phase distance (Figure 55) and the phase divergence curves (Figure 56) were found in test case I as well. This confirms the dependence of phase variability on the diversity of the population.

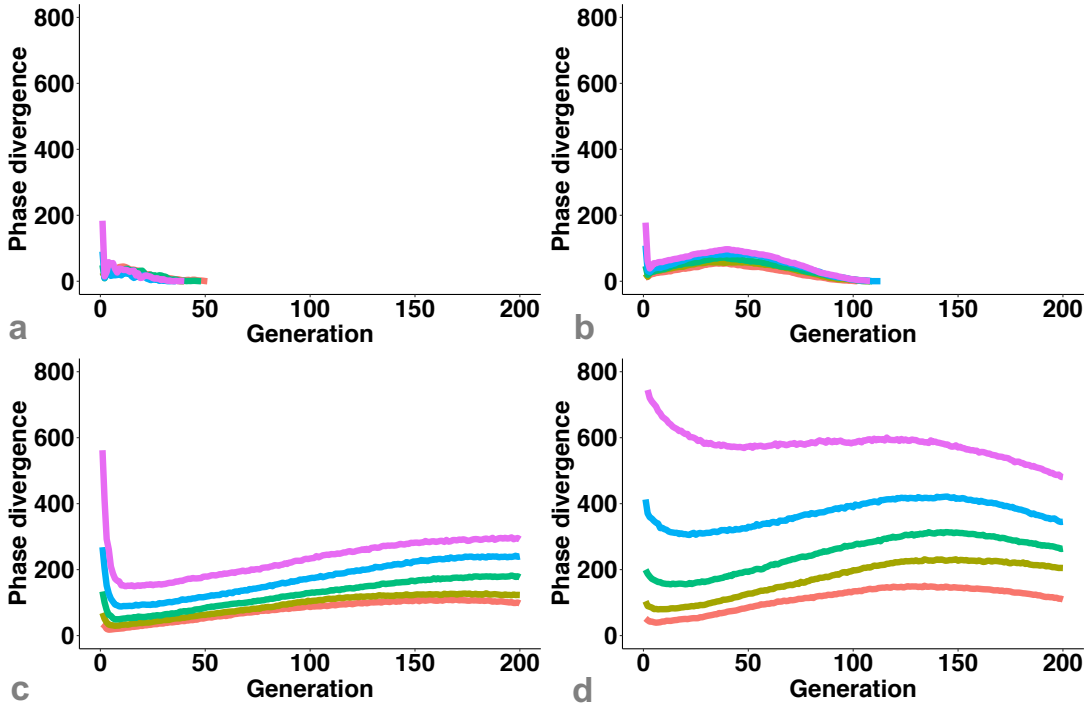


Figure 56 The growth of phase divergence in GA design 1 (Figure 56a), GA design 2a (Figure 56b), GA design 2b (Figure 56c), GA design 3 (Figure 56d) for all the phase variabilities in the test case I.

Improvement in MCC

The convergence for design 1, 2a 2b, 3 was observed at approximately 20, 50, 150, 180 generations respectively. From design 1 to design 3, the convergence was delayed by nearly 200 generations, allowing the population to develop in the right direction. GA design 2b did not show reasonably improvement in the MCC, although it is much better than GA design 1.

Considerable improvement in the MCC from 0.4944 to 0.8117 (80.2%) for GA design 2b at generation 400 and to 0.8369 (86.6%) for GA design 3 at generation 300, was observed, for phase variability of 4° in test case II (Figure 57a). Using static mutations, this improvement in the final MCC was pushed to 0.8474 (89.2%) and with dynamic mutations to 0.8540 (90.9%) at generation 400. In this test case, the MCC of final GA population was very near to the *final point* of GA which is 0.8898 to the final map. The

phase error decreased from 52.76° to 38.30° in GA design 3c at generated 400 (Figure 57d).

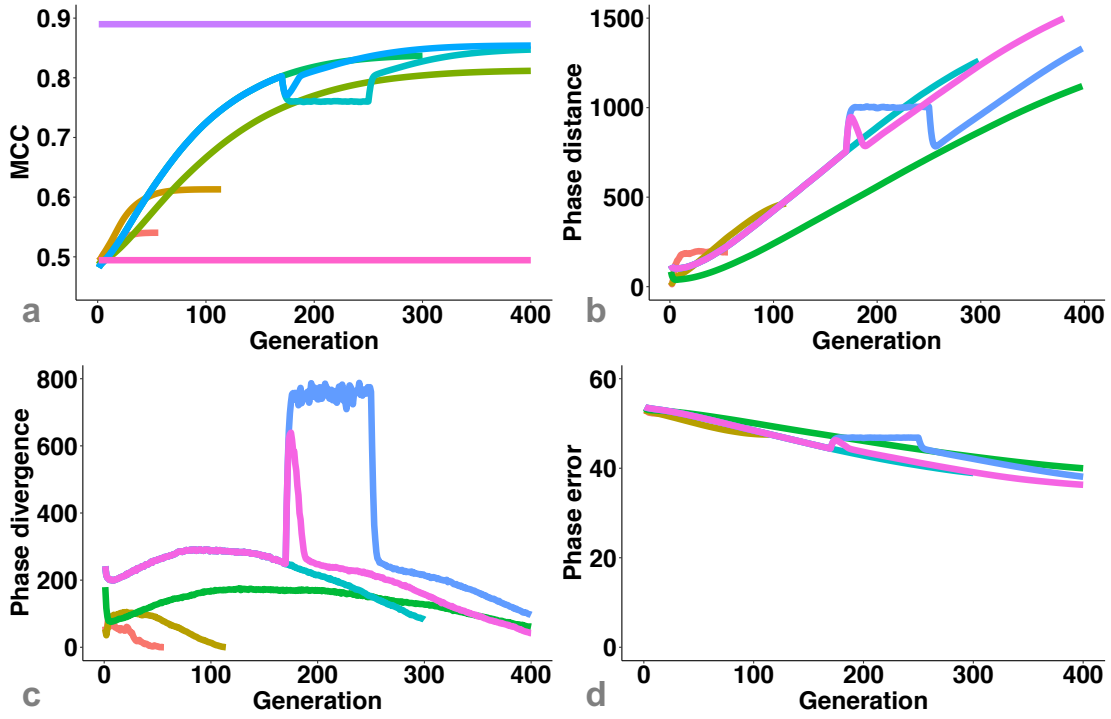


Figure 57 The comparative performance of different designs of GA in the test case II. The growth of MCC (Figure 57a), phase distance (Figure 57b), phase divergence (Figure 57c), phase error (Figure 57d), in all designs of GA for phase variability of 4°. * colour legend for Figure 57a ** colour legend for Figures 57b, 57c, 57d.

For test case I, the improvement in MCC from 0.6873 to 0.7839 (43%) for GA design 2b at generation 400 and to 0.8226 (61%) for GA design 3 at generation 300 was observed, for phase variability of 4° (Figure 58a). The *final point* of GA in this test case is having an MCC of 0.9088 to the final map. The rate of improvement in MCC was noticeably faster in GA design 3 than GA design 2b (Figure 58a).

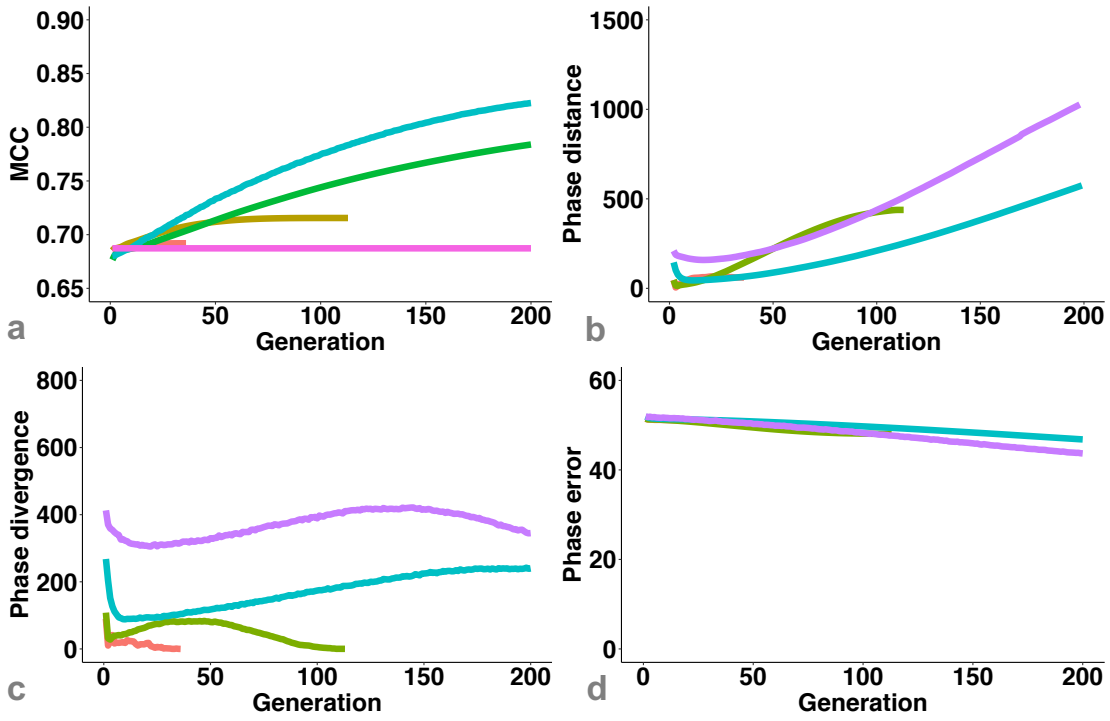


Figure 58 The comparative performance of different designs of GA in the test case I. The growth of MCC (Figure 58a), phase distance (Figure 58b), phase divergence (Figure 58c), phase error (Figure 58d), in all designs of GA for a phase variability of 4°. * colour legend for Figure 58a ** colour legend for Figures 58b, 58c, 58d.

Improvement in Model Quality

The improvement in the number of residues built by ARP/wARP during model building from initial to final solution in all designs of GA using all phase variabilities for test case II are shown in Figure 59. The number of residues built for test case II was increased from 50 residues in GA design 1 to more than 185 residues in GA designs 3a, 3b and 3c (Figure 59). The model improved from complete random fragments to full structure (Figure 60). The final structure built from the map generated by GA design 3c aligned perfectly with the structure deposited in the PDB (Figure 61). The alignment of the GA map with density modified structure of the GA map and the PDB structure shows little or no need for further density modification (Figure 62).

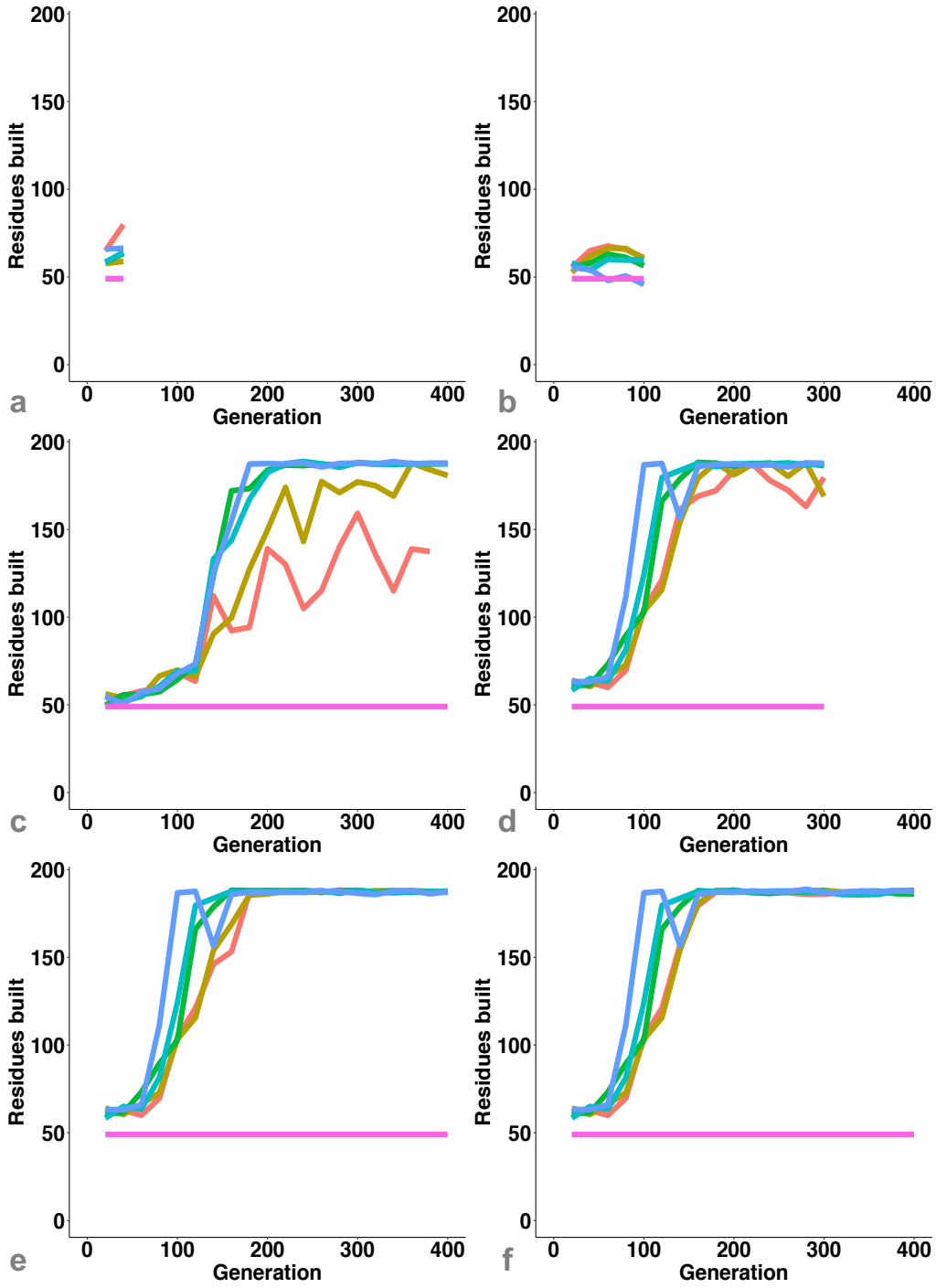


Figure 59 Improvement in the number of residues built. The growth of number of residues built by ARP/wARP from the final map generated by GA design 1 (Figure 59a), GA design 2a (Figure 59b), GA design 2b (Figure 59c), GA design 3a (Figure 59d), GA design 3b (Figure 59e), GA design 3c (Figure 59f) for all phase variabilities in the test case II.

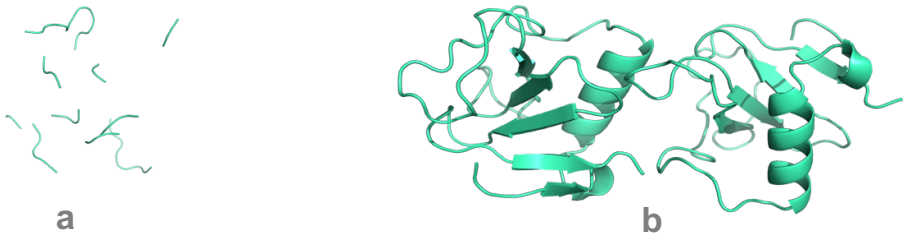


Figure 60 Improvement in the model quality. Figure 60a shows the model built from the *GA starting point* for the test case II. Figure 60b show the model built from the map generated by GA design 3c at generation 400 for the test case II.

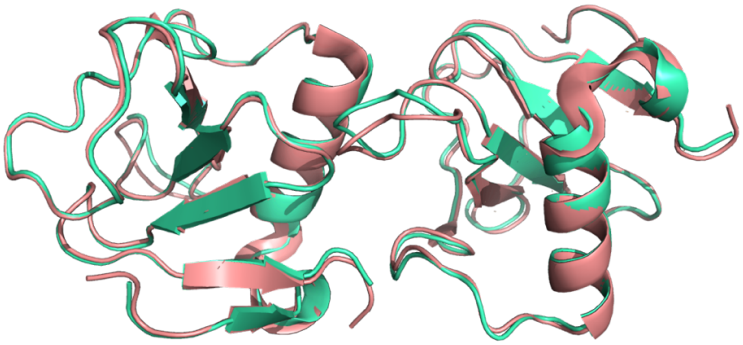


Figure 61 Alignment of model built from the map generated by GA design 3c at generation 400 (green cyan) with PDB structure (salmon red) of the test case II.

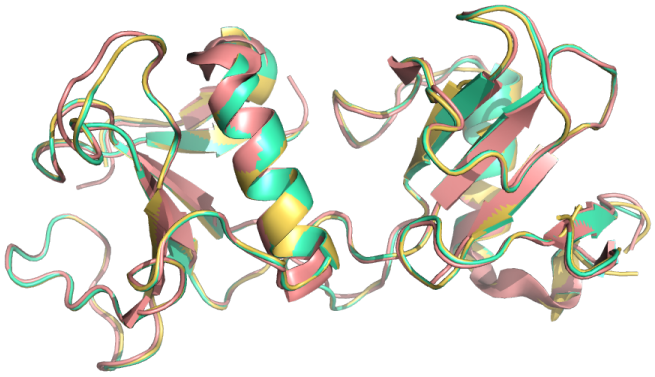


Figure 62 Alignment of model generated from the GA* map (green cyan) with PDB structure (salmon red) and model generated from the GA* map after density modification (gold) of the test case II.

* GA map generated by using the GA design 3c at generation 400.

Map Characteristics as a Fitness Function

Map Moments – Skewness and Kurtosis

A study performed on ribonuclease from *Streptomyces aureofaciens* to investigate the relationship between map moments and the phase error provided an important insight for the current study (Lamzin 2013). The correlation between skewness and kurtosis of the electron density map with the phase error was monitored for test case II. For this study, the X-ray data were truncated to 2.0 Å and 2.5 Å and phases were taken without phase discretisation. Starting with these initial phases having 0° phase error, a uniform phase error of 0.09° was introduced incrementally until an average phase error of 90° was achieved. The skewness and kurtosis were computed at each increment of the phase error. These results indicate that the poorer the phases, the lower the skewness and kurtosis of the density map is (Figure 63).

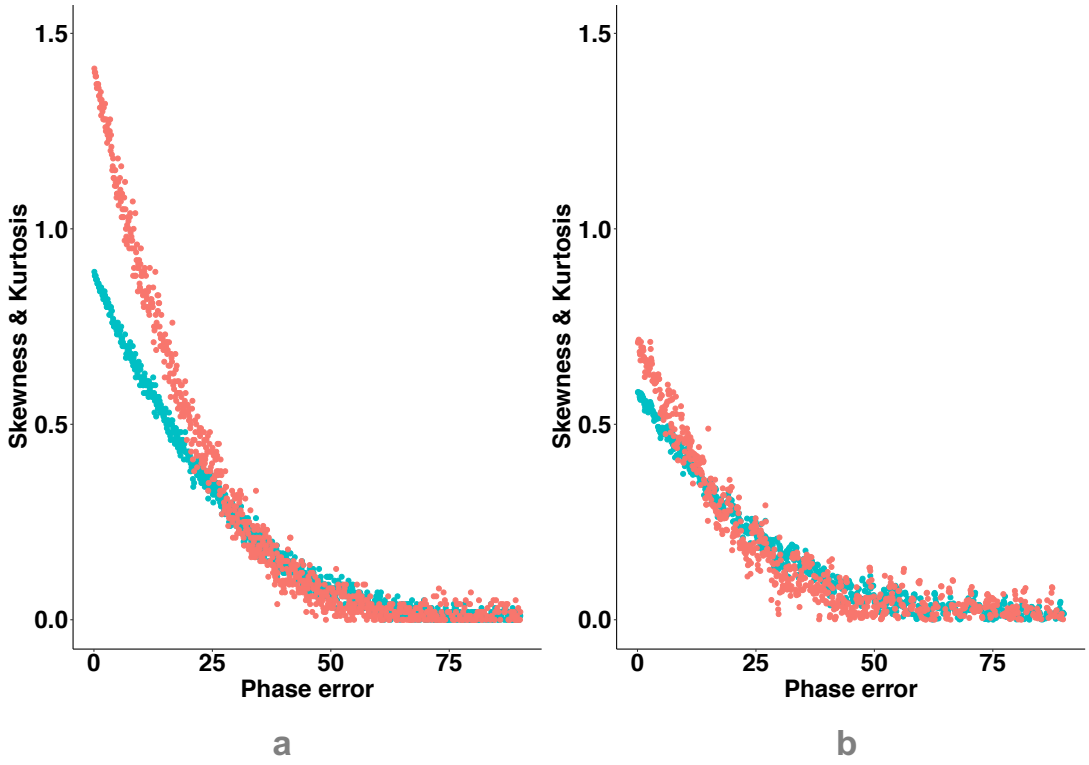


Figure 63 The correlation (In test case II) of skewness (cyan) and kurtosis (red) to the phase error at resolution of 2.0 Å (Figure 63a) and 2.5 Å (Figure 63b).

As skewness and kurtosis correlate with the phase error inversely, these two parameters and their combinations were studied in this work for their applicability as a fitness function in the phase optimisation with GA. The skewness and kurtosis were combined using the following equation:

$$c_1 = wg_1 + (1 - w)g_2 \quad (17)$$

where g_1 is skewness, g_2 is kurtosis and w is 0.8. The weighting factor of 0.8 was decided based on a study performed to estimate the effectiveness of different weighting factors: 0, 0.1, 0.2…0.8, 0.9, 1. The performance of these weights was tested on a dataset having 10 groups of phase sets with an overall phase error of 0° to 90°. Each group of phase sets differed by a sub-range of phase error. For example, group 1 had phase sets with a phase error in the range of 0° to 9° and group 10 in the range of 82° to 90°. After evaluating the quality of these phase sets by using equation (17) with all weighting factors from 0 to 1, the maximum fitness value was identified for each group (highlighted in red in Table 9). The maximum fitness value in most of these groups was observed when the weighting factor of 0.8 was used (Table 9). Therefore, 0.8 was selected as a weighting factor.

Table 9 The performance of different weighting factors in different phase error groups.

Weight	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9	Group 10
0.0	0.292	0.461	0.309	0.380	0.371	0.278	0.429	0.441	0.274	0.371
0.1	0.332	0.510	0.349	0.413	0.415	0.316	0.461	0.488	0.316	0.412
0.2	0.373	0.558	0.390	0.445	0.459	0.356	0.492	0.536	0.359	0.451
0.3	0.412	0.604	0.430	0.475	0.502	0.395	0.520	0.580	0.401	0.489
0.4	0.449	0.645	0.466	0.500	0.540	0.430	0.544	0.620	0.439	0.523
0.5	0.480	0.677	0.496	0.520	0.573	0.459	0.561	0.652	0.472	0.550
0.6	0.504	0.700	0.518	0.533	0.598	0.481	0.573	0.676	0.498	0.569
0.7	0.520	0.713	0.532	0.539	0.615	0.495	0.577	0.689	0.514	0.580
0.8	0.529	0.716	0.538	0.539	0.624	0.501	0.576	0.694	0.523	0.582
0.9	0.531	0.711	0.538	0.533	0.626	0.500	0.569	0.690	0.524	0.579
1.0	0.528	0.700	0.532	0.523	0.622	0.494	0.558	0.681	0.521	0.570

These three parameters: skewness, kurtosis and combination of skewness and kurtosis were evaluated for their relative performance in selecting best phase sets. This was achieved in three steps. In the first step, a certain fraction (1/2 to 1/9) of phase sets having high MCC were marked before selection. In the second step, selection was performed using skewness, kurtosis and a combination of skewness and kurtosis. In the last step, the percentage of solutions (or phase sets) picked up by these fitness functions that were marked as having high MCC was identified. This efficiency (percentage of good solutions picked up by the fitness function) was then plotted as a function of the fraction of phase sets marked before selection, Figure 10. In this study, skewness performed much better than kurtosis (Figure 64). The performance of skewness alone was almost identical to that of the combination of skewness and kurtosis (Figure 64). Hence, skewness alone was used as a fitness function in the subsequent GA studies.

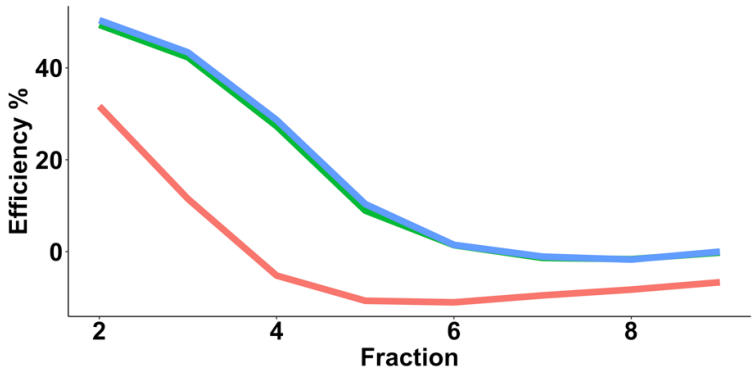


Figure 64 The efficiency of skewness (blue), kurtosis (red) and a combination of skewness and kurtosis (green) in identifying the quality of the phase set.

Premature Convergence

The skewness as a fitness function was initially implemented in GA design 1 with one-point crossover variant 2. The phase variabilities tested were: 0.5°, 0.6°, 1° and 5°. A very small improvement in the MCC was observed with a phase variability of 0.6° (Figure 65a). The skewness curve tapered off in less than 20 generations for the phase variability of 5° (Figure 65b). A non-linear growth in the skewness was observed for other phase variabilities (Figure 65b). The phase error was higher than the *first parent* for all phase variabilities used (Figure 65c). This early convergence was consistent

with the premature convergence observed with GA design 1 with MCC as a fitness function (Figure 42).

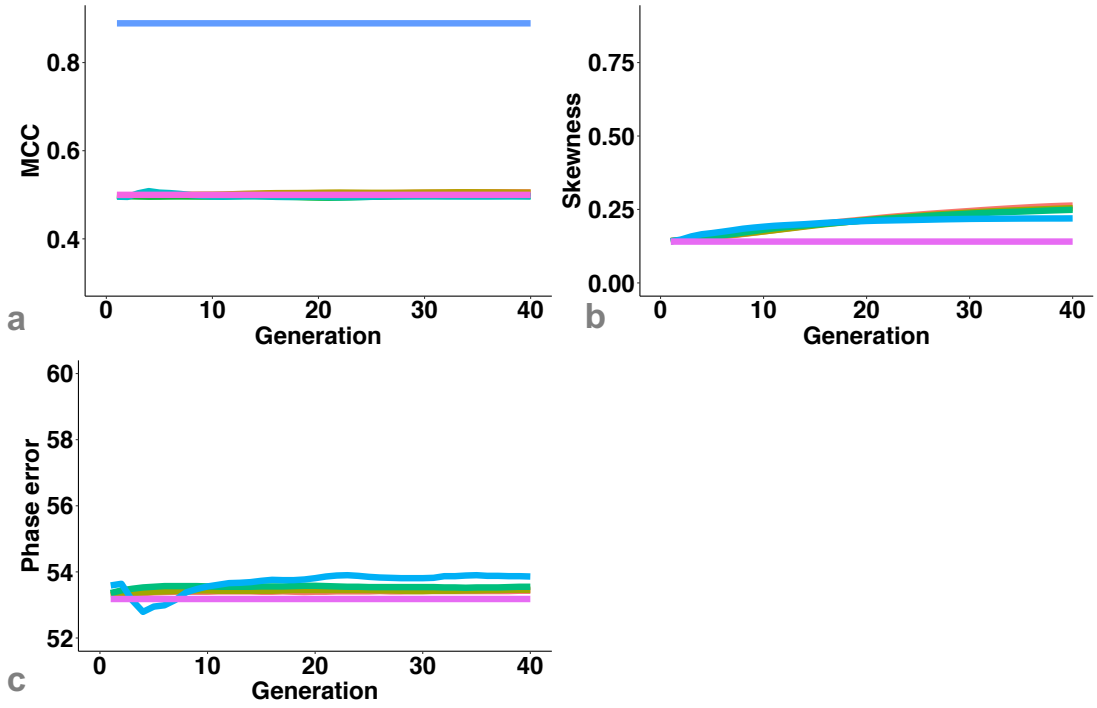
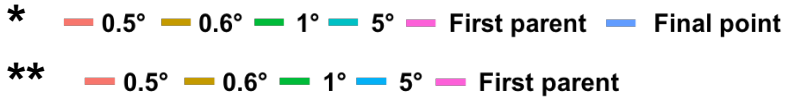


Figure 65 GA design 1 with skewness as a fitness function in the test case II. The growth of MCC (Figure 65a), skewness (Figure 65b) and phase error (Figure 65c) for the phase variabilities: 0.5°, 0.6°, 1° and 5 are plotted as a function of generation. * colour legend for Figure 65a ** colour legend for Figures 65b, 65c.



Crossover

With the skewness as a fitness function, the two variants of the one-point crossover were tested for their relative performance. We ran 20 generations with the first crossover variant and 40 generations with the second variant in test case II. The skewness growth curve showed the signs of curve flattening in both variants (Figure 66b). The MCC started to improve and became better than the *first parent*, but the growth rate was negligible (Figure 66a). However, design 1b (one-point crossover with variant 2) showed a higher improvement in skewness compared to design 1a (one-point crossover with variant 1) (Figure 66b). The design 1b also showed a lower phase error than design 1a at generation 20 (Figure 66c).

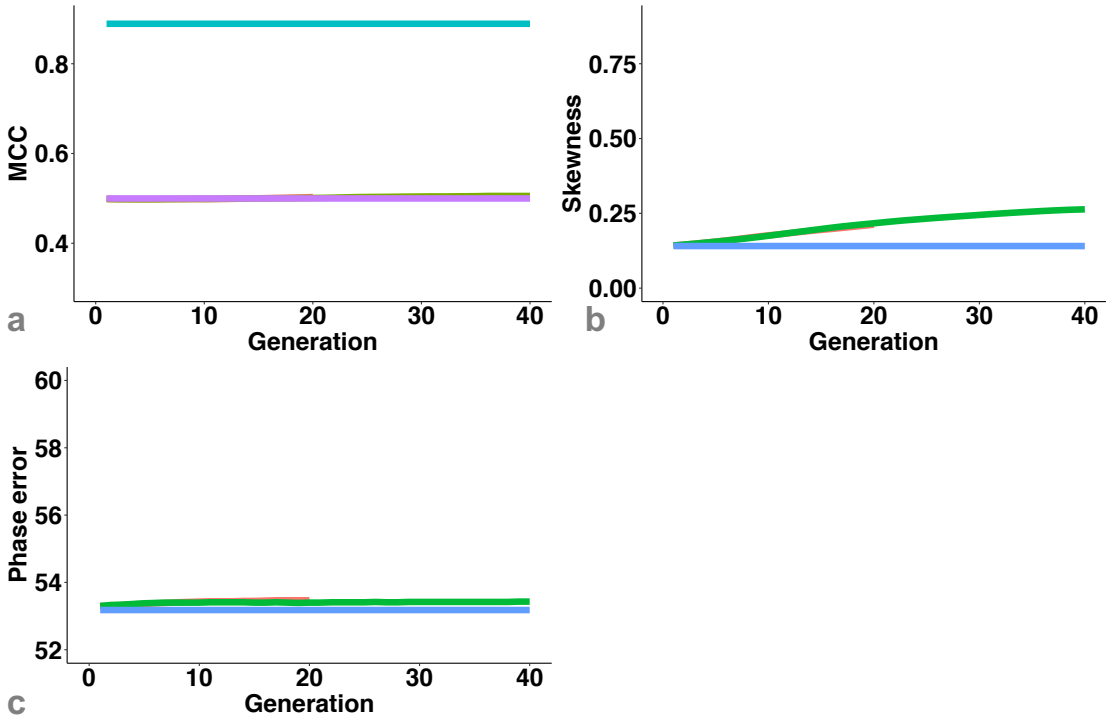


Figure 66 Comparative performance of GA design 1a (one-point crossover with variant 1) and GA design 1b (one-point crossover with variant 2) in test case II using skewness as a fitness function. The growth of MCC (Figure 66a), skewness (Figure 66b) and phase error (Figure 66c) for the phase variability of 0.5° in test case II was plotted as a function of generation. * colour legend for Figure 66a ** colour legend for Figures 66b, 66c.

Selection

With this fitness function, two different selection operators were studied: tournament of size 9 and SUS. Further, these two operators were additionally studied by slightly modifying the population before selection. This modification involved discarding 30% of the population with the lowest fitness value (lowest skewness) before selection. The remaining members were then subjected to selection by using tournament (called “tournament biased” in this work) and SUS (called “SUS biased” in this work). These four selection approaches were tested in test case II using GA design I for four generations.

SUS did not show any improvement in skewness and phase error. SUS-biased selection showed a greater improvement in MCC and phase error (Figure 67a and 67c respectively) but the skewness was much lower than the tournament and the tournament biased selection operators (Figure 67b). Tournament-biased selection showed better improvement in MCC and phase error than a simple tournament. However, the improvement of the skewness was much higher for tournament than tournament biased.

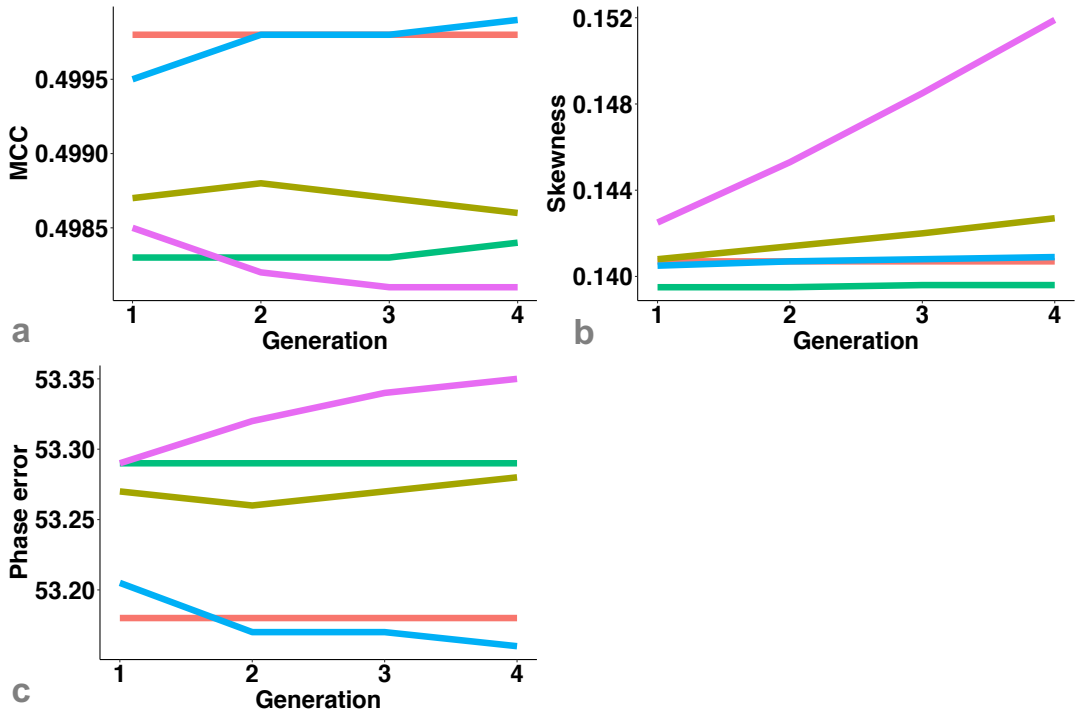


Figure 67 Comparative performance of four different selection approaches. The growth of MCC (Figure 67a), skewness (Figure 67b) and phase error (Figure 67c) in GA design 1a using skewness as a fitness function with the phase variability of 0.5° in the test case II.

The removal of the worst performing members of the population led to a loss of diversity, Figure 40, and did not provide any evidence of considerable improvement in the MCC in other studies (Study S.2). As this approach did not show considerable improvement in all parameters in this study, it was not implemented in GA. Between tournament and SUS, SUS showed peculiar growth behaviour with no change in the

growth of skewness and phase error for four generations and a small increment in MCC after no change in the growth for three generations. This may indicate the insensitivity of the selection operator to the fitness function (skewness) or *vice versa*. Among four selection approaches, a simple (without discarding the 30% worst members) tournament selection operator was used in all GA designs.

In tournament selection, the tournament of size 2 showed greater improvement in MCC when the MCC was used as a fitness function (Figure 68). Therefore, the performances of the tournament selection with size 9 (GA design 2a) and size 2 (GA design 2b) were tested in test case II using skewness as a fitness function. The results were illustrated in Figure 68. The *trajectory deflection* in the growth of MCC and skewness was more prominent in GA design 2a than GA design 2b (Figures 68a, 68b, 68c and 68d). The MCC growth curves showed steady improvement in the opposite direction i.e., improvement lower that the MCC of the *first parent* (Figure 68a). The MCC using design 2b also showed improvement in the opposite direction but the rate of improvement was comparatively smaller than GA design 2a (Figure 68b). The improvement in the skewness was much higher and the growth trajectory was more deflected for GA design 2a than GA design 2b (Figure 68c and 68d). However, a constant growth in skewness with no signs of curve flattening was observed in GA design 2b compared to GA design 2a (Figure 68c and 68d). The phase error was also much lower for all phase variabilities in GA design 2b than GA design 2a (Figure 68e and 68f). Overall, the performance of the GA design 2b was much better than GA design 2a when the skewness was used as a fitness function in the test case II (Figure 68). These results were also consistent with the results obtained by using MCC as a fitness function in the same test case, Figure 44.

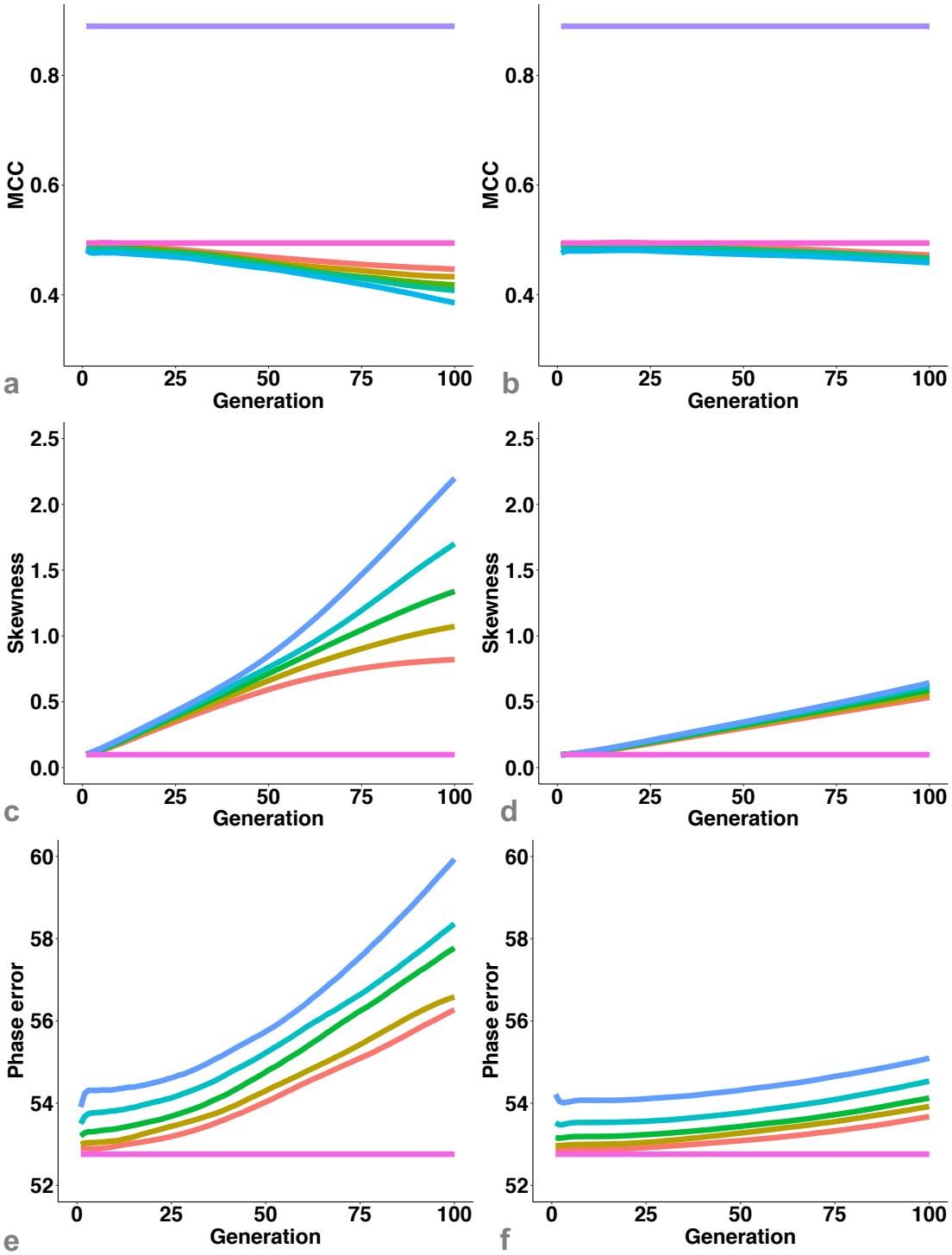


Figure 68 Comparative performance of GA design 2a and GA design 2b when skewness was used as a fitness function in test case II. Figure 68a and 68b shows the comparison of MCC in design 2a and 2b respectively. Figure 68c and 68d shows the comparison of skewness in design 2a and 2b respectively. Figure 68e and 68f

shows the comparison of phase error in design 2a and 2b respectively. * colour legend for Figures 68a and 86b **colour legend for Figures 68c, 68d, 68e and 68f.

* 0.5° 1° 2° 4° 8° First parent Final point

** 0.5° 1° 2° 4° 8° First parent

Intrinsic Processing of Phase Sets by Skewness

The population generated from test case II by using GA design 2a (with tournament size of 9) and skewness as a fitness function produced an abnormal high-density peak, resembling the so-called uranium solution at fractional coordinates: 0.5189 0.8050 0.1165 in survivors at generation 80 (Figure 69a). However, it was observed that the density histogram of the map computed from GA phases shifted or evolved gradually towards the histogram of the map computed with the phases from the refined model (Figure 69b). The distribution of centric and acentric reflections was even found to be without any preference for a specific phase value (Table B.3).

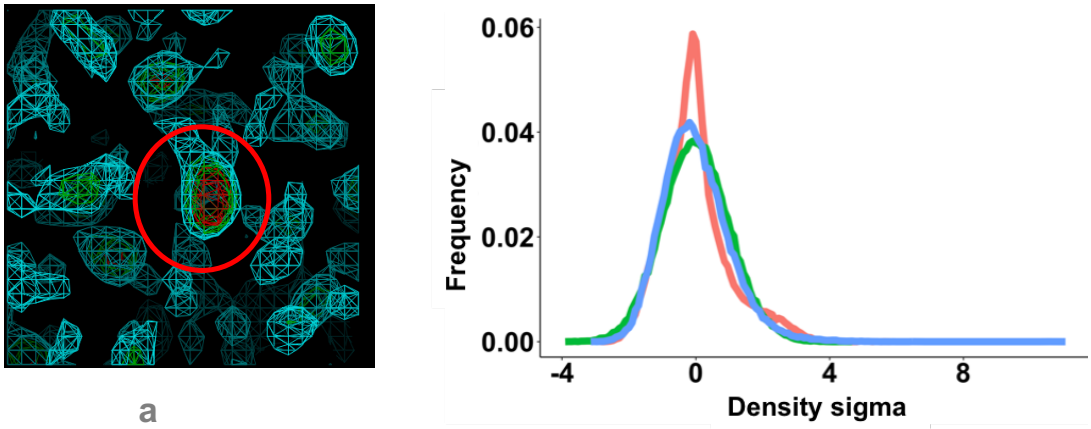


Figure 69 The test case II using GA design 2a. (a) The overlap of three maps is shown: map 1 at 1.5 sigma above the mean (in cyan), map 2 at 2.0 sigma (in green), map 3 at 5.0 sigma (in red). The uranium-like solution with a peak height of 11 height/rms in the map computed with phases generated by GA at generation 80 is highlighted in red circle (b) The histograms of density maps showing the shift of map generated with GA (blue) towards the map with correct phases (red). The histogram of the *first parent* is shown in green.

Skewness at 1 Å

As skewness was found to ineffective as a fitness function at a resolution of 2.5 Å and when all reflections were permuted, its effectiveness was tested at a resolution of 1 Å in test case II using GA design 3a for the phase variability of 1°. At this resolution, the skewness and MCC was higher than the *first parent* and was steadily increasing over 150 generations (Figure 70a and 70b). The phase error was considerably lower than the *first parent* after 100 generations (Figure 70c). This proved that the skewness can be employed as a fitness function at a higher resolution of 1 Å and its applicability drops with the decrease in resolution. However, further investigation needed to be performed to observe its behaviour in other lower resolution ranges and in other successful designs.

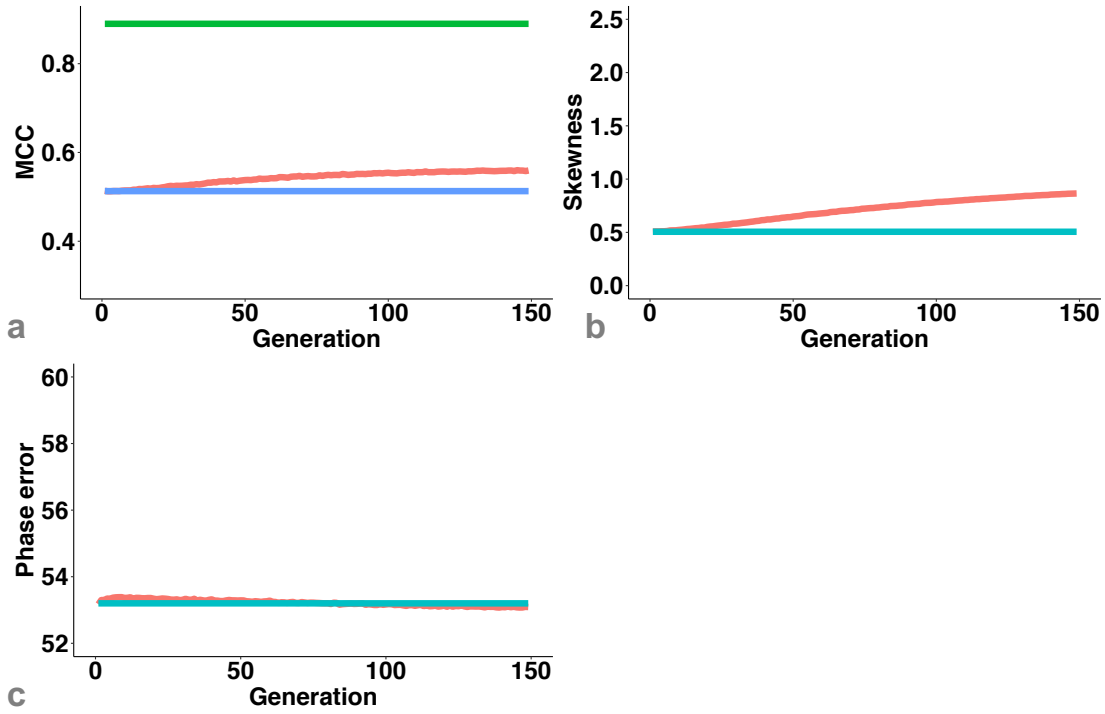


Figure 70 The performance of skewness at a resolution of 1 Å. The growth of MCC (Figure 70a, MCC in red, *first parent* in blue, *final point* in green), skewness (Figure 70b, skewness in red, *first parent* in cyan), and phase error (Figure 70c, phase error in red, *first parent* in cyan) in test case II using GA design 3a for the phase variability of 1°.

Map Connectivity

In the current study we evaluated the use of skewness and kurtosis both individually and in combination. Kurtosis found to be less effective than skewness and skewness found to less effective at lower resolutions. Therefore, at the resolution of 2.5 Å, the skewness was supplemented with a 3-dimensional parameter: map connectivity as a fitness function.

The rationale behind the use of the map connectivity information in fitness function was based on the fact that the better the map, the better the connectivity of the map skeleton with less free points and more connected fragments as shown in the Table 10.

Table 10 Table showing number of map points and fragments in the *first parent* (bad map) and the *final point* (good map) of test case II at various density levels.

Map type / Density level	**Good map / 0.8σ**	**Bad map / 0.8 σ**	**Good map / 1.0 σ**	**Bad map / 1.0 σ**	**Good map / 1.5 σ**	**Bad map / 1.5 σ**	**Good map / 2.0 σ**	**Bad map / 2.0 σ**
Points above density level	52733	59722	43072	45190	26695	19648	15988	7280
Points in final skeleton	7001	8999	6181	7550	4445	4640	3798	2505
Total fragments	476	164	623	384	411	909	285	791
Longest fragment	2258	8327	1132	3371	346	68	213	13

To develop map connectivity formula, a set of almost random electron density maps were generated with a phase error between 85° to 90° for test case II. A Greer's skeletonising method (Greer, 1974) was applied using Mapread module in ARP/wARP. From 1000 generated skeletons, the logarithm of the occurrence of each fragment length showed a linear dependence on its fragment length (Figure 71).

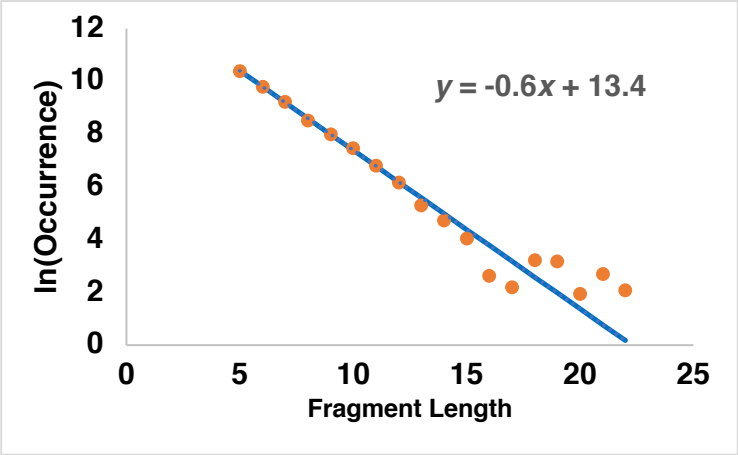


Figure 71 The dependence of occurrence of fragment length on fragment length.

From this, the number of observations of fragment length, l found at random can be described as

$$\ln(n_{frag}) = al + b \tag{18}$$

where a is the slope and b is the intercept. This can be written as

$$n_{frag} = ce^{al} \tag{19}$$

where a = -0.61 and $c = 1060$ (the fraction of map points represented as a skeleton by Mapread). The expected number of fragments of length l to appear at random (expected noise) follows a Poisson distribution:

$$p(q;\lambda) = \frac{\lambda^q e^{-\lambda}}{q!} \tag{20}$$

where, $\lambda = n_{frag}$ and $q = 1$. Given the noisy data, the probability of observing a single fragment of fragment length l with $q = 0$ is considered as a signal event.

The fraction of the signal can then be computed as follows:

$$\sum_{l=0}^{k} \frac{mp(0;\lambda)}{mp(0;\lambda) + mp(1;\lambda)} \tag{21}$$

where m is the total obervations of fragment length l, and k is the total number of different fragment lengths observed. Using this function, the aim is to maximise the number of connected atoms. To make it human-interpretable, it was encoded in terms of probability.

We defined this fraction as a connectivity function. To observe the dependence of this function on the phase error, a map with correct phases was rounded and noise added by introducing small phase error increments up to the total phase error of 90°. With the increase in the phase error the map connectivity value decreased along with the decrease in MCC (Figure 72).

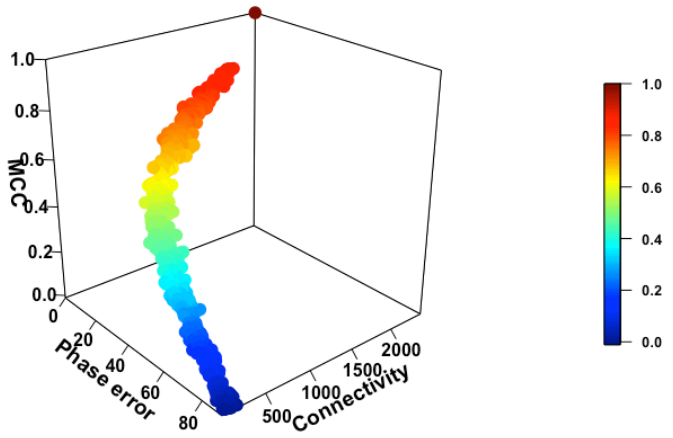


Figure 72 The correlation of connectivity to the phase error and MCC in test case II.

The applicability of connectivity as a fitness function together with skewness was tested using the following equation with different weighting fractions w: 0.25, 0.50 and 1.

$$c_2 = wg_1 + (1 - w)g_2 \tag{22}$$

where g_1 is skewness and g_2 is map connectivity.

Combination of Skewness and Connectivity

The combination of skewness and connectivity was tested with $w = 0.5$ in test case II using GA design 2a and 2b with all phase variabilities. The use of skewness combined with connectivity with a $w = 0.5$ in test case II using GA design 2a showed less deterioration in MCC compared to skewness alone (Figure 73a and 68a). The improvement in the skewness was less compared to skewness alone (Figure 73b and

68c). The growth in the phase error was also found be less than skewness alone (Figure 11d and 68e). The connectivity showed the signs of curve flattening before convergence to the correct minimum, Figure 73c.

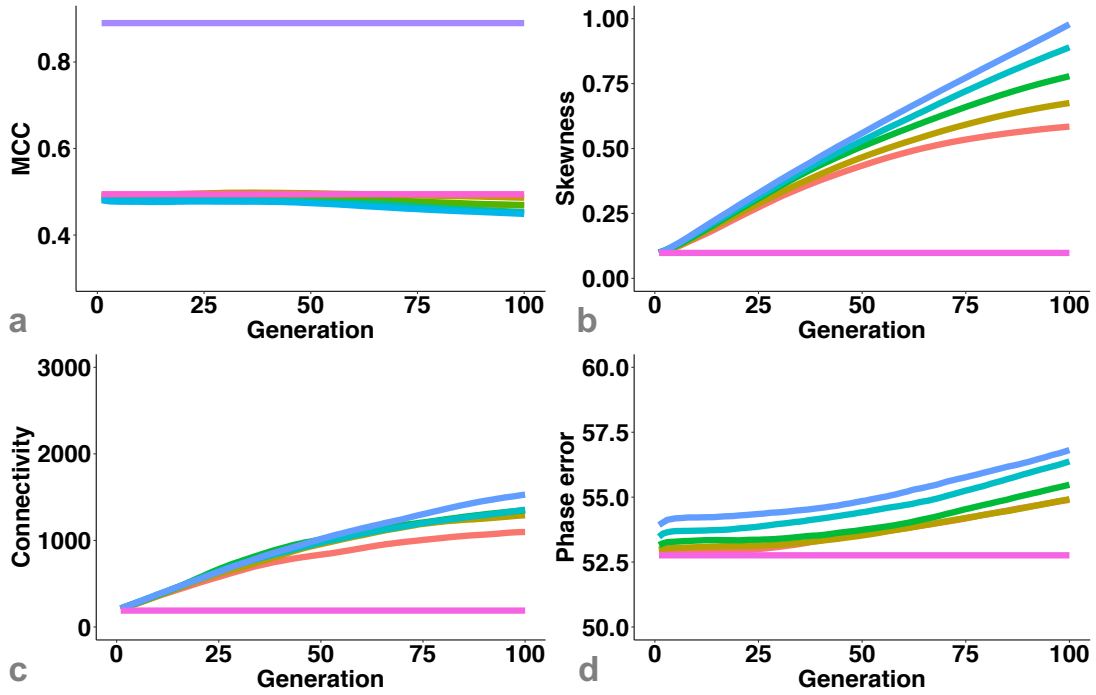
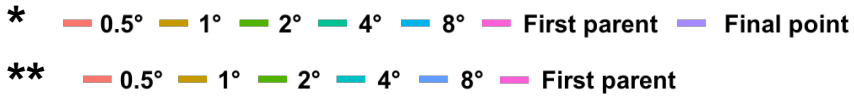


Figure 73 The growth of MCC (Figure 73a), skewness (Figure 73b), connectivity (Figure 73c) and phase error (Figure 73d) in the test case II using GA design 2a for all phase variabilities with a combination of skewness and connectivity as a fitness function. The weighting factor used was 0.50. * colour legend for Figure 73a ** colour legend for Figures 73b, 73c and 73d.



For a combination of skewness and connectivity in test case II using GA design 2b, the decrement in the MCC was much less compared to skewness alone (Figure 74a and 68b). It was also less than GA design 2a ran using a combination of skewness and connectivity in test case II with $w = 0.50$ (Figure 74a and 73a). The skewness improved linearly (Figure 74b) and the connectivity did not show any signs of the curve flattening until generation 200 (Figure 74c). The phase error was less than 55.5° which was much lesser than the phase error in GA design 2a with the same parameters (Figure 74d and 73d).

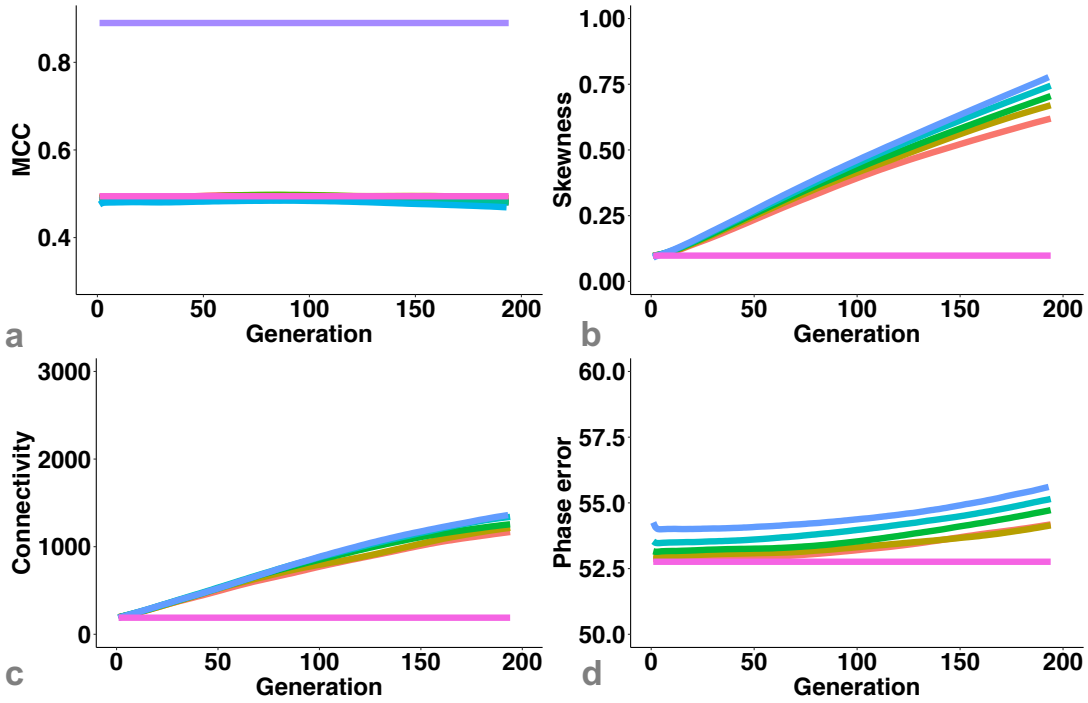
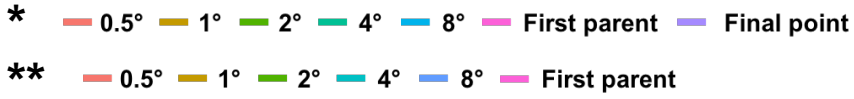


Figure 74 The growth of MCC (Figure 74a), skewness (Figure 74b), connectivity (Figure74c) and phase error (Figure 74d) in test case II using GA design 2b for all phase variabilities with a combination of skewness and connectivity as a fitness function. The weighting factor used was 0.50. * colour legend for Figure 74a ** colour legend for Figures 74b, 74c and 74d.



The combination of skewness and connectivity with $w = 0.25$ did not show comparatively better improvement in MCC and phase error than $w = 0.50$ (Figure 76). The growth of MCC, skewness, connectivity and phase error for all phase variabilities with skewness and connectivity as a fitness function using $w = 0.25$ is illustrated in Figure 75. The growth of all these parameters was worsened by a small amount compared to $w = 0.50$ (Figure 75 and Figure 74).

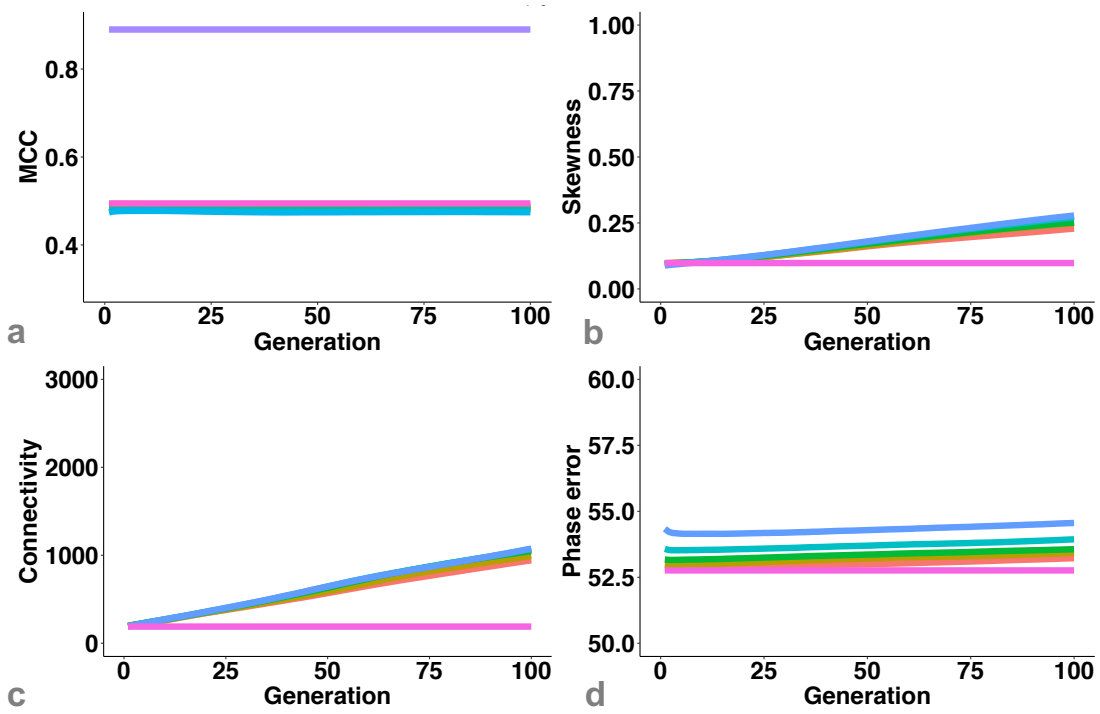


Figure 75 The growth of MCC (Figure 75a), skewness (Figure 75b), connectivity (Figure 75c) and phase error (Figure 75d) in test case II using GA design 2b for all phase variabilities with a combination of skewness and connectivity as a fitness function. The weighting factor used was 0.25. * colour legend for Figure 75a ** colour legend for Figures 75b, 75c and 75d.

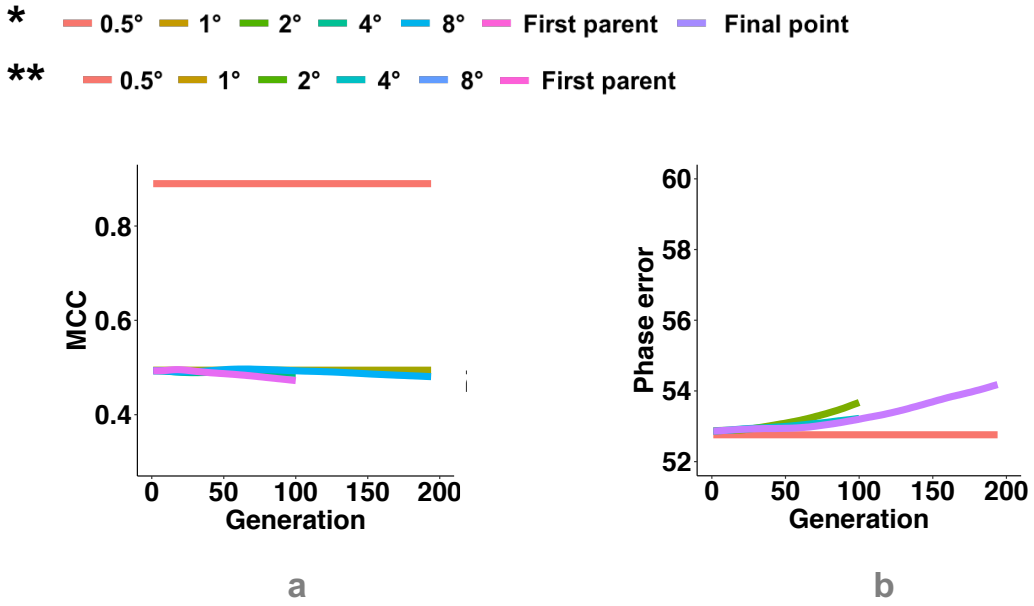


Figure 76 The comparative performance of skewness alone, skewness together with connectivity with $w = 0.50$ and skewness together with connectivity with $w = 0.25$.

Figure 76a shows the growth of MCC and Figure 76b showing the growth of phase error for these three parameters. * colour legend for Figure 76a ** colour legend for Figure 76b.

Skewness w25 Skewness w50 Skewness w100

* First parent Final point

** Skewness w25 Skewness w50 Skewness w100 First parent

The use of skewness and connectivity in combination with $w = 0.50$ in test case I and GA design 2b showed an improvement in the MCC. The MCC of the population was higher than the MCC of the *first parent* after 30 generations (Figure 77a). The skewness and connectivity also increased steadily (Figure 77b and 77c). The phase error was steadily decreasing and became nearly equivalent to the phase error of the *first parent* in 100 generations (Figure 77d). This fitness function showed positive results in this test case with experimental phase error.

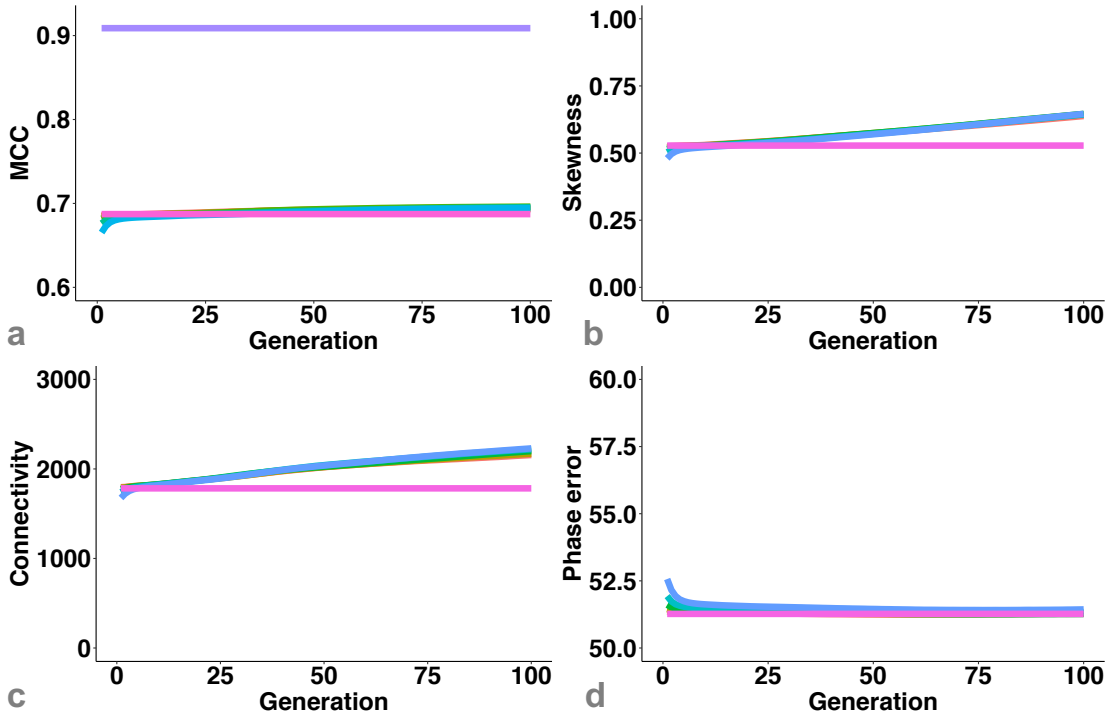


Figure 77 The growth of MCC (Figure 77a), skewness (Figure 77b), connectivity (Figure 77c) and phase error (Figure 77d) in test case I using GA design 2b for all phase variabilities with a combination of skewness and connectivity as a fitness

function. The weighting factor used was 0.50. * colour legend for Figure 77a ** colour legend for Figures 77b, 77c and 77d.

Concluding Remarks

Using skewness, the results obtained with GA design 1, GA design 2a and GA design 2b showed similar relative performance of GA designs to the results obtained by using MCC as a fitness function. Among these three designs, GA design 2b produced the best results.

Among various map characteristics tested in test case II, skewness alone found to better performing that kurtosis alone or a combination of skewness and kurtosis. However, the skewness found to ineffective measure as a fitness function at a lower resolution of 2.5 Å. At a higher resolution of 1 Å, skewness produced promising results in test case II using GA design 3 with a phase variability of 1°. Its applicability should be tested further in other resolution ranges.

The combination of skewness and connectivity found to be a better fitness function compared to the use of skewness alone. In particular, the skewness and connectivity combined with a weight factor, $w = 0.50$ proved more effective that other weighting factors. This parameter combination showed comparatively less negative improvement in MCC than using skewness alone as a fitness function in a test case II using GA design 2a and 2b. In the test case I, this combination showed positive results. The combination further needs to be tested in the most successful design, GA design 3.

Chapter 6

Conclusion and Outlook

This thesis work focused on two different aspects of development of GAs: optimisation of parameters of GAs for the phase improvement and identification of fitness function that best represents the quality of the electron density map. The major focus was to optimise a GA for phase improvement. To achieve this, different GA parameters were evaluated and the best performing parameters were identified.

In crossover, the uniform crossover showed better performance than one-point crossover (Figure 43). The variant of uniform crossover that produces two children per a pair of parents and discards the second child proved to enrich the diversity of the population more than the other variant that keeps both children (Figure 43). Further, the crossing of parent with only one selected partner many times resulted in early loss in diversity. Crossing a parent with different partners proved to be more advantageous in enhancing the diversity (Figure 66).

The two selection methods SUS and tournament were tested. The tournament showed better performance than SUS (Figure 67). In tournament selection, the size of the tournament was found to be an extremely important parameter for controlling selection intensity and the diversity of the population. The tournament size of 9 was shown to produce a population that has little or no representation from the far-worst performing individuals. This resulted in early loss of diversity leading to *premature convergence*. Whereas a tournament size of 2 retained a good number of individuals from different performance ranges. This allowed population to grow towards the global optimum without convergence (Figure 44).

Mutations with a rate higher than 2 were proved to be detrimental for the improvement of the method (Figure C.2). Three different types of mutations were studied: static, dynamic and directed using a mutation rate lower than 0.5. Both static and dynamic mutations proved to be beneficial when introduced in the late developmental stages (Figure 45). Next-generation mutations called “directed mutations” that target genes with little or no change in their gene value over few generations was developed. Initial results presented promising changes in gene value combinations (Figures 47 and 50). This work should be pursued in the future.

The diversity in the initial population was also proved to influence the performance of the population. In a better performing GA designs, phase variability higher than 4° produced better results compared to other phase variabilities (Figure 51).

In the second part of this thesis work, map characteristics (skewness, kurtosis) and map connectivity were studied for their use as a fitness function. Skewness alone performed better than kurtosis and a combination of skewness and kurtosis (Figure 64). Skewness was found be a better map quality indication at resolution of 1 Å (Figure 70) and proved to be ineffective at a resolution of 2.5 Å or lower (Figure 68). Further studies need to be performed at different resolution ranges to assess its usefulness as a fitness function. The use of skewness together with connectivity showed comparatively better results that the use of skewness alone as fitness function at a resolution of 2.5 Å (Figure 76).

GAs in this work were developed as a proof of concept, and is far from ideal implementation. The computations for each generation take 90 to 120 minutes. Nearly 30% of this time is taken by file conversions (ASCII to binary). This can be reduced by working directly with mtz files. Further reduction in the computational time is feasible by introducing a few modifications in the program for file handling and processing.

GAs optimised in this work were tested in test cases with a phase error of 53°. Further improvement and optimisation of the algorithm can be useful in extending its application for *ab inito* phasing. The study of map quality indicators identified for phase improvement in this work can be applied to other stages of the structure solution process such as model building and refinement that are dependent of the quality of electron density maps.

The phase improvement problem is generally a local optimisation problem. GAs are ideally used for global optimisation problems. An incorrect selection of parameters based on its global optimisation property might lead a big failure. Therefore, this work presents a successfully customization of parameters of GA for phase optimisation which is a local optimisation problem.

Bibliography

Allerhand, A., Doddrell, D., Glushko, V., Cochran, D. W., Wenkert, E., Lawson, P. J. & Gurd, F. R. (1971). *J. Am. Chem. Soc.* **93**, 544–546.

Alvarez, G. (2002). Can we make genetic algorithms work in high-dimensionality problems. Stanford.

Astbury, W. T. & Street, A. (1932). *Philos. Trans. R. Soc. London A Math. Phys. Eng. Sci.* **230**, 75–101.

Bäck, T., Hoffmeister, F. & Schwefel, H.-P. (1991). *Proc. Fourth Int. Conf. Genet. Algorithms*. **9**, 8.

Baker, J. E. (1985). *Proceedings of the First International Conference on Genetic Algorithms*, Vol. pp. 101–111. Erlbaum.

Baker, J. E. (1987). *Proceedings of the Second International Conference on Genetic Algorithms and Their Application*, Vol. pp. 14–21. Erlbaum.

Banerjee, S., Bartesaghi, A., Merk, A., Rao, P., Bulfer, S. L., Yan, Y., Green, N., Mroczkowski, B., Neitz, R. J., Wipf, P., Falconieri, V., Deshaies, R. J., Milne, J. L. S., Huryn, D., Arkin, M. & Subramaniam, S. (2016). *Science.* **351**, 871–875.

Bartesaghi, A., Merk, A., Banerjee, S., Matthies, D., Wu, X., Milne, J. L. S. & Subramaniam, S. (2015). *Science.* **348**, 1147–1151.

Baum, J., Dobson, C. M., Evans, P. A. & Hanley, C. (1989). *Biochemistry*. **28**, 7–13.

Bernal, J. D. & Crowfoot, D. (1934). *Nature*. **133**, 794–795.

Bethke, A. D. (1980). Genetic Algorithms as Function Optimizers. Ph.D. thesis, University of Michigan.

Blickle, T. & Thiele, L. (1995). *Proc. Sixth Int. Conf. Genet. Algorithms*. 9–16.

Blickle, T. & Thiele, L. (1996). *Evol. Comput.* **4**, 361–394.

Bloch, F. (1946). *Phys. Rev.* **70**, 460–474.

Bracegirdle, B. (1989). *Trends Biochem. Sci.* **14**, 464–468.

Bragg, W. L. (1913). *Proc. R. Soc. London A*. **89**, 248–276.

Bragg, W. L. (1929). *Proc. R. Soc. London A*. **123**, 537–559.

Bramlette, M. F. (1991). *Proceedings of the Fourth International Conference on Genetic Algorithms*, Vol. pp. 100–107. Morgan Kaufman.

Burley, S. K., Berman, H. M., Bhikadiya, C., Bi, C., Chen, L., Costanzo, L. Di, Christie, C., Duarte, J. M., Dutta, S., Feng, Z., Ghosh, S., Goodsell, D. S., Green, R. K., Guranovic, V., Guzenko, D., Hudson, B. P., Liang, Y., Lowe, R.,

Peisach, E., Periskova, I., Randle, C., Rose, A., Sekharan, M., Shao, C., Tao, Y. P., Valasatava, Y., Voigt, M., Westbrook, J., Young, J., Zardecki, C., Zhuravleva, M., Kurisu, G., Nakamura, H., Kengaku, Y., Cho, H., Sato, J., Kim, J. Y., Ikegawa, Y., Nakagawa, A., Yamashita, R., Kudou, T., Bekker, G. J., Suzuki, H., Iwata, T., Yokochi, M., Kobayashi, N., Fujiwara, T., Velankar, S., Kleywegt, G. J., Anyango, S., Armstrong, D. R., Berrisford, J. M., Conroy, M. J., Dana, J. M., Deshpande, M., Gane, P., Gáborová, R., Gupta, D., Gutmanas, A., Koča, J., Mak, L., Mir, S., Mukhopadhyay, A., Nadzirin, N., Nair, S., Patwardhan, A., Paysan-Lafosse, T., Pravda, L., Salih, O., Sehnal, D., Varadi, M., Vařeková, R., Markley, J. L., Hoch, J. C., Romero, P. R., Baskaran, K., Maziuk, D., Ulrich, E. L., Wedell, J. R., Yao, H., Livny, M. & Ioannidis, Y. E. (2019). *Nucleic Acids Res.* **47**, D520–D528.

Callaway, E. (2015). *Nature*. **525**, 172–174.

Campbell, I. D. (2002). *Nat. Rev. Mol. Cell Biol.* **3**, 377–381.

Campos, I. T. N., Souza, T. A. C. B., Torquato, R. J. S., De Marco, R., Tanaka-Azevedo, A. M., Tanaka, A. S. & Barbosa, J. A. R. G. (2012). *Acta Crystallogr. Sect. D*. **68**, 695–702.

Chang, G. & Lewis, M. (1994). *Acta Crystallogr. Sect. D Biol. Crystallogr.* **50**, 667–674.

Chang, G. & Lewis, M. (1997). *Acta Crystallogr. Sect. D*. **53**, 279–289.

Cochran, W. (1955). *Acta Crystallogr.* **5**, 65–67.

Cowtan, K. (2010). *Acta Crystallogr. Sect. D Biol. Crystallogr.* **66**, 470–478.

Crow, J. F. & Kimura, M. (1970). An introduction to population genetics theory. Harper & Row.

Davis, L. (1989). *Proceedings of Third International Conference on Genetic Algorithms, 1989*, Vol. pp. 61–69.

Davisson, C. & Germer, L. H. (1927). *Phys. Rev.* **30**, 705–740.

Deb, K. & Goldberg, D. E. (1993). *Foundations of Genetic Algorithms*, Vol. *2*, pp. 93–108. Elsevier.

Delbrueck, M. (1966). *A Physicist Looks at Biology*, Vol. pp. 9–22. New York, NY: Cold Spring Habor Laboratory of Quantitative Biology.

Drenth, J. (1999). Principles of Protein X-Ray Crystallography: Second Edition. New York, NY: Springer.

Emsley, P. & Cowtan, K. (2004). *Acta Crystallogr. Sect. D*. **60**, 2126–2132.

Eshelman, L., Caruana, R. & Schaffer, J. (1989). *Proceedings of the Third International Conference on Genetic Algorithms*, Vol. pp. 10–19. Morgan Kaufman.

Evrard, G., Mareuil, F. & Bontems, F. (2011). *J. Appl. Crystallogr.* **44**, 1264–1271.

Ewald, P. P. (1962). *Fifty Years of X-Ray Diffraction*, Vol. pp. 31–56. International Union of Crystallography.

Ten Eyck, L. F. (1973). *Acta Crystallogr. Sect. A*. **29**, 183–191.

Fersht, A. R. (2008). *Nat. Rev. Mol. Cell Biol.* **9**, 650–654.

Foos, N. & Nanao, M. H. (2019). *Acta Crystallogr. Sect. D Struct. Biol.* **75**, 200–210.

Forrest, S. & Mitchell, M. (2016). *Commun. ACM*. **59**, 58–63.

Goldberg, D. E. (1989*a*). *Complex Syst.* **3**, 153–171.

Goldberg, D. E. (1989*b*). *Complex Syst.* **3**, 129–152.

Goldberg, D. E. (1989*c*). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley.

Goldberg, D. E. (1989*d*). *Proceedings of the Third International Conference on Genetic Algorithms*, Vol. pp. 70–79. Morgan Kaufmann.

Goldberg, D. E. & Deb, K. (1991). *Foundations of Genetic Algorithms*, Vol. *1*, pp. 69–93. Elsevier.

Grefenstette, J. (1986). *IEEE Trans. Syst. Man. Cybern.* **16**, 122–128.

Grefenstette, J. J. (1993). *Foundations of Genetic Algorithms*, Vol. *2*, pp. 75–91. Elsevier.

Grefenstette, J. J. & Baker, J. E. (1989). *Proceedings of Third International Conference on Genetic Algorithm*, Vol. pp. 20–27. Morgan Kaufmann.

Gruene, T., Wennmacher, J. T. C., Zaubitzer, C., Holstein, J. J., Heidler, J., Fecteau-Lefebvre, A., De Carlo, S., Müller, E., Goldie, K. N., Regeni, I., Li, T., Santiso-Quinones, G., Steinfeld, G., Handschin, S., van Genderen, E., van Bokhoven, J. A., Clever, G. H. & Pantelic, R. (2018). *Angew. Chem. Int. Ed. Engl.* **57**, 16313–16317.

Hanson, A. J., Cheung, E. Y., Habershon, S. & Harris, K. D. M. (2005). *Acta Crystallogr. Sect. A*. **61**, c162.

Harker, D. & Kasper, J. S. (1948). *Acta Crystallogr.* **1**, 70–75.

Harris, K. D. M., Johnston, R. L. & Kariuki, B. M. (1998). *Acta Crystallogr. Sect. A Found. Crystallogr.* **54**, 632–645.

Hauptman, H. A. (1991). *Reports Prog. Phys.* **54**, 1427–1454.

Heims, S. J. (1980). John von Neumann and Norbert Weiner: From Mathematics to The Technologies of Life and Death. MIT Press.

Hillis, W. D. (1990). *Phys. D Nonlinear Phenom.* **42**, 228–234.

Hoinkes, H. (1967). *J. Glaciol.* **6**, 757.

Holland, J. H. (1975). Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press.

Holliday, R. (2006). *J. Genet.* **85**, 93–97.

Hooke, R. (1665). Micrographia: or Some physiological descriptions of minute bodies made by magnifying glasses. With Observations and Inquiries Thereupon. London: The Royal Society.

Immirzi, A., Erra, L. & Tedesco, C. (2008). *J. Appl. Crsytallography*. **41**, 784–790.

De Jong, K. A. (1975). Analysis of the Behavior of a Class of Genetic Adaptive Systems. Ph.D. Thesis, University of Michigan.

Jong, K. A. De (1993). *Foundations of Genetic Algorithms*, Vol. *2*, pp. 5–17. Elsevier.

Jong, K. A. De & Sarma, J. (1993). *Foundations of Genetic Algorithms*, Vol. *2*, pp. 19–28. Elsevier.

Jorda, J. & Michael, R. (2014). *Acta Crystallogr. - Sect. D Biol. Crystallogr.* **70**, 1538–1548.

Jorda, J., Sawaya, M. R. & Yeates, T. O. (2016). *Acta Crystallogr. Sect. D Struct. Biol.* **72**, 446–453.

Kariuki, B. M., Serrano-González, H., Johnston, R. L. & Harris, K. D. M. (1997). *Chem. Phys. Lett.* **280**, 189–195.

Karle, J. & Hauptman, H. (1950). *Acta Crystallogr.* **3**, 181–187.

Kay, L. E. (1996). *J. Hist. Biol.* **29**, 477–479.

Kendrew, J. C., Bodo, G., Dintzis, H. M., Parrish, R. G., Wyckoff, H. & Phillips, D. C. (1958). *Nature*. **181**, 662–666.

Kendrew, J. C., Dickerson, R. E., Strandberg, B. E., Hart, R. G., Davies, D. R., Phillips, D. C. & Shor, V. C. (1960). *Nature*. **185**, 422–427.

Kepler, J. (1966). The six-cornered snowflake; Oxford: Clarendon P.

Kirk, R. (2014). *Nature*. **511**, 13.

Kissinger, C. R., Gehlhaar, D. K. & Fogel, D. B. (1999). *Acta Crystallogr. Sect. D*. **55**, 484–491.

Kumar, A. (2013). *Int. J. Adv. Res. IT Eng.* **2**, 1–7.

Lamzin, V. S. (2013). The correlation of skewness and kurtosis on the phase error .Personal communication.

Lamzin, V. S. & Wilson, K. S. (1993). *Acta Crystallogr D Biol Crystallogr*. **49**, 129–147.

Lane, N. (2015). *Philos. Trans. R. Soc. B Biol. Sci.* **370**, 1–10.

Langer, G. G., Hazledine, S., Wiegels, T., Carolan, C. & Lamzin, V. S. (2013). *Acta Crystallogr. Sect. D*. **69**, 635–641.

Leeuwenhoek, A. van & Hoole, S. (1800). The Select Works of Antony van Leeuwenhoek, Containing His Microscopical Discoveries in Many of the Works of Nature. G. Sidney.

Levdikov, V. M., Barynin, V. V, Grebenko, A. I., Melik-adamyan, W. R., Lamzin, V. S. & Wilson, K. S. (1998). *Structure*. **6**, 363–376.

Liebschner, D., Afonine, P. V, Baker, M. L., Bunkóczi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L.-W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V, Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J. & Adams, P. D. (2019). *Acta Crystallogr. Sect. D*. **75**, 861–877.

Lunin, V. Y. (1993). *Acta Crystallogr. Sect. D*. **49**, 90–99.

Lunin, V. Y. & Woolfson, M. M. (1993). *Acta Crystallogr. D. Biol. Crystallogr.* **49**, 530–533.

Mason, A. J. (1993). Crossover non-linearity ratios and the genetic algorithm : escaping the blinkers of schema processing and intrinsic parallelism. Report no.535b, School of Engineering, University of Auckland.

Merk, A., Bartesaghi, A., Banerjee, S., Falconieri, V., Rao, P., Davis, M. I., Pragani, R., Boxer, M. B., Earl, L. A., Milne, J. L. S. & Subramaniam, S. (2016). *Cell*. **165**, 1698–1707.

Miller, S. T., Hogle, J. M. & Filman, D. J. (1996). *Acta Crystallogr. Sect. D Biol. Crystallogr.* **52**, 235–251.

Mitchell, M. (1998). An Introduction to Genetic Algorithms. MIT Press.

Mitchell, M., Forrest, S., Holand, J. H., Mexico., U. of N. & Science., D. of C. (1992). *Proc. First Eur. Conf. Artif. Life.* **1**, 245–254.

Mitchell, M., Holland, J. H. & Forrest, S. (1993). *International Conference on Neural Information Processing Systems 6*, Vol. pp. 51–58. Morgan Kaufmann.

Mitra, A. K. (2019). *Acta Crystallogr. Sect. F*. **75**, 3–11.

Morris, R. J. & Bricogne, G. (2003). *Acta Crystallogr. Sect. D*. **59**, 615–617.

Mühlenbein, H. & Schlierkamp-Voosen, D. (1993). *Evol. Comput.* **1**, 25–49.

Murshudov, G. N., Vagin, A. & Eleanor, D. (1997). *Acta Crystallogr.* **53**, 240–255.

Neumann, J. V (2017). *Syst. Res. Behav. Sci. A Sourceb.* **V**, 97–107.

Nishibori, E., Ogura, T., Aoyagi, S. & Sakata, M. (2008). 292–301.

Parsons, R. J., Forrest, S. & Burks, C. (1995). *Mach. Learn.* **21**, 11–33.

Patterson, A. L. (1934). *Phys. Rev.* **46**, 372–376.

Pauling, L. & Corey, R. B. (1951). *Proc. Natl. Acad. Sci. U. S. A.* **37**, 235–240.

Pauling, L. & Corey, R. B. (1953). *Proc. Natl. Acad. Sci. U. S. A.* **39**, 84–97.

Peck, C. C. & Dhawan, A. P. (1993). A Review and Critique of Genetic Algorithm Theories. Technical Report TR 153/6/93/ECE, Department of Electrical and Computer Engineering, College of Engineering University of Cincinnati.

Perutz, M. (1985). *Methods Enzymol.* **114**, 3–18.

Perutz, M. F., Rossmann, M. G., Cullis, A. N. N. F., Muirhead, H., Will, G. & North, A. C. T. (1960). *Nature*. **185**, 416–422.

Pervushin, K., Riek, R., Wider, G. & Wüthrich, K. (1997). *Proc. Natl. Acad. Sci.* **94**, 12366–12371.

Podjarny, A. D. & Yonath, A. (1977). *Acta Crystallogr. Sect. A Found. Crystallogr.* **33**, 655–661.

Purcell, E. M., Torrey, H. C. & Pound, R. V (1946). *Phys. Rev.* **69**, 37–38.

Read, R. J. & Schierbeek, A. J. (1988). *J. Appl. Crystallogr.* **21**, 490–495.

Röntgen, W. C. (1896). *Science.* **3**, 227–231.

De Rosier, D. J. & Klug, A. (1968). *Nature*. **217**, 130–134.

Rossmann, M. G. & Blow, D. M. (1962). *Acta Crystallogr.* **15**, 24–31.

Rossmann, M. G. & Blow, D. M. (1963). *Acta Crystallogr.* **16**, 39–45.

Sastry, K. & Goldberg, D. E. (2002). Analysis of Mixing in Genetic Algorithms : A Survey.

Sayre, D. (1952). *Acta Crystallogr.* **5**, 60–65.

Schaffer, J. D., Caruana, R. A., Eshelman, L. J. & Das, R. (1989). *Proceedings of the Third International Conference on Genetic Algorithms*, Vol. pp. 51–60. Morgan Kaufmann.

Schaffer, J. D., Eshelman, L. J. & Offutt, D. (1991). *Foundations of Genetic Algorithms*, Vol. *1*, pp. 102–112. Elsevier.

Schneider, T. R. (2002). *Acta Crystallogr. Sect. D Biol. Crystallogr.* **58**, 195–208.

Schraudolph, N. N. & Belew, R. K. (1992). *Mach. Learn.* **9**, 9–21.

Schwarzenbach, D. (2012). *Acta Crystallogr. Sect. A*. **68**, 57–67.

Senaratna, N. I. (2005). Genetic Algorithms : The Crossover-Mutation Debate. A literature survey (CSS3137-B), Degree of Bachelor of Computer Science, University of Colombo.

Sevick, J., Lamzin, V. S., Dauter, Z. & Keith, W. S. (2002). *Acta Crystallogr. - Sect. D Biol. Crystallogr.* **58**, 1307–1313.

Sharon, M. (2010). *J. Am. Soc. Mass Spectrom.* **21**, 487–500.

Shi, D., Nannenga, B. L., Iadanza, M. G. & Gonen, T. (2013). *Elife*. **2**, e01345.

Spears, W. M. (1993). *Foundations of Genetic Algorithms*, Vol. *2*, edited by L.D.B.T.-F. of G.A. WHITLEY, pp. 221–237. Elsevier.

Spears, W. M. & Jong, K. A. De (1991). *Foundations of Genetic Algorithms*, Vol. *1*, pp. 301–315. Elsevier.

Strandberg, B., Dickerson, R. E. & Rossmann, M. G. (2009). *J. Mol. Biol.* **392**, 2–32.

Syswerda, G. (1991). *Foundations of Genetic Algorithms*, Vol. *1*, pp. 94–101. Elsevier.

Taylor, G. (2003). *Acta Crystallogr. D. Biol. Crystallogr.* **59**, 1881–1890.

Thierens, D. & Goldberg, D. E. (1993). *Proceedings of the Fifth International Conference on Genetic Algorithms*, Vol. pp. 38–47. Morgan Kaufman.

Thomson, G. P. & Reid, A. (1927). *Nature*. **119**, 890.

Truong, N. X., Whittaker, E. & Denecke, M. A. (2017). *J. Appl. Crystallogr.* **50**, 1637–1645.

Turing, A. M. (1950). *Mind*. **49**, 433–460.

Uervirojnangkoorn, M., Hilgenfeld, R., Terwilliger, T. & Read, R. j (2013). *Acta Crystallogr. - Sect. D Biol. Crystallogr.* **69**, 2039–2049.

Wang, B.-C. (1985). *Diffraction Methods for Biological Macromolecules Part B*, Vol. *115*, pp. 90–112. Academic Press.

Warren, M. (2018). *Nature*. **563**, 16–17.

Watson, J. D. & Crick, F. H. C. (1953). *Nature*. **171**, 737–738.

White, D. (2014). *Comput. Res. Repos.*

Whitley, D. (1989). *Proceedings of the Third International Conference on Genetic Algorithms*, Vol. pp. 116–121. Morgan Kaufman.

Whitley, L. D. (1991). Vol. *1*, pp. 221–241. Elsevier.

Winn, M. D., Ballard, C. C., Cowtan, K. D., Dodson, E. J., Emsley, P., Evans, P. R., Keegan, R. M., Krissinel, E. B., Leslie, A. G. W., McCoy, A., McNicholas, S. J., Murshudov, G. N., Pannu, N. S., Potterton, E. A., Powell, H. R., Read, R. J., Vagin, A. & Wilson, K. S. (2011). *Acta Crystallogr. D. Biol. Crystallogr.* **67**, 235–242.

Wright, S. (1931). *Genetics*. **16**, 97–159.

Wüthrich, K. (1998). *Nat. Struct. Biol.* **5**, 492–495.

Wüthrich, K. & Wagner, G. (1975). *FEBS Lett.* **50**, 265–268.

Xi, S., Borgna, L. S. & Du, Y. (2015). *J. Synchrotron Radiat.* **22**, 661–665.

Xi, S., Borgna, L. S., Zheng, L., Du, Y. & Hu, T. (2017). *J. Synchrotron Radiat.* **24**, 367–373.

xtal.iqfr.csic.es (2020). Structural Resolution.

Yakimov, Y. I., Semenkin, E. S. & Yakimov, I. S. (2009). *Acta Crystallogr. Sect. A*. **65**, s320.

Yakimov, Y., Semenkin, E. & Yakimov, I. (2008). *Acta Crystallogr. Sect. A*. **64**, C226.

Zander, U., Cianci, M., Foos, N., Silva, C. S., Mazzei, L., Zubieta, C., Maria, D. & Nanao, M. H. (2016). *Acta Crystallogr. Sect. D Struct. Biol.* **72**, 1026–1035.

Zhang, K. Y. J. & Main, P. (1990). *Acta Crystallogr. Sect. A*. **46**, 41–46.

Zhou, Y. & Su, W. P. (2004). *Acta Crystallogr. Sect. A Found. Crystallogr.* **60**, 306–310.

Zitzler, E. & Thiele, L. (1999). *IEEE Trans. Evol. Comput.* **3**, 257–271.

Supplementary Studies

Study S.1: Case III: Infestin 4 (PDB ID:2ERW)

The structure of Infestin 4 (Campos *et al.*, 2012) was solved using molecular replacement and refined at 1.4 Å to a Rfactor of 0.19, Figure A.1. The crystals belong to space group $P2_12_12_1$ and there is one molecule per asymmetric unit. The X-ray dataset with molecular replacement was truncated to 2.0 Å resolution. The Wilson B factor was upweighted by 23 $Å^2$ accordingly. There are 4784 unique reflections in this data at the selected resolution. The phases from the model refined against the 2.5 Å data were subject to an additional uniformly distributed phase error of 50°, as described in section "Test cases" of the chapter 2. These phases of acentric reflections were then rounded to the nearest value of 45°/135°/225°/315°, introducing an additional but small phase error of about 3°. This reduced the MCC from 0.5583 to 0.5106. These rounded phases were used as the starting point for GA. The "final point" to be reached here is a map at 2.0 Å with phases rounded to 45°/135°/225°/315° having an MCC of 0.8912 to the final map (map at 1.4 Å without any artificially introduced phase error).

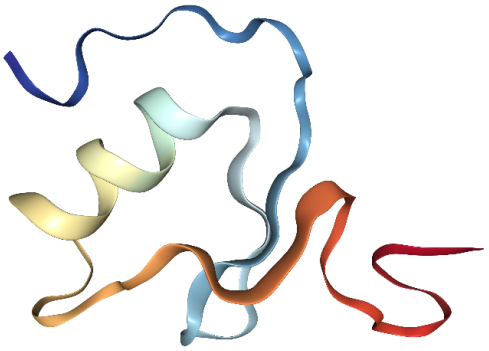


Figure A.1 Structure of Infestin 4 solved at resolution 1.4 Å (PDB ID: 2ERW). Figure taken from the PDB, (Burley *et al.*, 2019).

In the population generated by the test case III using GA design 1, no improvement in the MCC was observed. The MCC for phase variabilities: 0.5° and 1° was lower than the MCC of the first parent, Figure A.2a. Skewness improved for a few generations (approximately up to 30 generations) and reached steady-state in the subsequent

generations, Figure A.2b. Phase error in 80 generations was higher than the phase error of the first parent, Figure A.2c. This indicates that the success of the design is independent of the size of the test case.

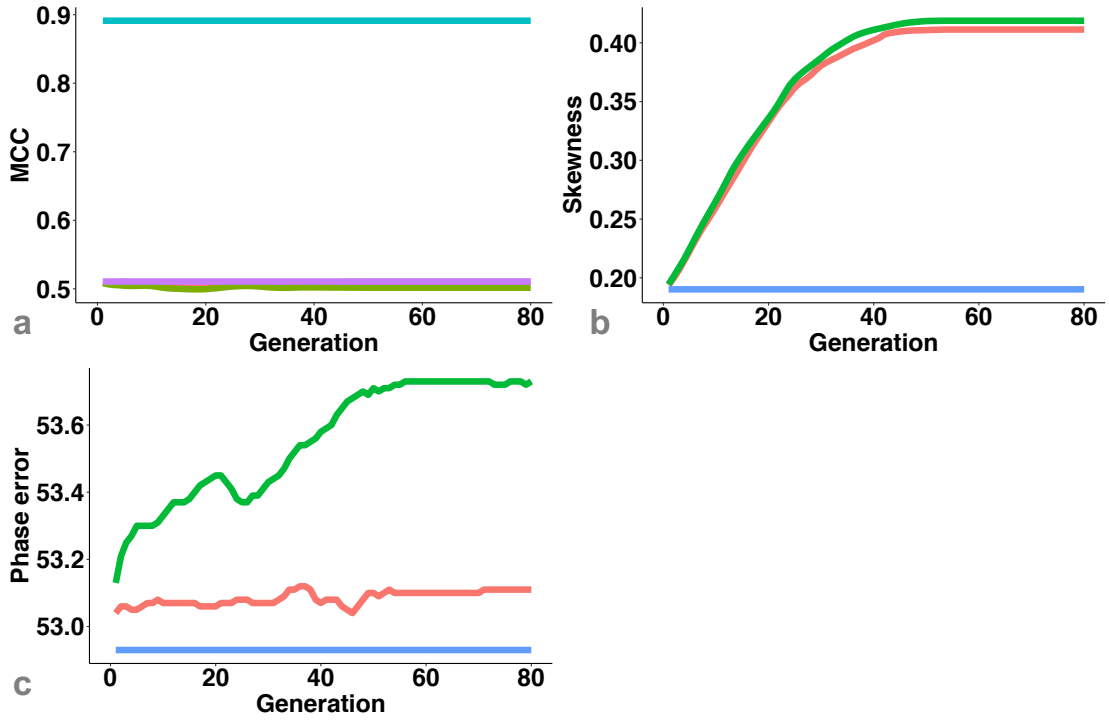


Figure A.2 Testing GA design in a smaller test case. The growth of MCC, skewness and phase error in test case III using GA design 1 with phase variabilities 0.5° and 1°. * colour legend for Figure A.2a ** colour legend of Figures A.2b and A.2c.

* 0.5° 1° First parent Final point

** 0.5° 1° First parent

Study S.2: GA with No Redundant Representations

In this study, the population was generated from the test case III using GA design 1 with phase variabilities: 0.3°, 0.5°, 1° and 2°. Each member's (or phase set's) fingerprint was generated using the md5 hashing method to identify duplicates. These duplicate or redundant members were then removed from the population. Only unique members were allowed to propagate. In this approach, the skewness reached steady state in less than 30 generations and no improvement in the MCC with little improvement in the phase error was observed. However, the skewness was higher (for phase variabilities 0.5° and 1°), Figure A.3b, compared to the same GA parameters and data set (population generated from test case III using GA design 1) ran without

removing redundant members, Figure A.2b. Overall, no improvement was observed by removing redundant members, so this removal was not implemented in successive GA designs.

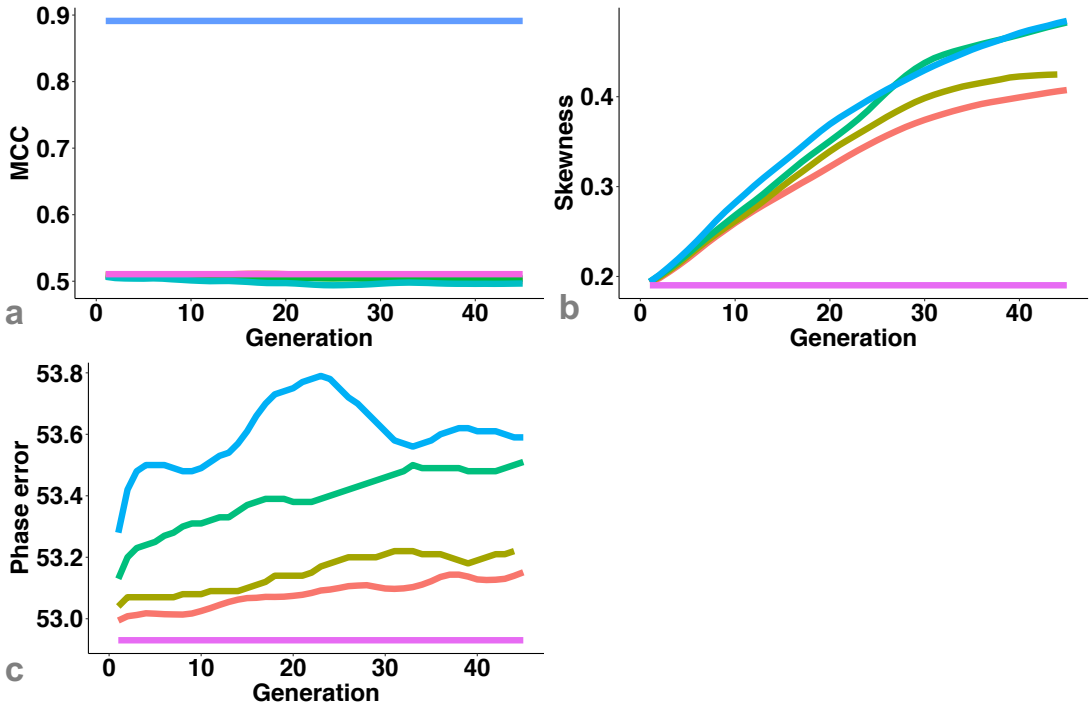


Figure A.3 GA without duplicate population. The growth of MCC, skewness and phase error in test case III using GA design 1 with phase varibilities 0.3°, 0.5°, 1° and 2°. * colour legend for Figure A.3a ** colour legend for Figures A.3b and A.3c.

* 0.3° 0.5° 1° 2° First parent Final point

** 0.3° 0.5° 1° 2° First parent

Study S.3: Significance of Precision Level (Number of Decimal Places Passed by Scoring Function)

In this study, the first parent generated from the test case I was taken. In this phase set, 50 genes or reflections were selected randomly and mutated to other possible phase values 45°/135°/225°/315°. For example, if a reflection's phase value was 45°, it was then mutated to 135°, 225° and 315° and thus three different variants of this reflection were generated. In other words, each variant differs only in single reflection's phase value compared to the first parent. By mutating 50 randomly selected

reflections, 150 variants were produced in total. The skewness and MCC were computed for these 150 variants with different precisions: 4, 5, 6, 7 and 8. The number of variants reflecting their changed phase value by giving different MCC and skewness values were identified for these precision levels, Figure A.4. A precision level higher than 8 for MCC and equal to 7 for skewness was required to record a change in the single reflection's phase value, Figure A.4. Therefore, a precision level of 8 was used for the computation of MCC and skewness in subsequent designs.

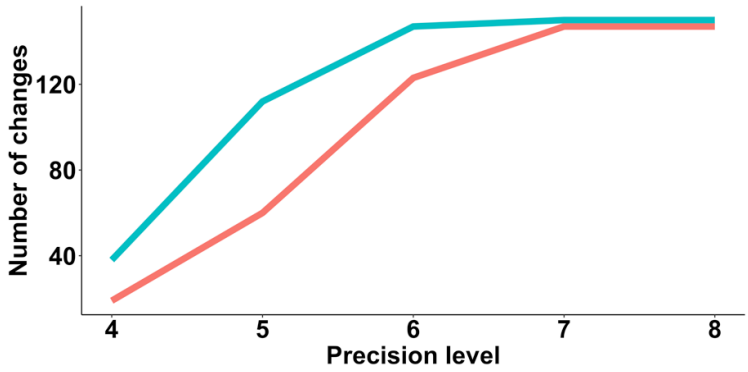


Figure A.4 Precision level required to record a change in single reflection's phase value for MCC (red) and skewness (cyan).

In the population generated from test case I using GA design 1 and with precision level of 4 and 8, the difference plots in the growth of MCC, skewness and phase error shows the importance of identifying the correct precision level, Figure A.5a. A better improvement in MCC (noticeably for 1°), Figure A.5a and A.5b, skewness (0.5° and 1°), Figure A.5c and A.5d, and phase error (0.5°, 2° and 4°), Figure A.5e and A.5f, was observed when precision level of 8 was used. However, the lack of reproducibility of this growth pattern in all phase variabilities is debatable. In this work, we stayed with the use of 8 decimal places based on the results of the variant analysis, Figure A.4.

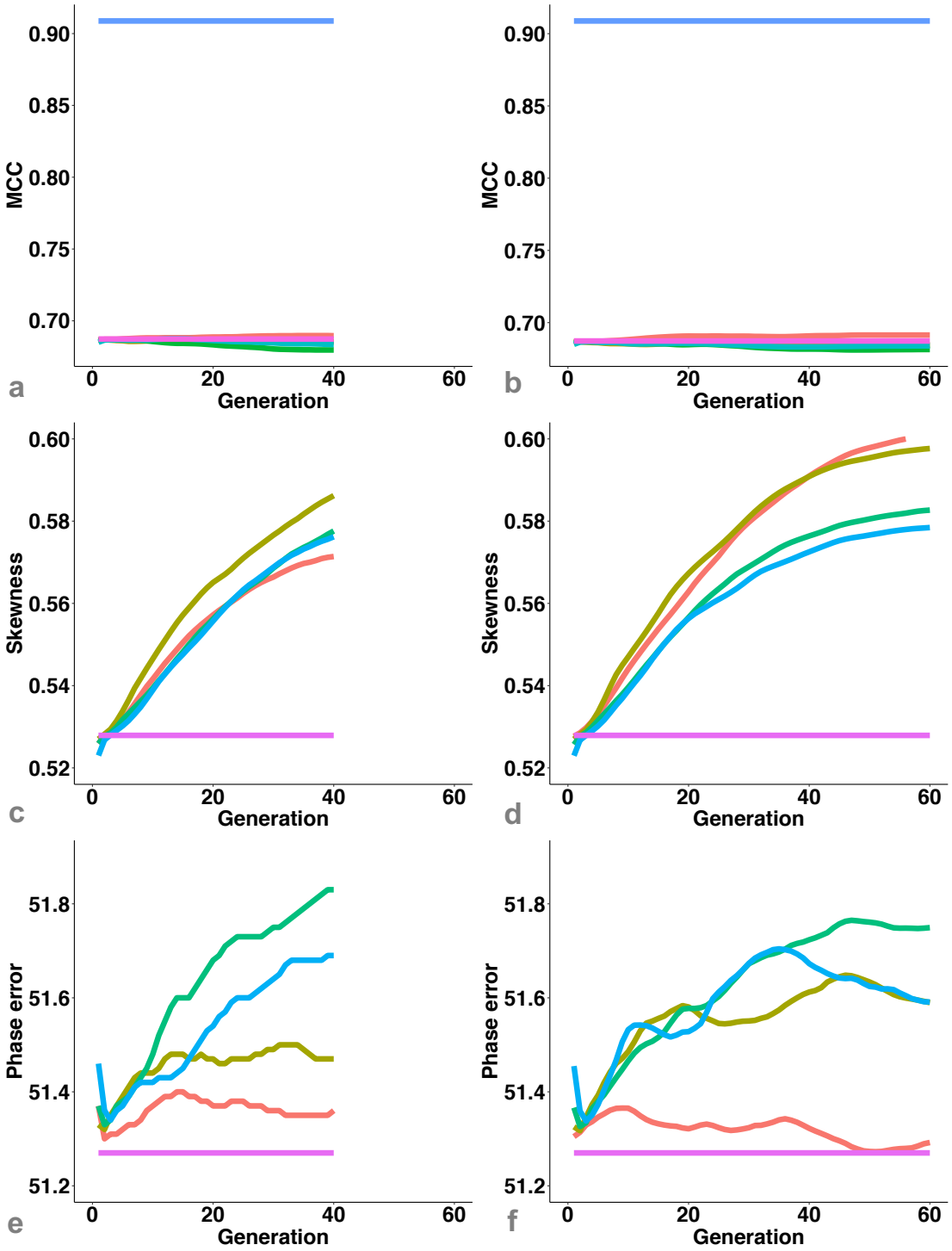


Figure A.5 The growth of MCC, skewness and phase error in test case I using GA design 1 with phase variabilities: 0.5°, 1°, 2° and 4°. Figure A.5a and A.5b shows the comparison of MCC with precision level of 4 and 8 respectively. Figure A.5c and A.5d shows the comparison of skewness with precision level of 4 and 8 respectively. Figure A.5e and A.5f shows the comparison of phase error with precision level of 4

and 8 respectively. * colour legend for Figures A.5a and A.5b **colour legend for Figures A.5c, A.5d, A.5e and A.5f.

* 0.5° 1° 2° 4° First parent Final point

** 0.5° 1° 2° 4° First parent

Study S.4: Dependence of MCC and Phase Error on Resolution

In this study, the module "ph_rms" of ARP/wARP (Lamzin & Wilson, 1993) was used to compute the average MCC and phase error of all reflections in test case II at different resolution ranges given in Table A.1 for the first parent, second parents, children and survivors. These were computed on the population generated using GA design 1 at generation 1. From these studies, no evidence of interpretable dependence on resolution was observed. Therefore, all reflections from all resolution ranges were used in GA designs, Figure A.6.

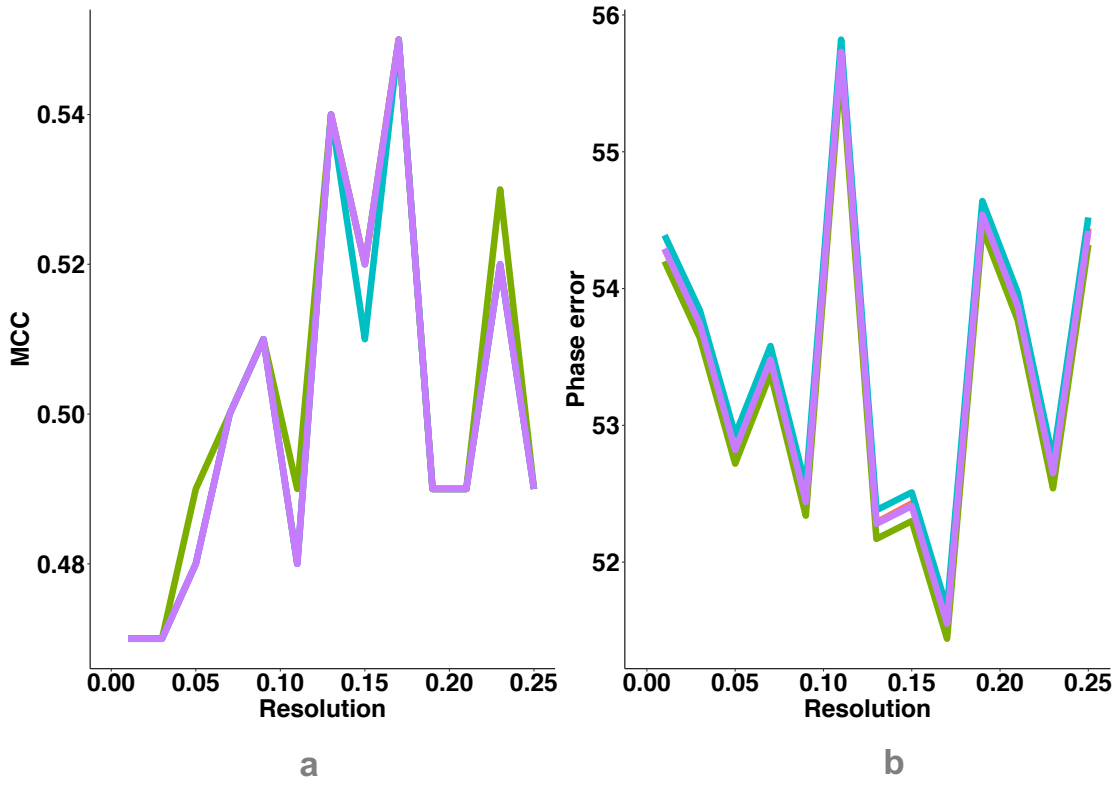


Figure A.6 Dependence of MCC (Figure A.6a) and phase error (Figure A.6b) on the resolution. X-axis showing resolution in $1/\text{Å}^2$ (Table A.1).

Children First parent Second parent Survivors

Table A.1 Resolution ranges used for the study S.4**.**

Å	1/Å^2
10.00 - 7.07	0.01
7.07 - 5.00	0.03
5.00 - 4.08	0.05
4.08 - 3.54	0.07
3.54 - 3.16	0.09
3.16 - 2.89	0.11
2.89 - 2.67	0.13
2.67 - 2.50	0.15
2.50 - 2.36	0.17
2.36 - 2.24	0.19
2.24 - 2.13	0.21
2.13 - 2.04	0.23
2.04 - 2.00	0.25

Study S.5: Dependence of MCC, Skewness, Phase Error on Structure Factor and Normalised Structure Factor

Uervirojnangkoorn *et al*., (2013) modified the phases for a subset of reflections with the strongest normalised structure factor amplitudes. Notably, among their three testsets, the highest phase improvement was reported when a fixed number, 100, of the reflections with strongest E-values were modified, regardless of the number of reflections in the X-ray data. In our design, the phases all reflections were subjected to change.

We carried out an additional investigation to study the effect of the percentage of strong reflections on skewness and MCC of the electron density map. We used the test case II at 2.0 Å, the *actual solution* (Figure 36d) and the one with 50° phase error without rounding phase values to 45°/135°/225°/315° (Figure 36a). Then we changed phases for a subset of reflections to their correct values.

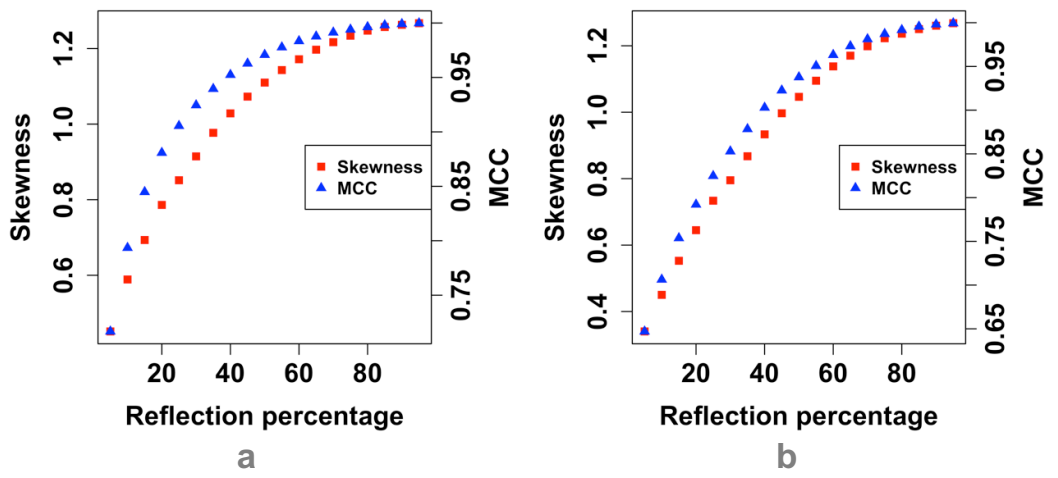


Figure A.7 The dependence of skewness (red square) and MCC (blue triangle) on structure factor amplitudes (Figure A.7a) and normalised structure factor amplitudes (Figure A.7b).

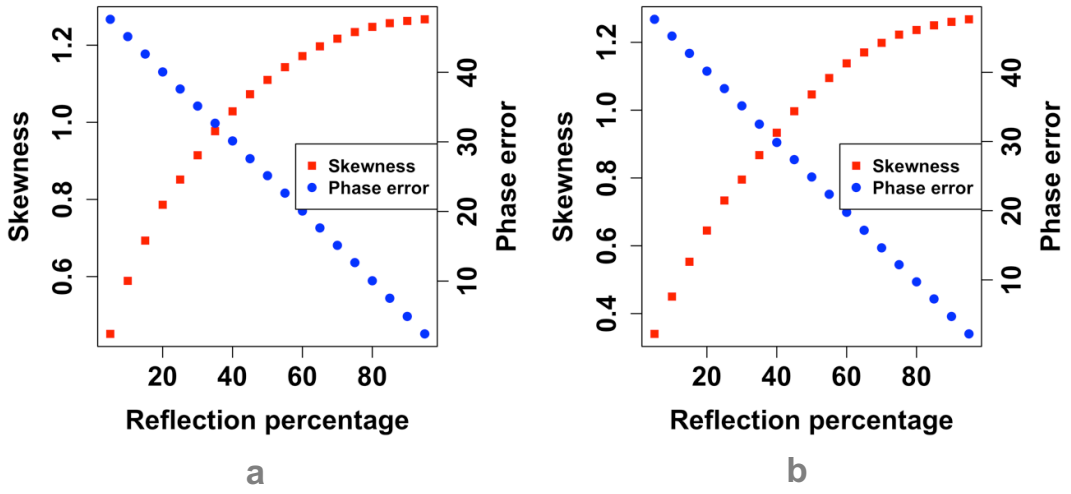


Figure A.8 The dependence of skewness (red square) and phase error (blue triangle) on structure factor amplitudes (Figure A.8a) and normalised structure factor amplitudes (Figure A.8b).

Figure A.7a presents the improvement of skewness and MCC when 0, 5, 10, etc, percent of strongest structure factors having an errored phase values were changed to the correct values. These results indicate that, to achieve an improvement of MCC

from 0.2 to 0.8, only 10% of phases of the strongest structure factors had to be changed to the correct values. Figure A.7b shows improvement of skewness and MCC when a subset (0%, 5%, 10% etc.) of strongest normalised structure factors having phase values with errors were changed to the correct values. In case of normalised structure factors, to achieve the same amount of improvement in MCC, 20 percent of phases of the strongest normalised structure factors had to be changed to the correct values. However, there was no significant difference observed in the dependence of phase error on structure factor amplitudes and normalised structure factor amplitudes, Figure A.8.

It is evident from the Lunin's formula on calculation of skewness in reciprocal space (Equation 14), that skewness is a function of the triple product of structure factor amplitudes, and not of *E*-values. Our simulation with strong reflections confirms this and indicates that a use of structure factor amplitudes should potentially be more efficient than the use of *E*-values. Hence, throughout this work structure factors amplitudes were used.

Appendix B

Supplementary Result Tables

Table B.1 The average number of acentric reflections with changed phases for different phase variabilities, if the number of reflections in a structure equals to 6000.

Phase Variability	**0.5°**	**1°**	**2°**	**4°**	**8°**
Average reflections with changed phases compared to the first parent (if N_{refl} = 6000)	25	50	100	200	400

Table B.2 Implementation parameters of static and dynamic mutations in GA design 3 for test case II

Phase variability	0.5°	1°	2°	4°	8°
Non-linear growth (in MCC) point (in generation number)	133	147	162	193	211
State of the system at non-linear growth generation					
Phase divergence at the non-linear growth generation	96	120	138	170	229
Phase divergence as a mutation rate*	0.014	0.017	0.02	0.025	0.033
Mutation rate to be applied**	0.005	0.007	0.01	0.015	0.023
Number of non-degenerated reflections at the non-linear growth generation	0	3	6	9	10
Number of completely degenerate reflections at the non-linear growth generation	90	135	183	224	288
Changes applied to the system					
Generation at which mutations were introduced	130	150	160	170	190
Generation up to static mutations applied	150	170	190	200	220
Generation up to dynamic mutations applied	135	157	170	185	213
Response of the system					
Recovery cycle for GA with static mutations	180	220	>300	>300	>300
Recovery cycle for GA with dynamic mutations	160	190	220	230	250

*Phase divergence / number of centric reflections
**Approximately 1/3 of phase divergence expressed as a mutation rate

Table B.3 The average distribution of centric and centric reflections in survivors over 80 generations. The data was generated from test case II with GA design 2a using skewness as a fitness function. The first row provides reference to the distribution in the first parent.

Generation	Ph_0	Ph_180	Ph_90	Ph_270	Ph_45	Ph_135	Ph_225	Ph_315
First parent	288.00	294.00	300.00	284.00	1386.00	1400.00	1424.00	1490.00
1	287.10	294.91	300.43	283.58	1386.55	1403.05	1422.22	1488.20
2	288.05	293.96	299.86	284.15	1386.31	1399.63	1424.04	1490.03
3	288.14	293.87	299.75	284.26	1386.26	1399.60	1423.88	1490.27
4	288.15	293.86	299.61	284.40	1386.17	1399.73	1423.71	1490.40
5	288.13	293.88	299.41	284.60	1386.30	1399.78	1423.59	1490.35
6	288.10	293.91	299.33	284.68	1386.32	1399.92	1423.59	1490.20
7	288.09	293.92	299.24	284.77	1386.38	1400.15	1423.49	1490.01
8	288.05	293.95	299.18	284.83	1386.41	1400.66	1423.37	1489.58
9	288.03	293.98	299.16	284.85	1386.21	1401.11	1423.23	1489.47
10	287.96	294.05	299.20	284.81	1386.14	1401.38	1423.25	1489.25
11	287.92	294.09	299.23	284.78	1386.21	1401.59	1423.10	1489.13
12	287.90	294.11	299.35	284.66	1386.10	1401.81	1422.87	1489.24
13	287.88	294.13	299.47	284.54	1386.05	1401.90	1422.74	1489.34
14	287.79	294.22	299.53	284.48	1386.10	1402.04	1422.77	1489.11
15	287.68	294.33	299.63	284.38	1386.10	1402.08	1422.67	1489.17
16	287.60	294.41	299.74	284.27	1386.04	1402.38	1422.62	1488.98
17	287.49	294.52	299.92	284.09	1386.23	1402.61	1422.50	1488.68
18	287.37	294.64	300.06	283.95	1386.26	1402.72	1422.56	1488.48
19	287.23	294.78	300.20	283.81	1386.40	1402.77	1422.34	1488.52
20	287.18	294.83	300.37	283.64	1386.48	1402.89	1422.28	1488.36
21	287.10	294.91	300.43	283.58	1386.55	1403.05	1422.22	1488.20
22	287.12	294.89	300.50	283.51	1386.67	1403.18	1422.24	1487.93
23	287.10	294.91	300.52	283.49	1386.68	1402.99	1422.17	1488.18
24	287.18	294.83	300.56	283.45	1386.89	1402.62	1422.35	1488.16
25	287.21	294.8	300.57	283.44	1386.91	1402.71	1422.21	1488.19
26	287.33	294.68	300.61	283.40	1386.92	1402.62	1422.15	1488.33
27	287.44	294.57	300.66	283.35	1387.01	1402.49	1422.02	1488.50
28	287.55	294.46	300.72	283.29	1387.35	1402.21	1421.91	1488.55
29	287.67	294.34	300.64	283.37	1387.57	1402.03	1421.77	1488.65
30	287.77	294.24	300.58	283.43	1387.9	1401.76	1421.50	1488.87
31	287.97	294.04	300.51	283.50	1388.31	1401.48	1421.52	1488.71
32	288.09	293.92	300.56	283.45	1388.51	1401.15	1421.48	1488.88
33	288.31	293.70	300.47	283.54	1388.75	1400.90	1421.40	1488.97
34	288.55	293.46	300.53	283.48	1389.01	1400.70	1421.39	1488.92
35	288.75	293.26	300.43	283.58	1389.32	1400.61	1421.40	1488.68
36	288.95	293.06	300.44	283.57	1389.86	1400.13	1421.36	1488.68
37	289.11	292.90	300.43	283.58	1390.11	1399.76	1421.34	1488.81

38	289.20	292.80	300.36	283.65	1390.20	1399.30	1421.49	1489.03
39	289.23	292.78	300.29	283.72	1390.73	1398.96	1421.58	1488.76
40	289.38	292.63	300.32	283.69	1390.93	1398.80	1421.36	1488.92
41	289.44	292.57	300.30	283.71	1390.93	1398.75	1421.04	1489.30
42	289.53	292.48	300.32	283.69	1390.82	1398.51	1421.19	1489.50
43	289.59	292.42	300.41	283.60	1390.85	1398.67	1421.06	1489.44
44	289.66	292.35	300.40	283.61	1390.74	1398.65	1421.07	1489.56
45	289.69	292.32	300.40	283.61	1390.42	1398.53	1421.20	1489.88
46	289.69	292.32	300.41	283.60	1390.31	1398.47	1421.19	1490.05
47	289.79	292.22	300.45	283.56	1390.44	1398.38	1421.13	1490.07
48	289.83	292.18	300.50	283.51	1390.30	1398.29	1421.05	1490.39
49	289.93	292.08	300.50	283.51	1389.90	1398.47	1421.05	1490.60
50	289.98	292.03	300.50	283.51	1389.74	1398.45	1421.07	1490.77
51	290.05	291.96	300.51	283.50	1389.43	1398.24	1421.37	1490.97
52	290.06	291.95	300.43	283.58	1389.21	1398.30	1421.18	1491.33
53	290.04	291.97	300.42	283.59	1388.75	1397.96	1421.85	1491.47
54	290.16	291.85	300.41	283.60	1388.35	1397.63	1422.45	1491.59
55	290.16	291.85	300.35	283.66	1387.63	1397.34	1423.37	1491.68
56	290.15	291.86	300.34	283.67	1386.98	1397.22	1424.05	1491.77
57	290.21	291.80	300.27	283.74	1386.41	1396.84	1424.61	1492.17
58	290.36	291.65	300.19	283.82	1385.70	1396.56	1425.43	1492.33
59	290.42	291.59	300.11	283.90	1385.26	1396.42	1425.95	1492.40
60	290.50	291.51	300.11	283.90	1384.37	1396.07	1426.91	1492.66
61	290.61	291.40	300.12	283.89	1383.48	1395.74	1427.66	1493.15
62	290.76	291.25	300.18	283.83	1383.21	1395.05	1428.06	1493.69
63	290.88	291.13	300.13	283.88	1382.58	1394.38	1428.90	1494.16
64	291.03	290.98	300.23	283.78	1382.25	1393.94	1429.57	1494.25
65	291.15	290.86	300.22	283.79	1381.38	1393.50	1430.44	1494.70
66	291.26	290.75	300.27	283.74	1380.70	1393.07	1431.45	1494.79
67	291.43	290.58	300.34	283.67	1380.34	1392.55	1432.18	1494.94
68	291.55	290.46	300.24	283.77	1380.18	1392.17	1432.70	1494.97
69	291.73	290.28	300.15	283.86	1380.05	1391.60	1433.30	1495.06
70	291.87	290.14	300.19	283.82	1380.03	1391.04	1433.84	1495.12
71	292.00	290.01	300.17	283.84	1380.01	1390.51	1434.24	1495.26
72	292.10	289.91	300.18	283.83	1379.71	1390.13	1434.72	1495.46
73	292.15	289.86	300.31	283.70	1379.84	1389.82	1434.99	1495.36
74	292.24	289.77	300.43	283.58	1379.83	1389.50	1435.16	1495.52
75	292.32	289.69	300.49	283.52	1379.85	1389.16	1435.24	1495.77
76	292.37	289.64	300.57	283.44	1379.95	1388.71	1435.16	1496.20
77	292.35	289.66	300.59	283.42	1380.10	1388.4	1435.28	1496.24
78	292.37	289.64	300.58	283.43	1380.18	1387.93	1435.55	1496.35
79	292.29	289.72	300.67	283.34	1380.34	1387.51	1435.83	1496.35
80	292.15	289.86	300.71	283.30	1380.62	1387.25	1436.05	1496.09

Supplementary Result Figures

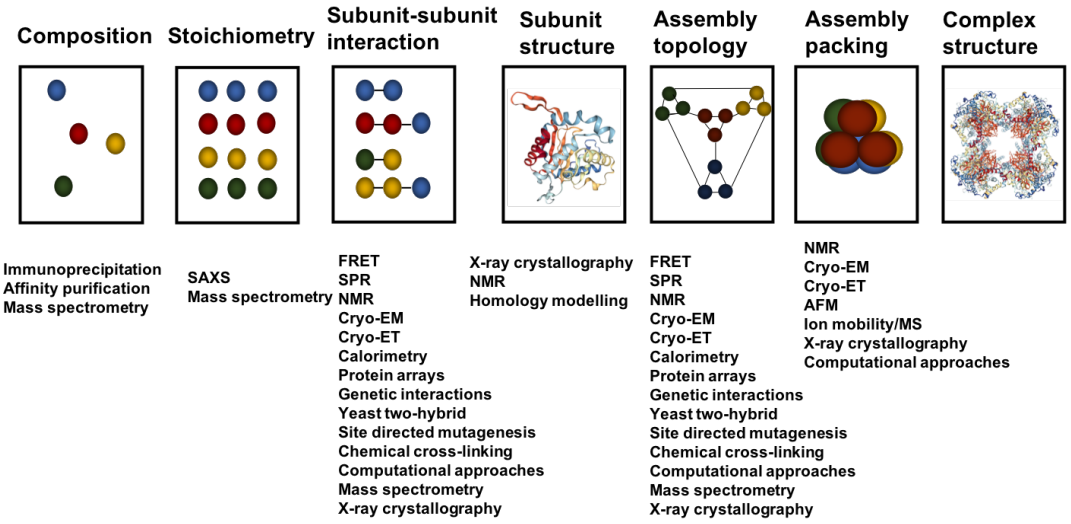


Figure C.1 Overview of different levels of structure determination methods commonly used at different stages of structure solution process. AFM: atomic force microscopy; Cryo-ET: cryo-electron tomography; EM: electron microscopy; FRET; fluorescence resonance energy-transfer; NMR: nuclear magnetic resonance; SAXS; small-angle X-ray scattering; SPR: surface plasmon resonance (Sharon, 2010).

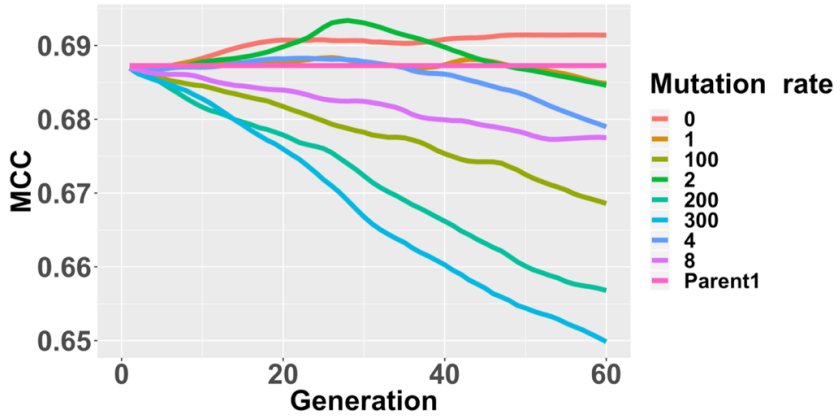


Figure C.2 Effect of mutations on the phase improvement. Mutation rate higher than 2 (shown in green colour) resulted in drastic decrement in the MCC. This plot was generated using test case I data. The GA protocol ran for 60 generations using design 1 parameters. Selection was performed using tournament with size 9. The second variant of the one-point crossover was used for generating children. Different mutation rates (1, 2, 4, 8, 100, 200, 300) were applied. These were compared with

the GA that ran without mutations. MCC was used as a fitness function. The growth in the MCC over generations was monitored to observe the improvement in phases.

Appendix D

List of Hazardous Substances

The presented work is purely theoretical. Therefore, no laboratory experiments were carried out and no hazardous, carcinogenic, mutagenic or toxic substances according to GHS were used.

Appendix E

Acknowledgements

In this short journey of four years, I have met many wonderful people who contributed to this thesis in one way or another.

Firstly, I would like to express my gratitude to Dr. Victor S. Lamzin for giving me an opportunity to work on this project in EMBL. As an alien to the field of crystallography, his guidance and support helped me in understanding various principles and concepts of crystallography. I am grateful to him and his group members for teaching me crystallography and statistics. I am extremely grateful to him and CCP4 guys for giving me an opportunity to travel around the world and teach about ARP/wARP. With this, I got a chance to meet many celebrities in crystallography, enrich my knowledge by participating in different workshops and conferences, gain useful insights from experts on my project and crystallography in general and go on world tour.

In EMBL, the Thesis Advisory Committee (TAC) oversees the progress of PhD project once a year. My TAC members - Prof. Dr. Andrew E. Torda, Dr. Thomas R. Schneider, Prof. Dr. Gerard J. Kleywegt and Prof. Dr. Richard J. Morris contributed greatly to the development of this project. The rich and thorough discussions on GA that went on for hours in TAC meetings provided many useful insights. TAC dinners were equally lively with discussions on science and everything including must watch classic movies. TAC support during a minor hiccup in the last few months was incredible. I am immensely grateful for Andrew Torda for all the support with thesis review and his mentoring during this difficult time. I would also like to thank Thomas Schenider for accepting me into his group, providing thesis related and other administration support in the past few months.

I am immensely grateful for the support of my group members - Grzegorz Chojnowski, Philipp Heuser, Egor Sobolev, Umut Oezugurel, Daria Beshnova and Joana Pereira. A special thanks to Grzegorz for patiently explaining the concepts of crystallography and helping me with the workshop presentations. I would further like to thank Egor for the support with Mapread and weighting factor calculations. Thanks to Joana for her guidance in writing TAC reports and thesis.

Being with an EMBL for the past four years was like being with a family. Thanks to Margret Fischer for taking care of me like a mother, worrying about me when I was sick. Thanks to Emma Ruoqi Xu, Diana Mendes Freire, Natasha Giannopoulou for the lively discussions in the office about kpop, stories from world trips and everything. I also enjoyed relaxing small talk during stressful thesis writing period with my new office colleagues – Christina Ritter, Edu Mullapudi, and George. Thanks to graduate office - Carolina G. Sabate, Matija Grgurinovic and Monika Lachner for providing administration support. Thomas Crosskey, Kim Bartels and Samuel Pazicky – thanks for helping with my little adventure of making an effort to organise a scientific workshop. A big thanks to Tom for reviewing this thesis as a native English speaker.

None of this would have been possible without my family. My parents, Nageswara Rao Kantamneni and Raja Rajeswari Kantamneni, and my sister, Sai Reshma Kantamneni, their care and wishes, although being miles apart, are great source of motivation. To my friends, Kanhaya Lal, Prashanti Mederametla, Arun Kumar Tonduru, Goutami Godavari, Harsha Chinmayi, Padma Chereddy, Pavani Ponnam, Sailaja Kankagiri, Gopi Talari and Ragini Gottipati thanks for supporting throughout this period. A special thanks to Naseemusalam TK for being an inspiration to this PhD. Thanks to Dr. Sachichidanand for making me realise the joy in doing research related to drug discovery. This scientific acheivement would not have been possible without your guidance – Dr. Madhavi Sastry, Dr. Verllarkad Viswanathan, Prof Prasad V. Bharatam.

Lastly, I am very grateful to EMBL for funding my PhD, and providing opportunities to develop personally and professionally.

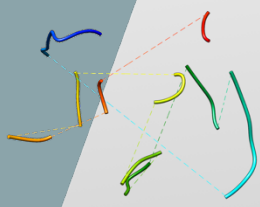
Declaration Upon Oath

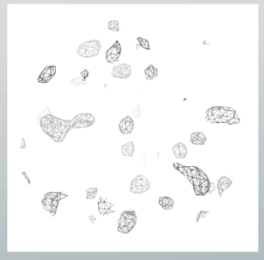
I hearby declare on oath, that I have written the present dissertation by my own and have not used other than the acknowledged resources and aids. The submitted written version corresponds to the version on the electronic storage medium. I hereby declare that I have not previously applied or pursued for a doctorate (Ph.D. studies).

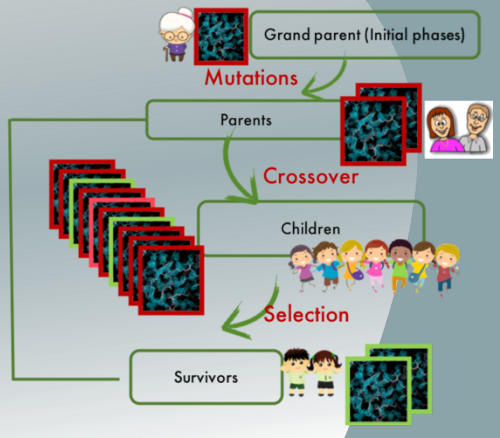
Date: 25 May 2020

Signature: 

(Sravya Mounika Kantamneni)






Grand parent (Initial phases)
Mutations
Parents
Crossover
Children
Iterations
Selection
Survivors