OPTIMIZATION OF ANTENNAS IN THE ASKARYAN RADIO ARRAY USING GENETIC ALGORITHMS

Corey Harris, Luke Letwin, Jacob Trevithick

Advisors: Professor Dean Arakaki, Professor Stephanie Wissel

Table of Contents

Abstract **4**

1 Background and Motivation **5**

1.1 Project Overview *5*

1.2 Importance of Cosmic Neutrinos *6*
1.2.1 Astrophysical Neutrinos 6
1.2.2 Cosmogenic Neutrinos 6

1.3 Landscape of Neutrino Detection *7*
1.3.1 IceCube 7
1.3.2 Askaryan Effect 7
1.3.3 The Antarctic Impulsive Transient Antenna 9
1.3.4 The Askaryan Radio Array 9

1.4 In-Ice Neutrino Detection Geometry *11*

1.5 Why apply optimization? *12*

2 Optimization and the Genetic Algorithm **13**

2.1 Introduction to Optimization and Mathematical Optimization *13*

2.2 Optimization Algorithms Inspired by Real-World Processes *15*

2.3 Introduction to Genetic Algorithms *17*

2.4 The Chromosome Genotype and Phenotype *19*
2.4.1 Binary String Genotypes 19
2.4.2 Continuous Variable Genotypes 20

2.5 Selection Methods *21*
2.5.1 Roulette Selection 21
2.5.2 Tournament Selection 23

2.6 Reproduction Methods *25*
2.6.1 Single-Point Crossover 25
2.6.2 Multi-Point Crossover 26
2.6.3 Uniform Crossover 27
2.6.4 Continuous Variable Average Crossover 28

2.7 Mutations *28*

2.8 Genetic Algorithm Implementation *29*

2.9 Genetic Algorithm Verification *31*
2.9.1 Two-Dimensional Test Case 31
2.9.2 Omnidirectional Radiation Pattern Test Case 32
2.9.3 Directional Radiation Pattern Test Case 35
2.9.4 Discussion of Genetic Algorithm Performance 36

3 Simulation Tools **36**

3.1 AraSim *36*

3.2 Effective Volume *38*

3.3 Directional Beam Pattern Testing *40*

3.4 Simplified AraSim 42

3.5 AraSimLite 43

3.6 AraSimLite2 47

4 Radiation Pattern Optimization with Neutrino Simulations 52

4.1 Optimization Results Using AraSimLite 52

4.1.1 Candidate Solution 1 52

4.1.2 Candidate Solution 2 53

4.1.3 Candidate Solution 3 54

4.2 Optimization Results Using AraSimLite2 55

4.2.1 Candidate Solution 4 55

4.2.2 Candidate Solution 5 56

4.2.3 Candidate Solution 6 57

4.3 Optimization Results Summary and Comparison 58

5 Conclusions 59

5.1 Antenna Recommendations 59

5.2 Future Work 61

5.2.1 AraSim Integration 61

5.3 Concluding Remarks 61

Appendix A: Spherical Harmonic Representations of Antenna Power Patterns 63

Works Cited 67

Abstract

The Askaryan Radio Array (ARA) is an experiment that detects ultra-high energy ($> 10^{18} eV$) neutrinos. Neutrino interactions within the Antarctic ice sheet produce electromagnetic signals in the 150-850 MHz radio frequency (RF) band, detectable by RF antennas. Antennas must be broadband, azimuthally symmetric, and conform to 10 cm diameter boreholes. These constraints create a unique set of antenna design parameters at approximately 1 meter wavelengths. This project's goal is to improve ARA neutrino sensitivity by optimizing antenna radiation patterns. Radiation patterns are modelled by spherical harmonics and expansion coefficients. A genetic algorithm is created to find sets of expansion coefficients that maximize detector effectiveness. Two neutrino simulation tools are created to evaluate antenna performance. Optimization results suggest downward directed radiation patterns with main lobes from $90° < \theta < 120°$ maximize neutrino detection. This feature and broadband requirements lead to the recommendation of a discone antenna for the ARA detector.

1 Background and Motivation

1.1 Project Overview

Ultra-high energy (UHE) neutrinos ($10^{18} - 10^{21}$ eV) are neutrally charged, weakly interacting particles that travel through space from energetic cosmic events. Due to their inert nature, neutrinos are the only feasible known particles for the study of ultra-high energy sources more than approximately 10^7 parsecs (1 parsec $\approx 30.8 \times 10^9$ km) from Earth [1]. However, these same properties complicate UHE neutrino detection. There is an expected interaction rate of less than 1 UHE neutrino per giga-ton of matter per year [1]. Therefore, large and highly sensitive detectors are required.

Neutrino-ice interactions produce in-phase radiation between a few MHz and 1 GHz. In-phase radiation, called coherent radiation, adds constructively. This makes the signal distinguishable from noise. Antennas convert this radiation into electrical potential difference. This project focuses on improving the antennas in the Askaryan Radio Array (ARA), a neutrino experiment located in Antarctica [2]. Genetic algorithms (GA) optimize radiation patterns to maximize neutrino detection. Multiple neutrino interaction simulators (discussed in Chapter 2) interface with the GA to perform the optimization.

Chapter 1 of this paper discusses the importance of UHE neutrinos, detection methods, and current detection experiments. Chapter 2 defines the genetic algorithm and describes optimization techniques and genetic algorithm verification. Chapter 3 discusses neutrino detection simulators. Chapter 4 presents optimization results. Chapter 5 suggests possible antenna designs and future work.

1.2 Importance of Cosmic Neutrinos

Parts of the high-energy universe are not viewable using traditional photon telescopes [3]. Since neutrinos are uncharged, magnetic fields do not affect their propagation direction. Neutrinos are weakly interacting; hence, neutrino absorption while traveling to Earth is rare. Therefore, neutrinos provide unique information about the universe since they travel directly from their source, unimpeded, to Earth. Ultra-high energy neutrino flux can provide information about ultra-high energy particle accelerators [4].

1.2.1 Astrophysical Neutrinos

IceCube (section 1.3.1) was the first experiment to detect astrophysical neutrinos with energies 10^{15} - 10^{17} eV. These are the highest-energy neutrinos detected [5]. These neutrinos are produced outside of the galaxy by two types of sources: cosmic ray accelerators and cosmic ray reservoirs. Cosmic-ray accelerators such as blazars and gamma-ray bursts produce neutrinos directly. Blazars are galaxies with black holes at their centers and emit high energy particles. In cosmic-ray reservoirs, neutrinos are produced from cosmic rays while still confined in the source. Examples of cosmic ray reservoirs include starburst galaxies and galaxy clusters [4].

1.2.2 Cosmogenic Neutrinos

Neutrinos can provide information about rare particles known as ultra-high energy cosmic rays (UHECR). UHECRs have an energy density comparable to the cosmic microwave background (CMB), a remnant of the early universe. Studying UHECRs provides information about their sources [6].

UHECRs interact with cosmic microwave background photons to produce cosmogenic neutrinos, among other particles [7]. This phenomenon is called the GZK effect. Measuring the cosmogenic neutrino flux tests UHECR production and propagation models. Their flux and energy

composition depends on UHECR composition (protons or heavier nuclei), the red shift of UHECR sources, and how UHECRs are accelerated [6]. Neutrino experiments discussed in section 1.3 measure the maximum cosmogenic neutrino flux value [4].

1.3 Landscape of Neutrino Detection

1.3.1 IceCube

The IceCube Neutrino Observatory is a neutrino detector located near the South Pole. IceCube detected over 80 high-energy neutrinos, including the first high-energy astrophysical neutrinos, with energies between 100 TeV and a few PeV [8]. For reference, 1 PeV is two orders of magnitude greater than energies generated in the Large Hadron Collider (13 TeV) at CERN [9]. IceCube uses optical sensors to monitor 1 km^3 of ice for neutrino interactions. The detector is located approximately 2.5 km below the ice surface [8]. IceCube has detected the highest energy neutrinos to date (1 PeV) but has yet to detect an ultra-high energy cosmogenic neutrino [10].

1.3.2 Askaryan Effect

In 1962, Gurgen Askaryan predicted that neutrally charged particles such as neutrinos could create coherent radio emission. When a neutrino collides with a nucleon, a particle shower can occur. Through processes such as Compton scattering and annihilation of positrons, this shower of secondary particles acquires a 20% net negative charge asymmetry [11]. If the charge is traveling faster than the phase velocity of light in the medium, the medium will emit electromagnetic radiation. The signal is enhanced in dielectrics with high densities. For wavelengths longer than the shower in the transverse direction, L_T (typically 3-5 cm in ice), the signal is coherent (Figure 1-1) at frequencies less than 1 GHz. Coherent signals are distinguishable from noise. Therefore, in-ice Askaryan radiation detectors must receive signals less than 1 GHz.

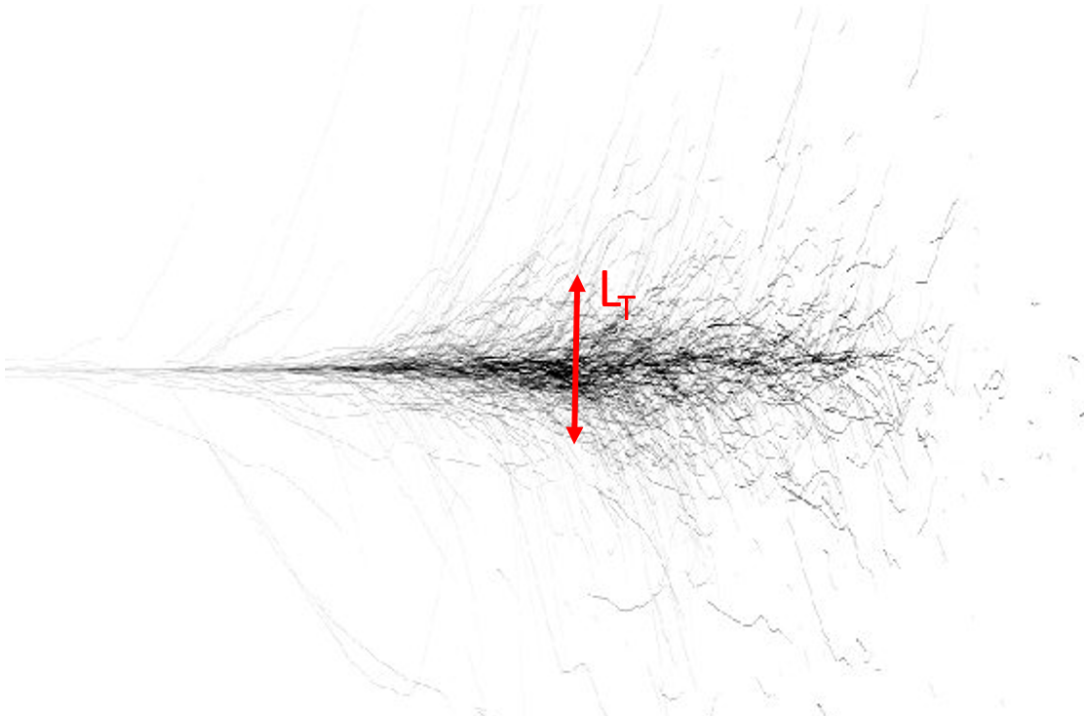

Figure 1-1: Illustration of transverse direction of a particle shower [32].

The Askaryan effect was first observed in a beam test at SLAC National Accelerator Laboratory, originally named Stanford Linear Accelerator Center, in 2001 [11]. The experiment produced a time-domain signal shown in Figure 1-2.

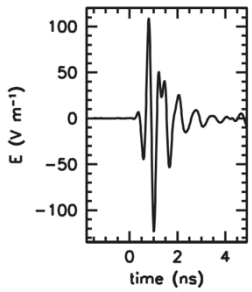

Figure 1-2: Radio signal from Askaryan Radiation measured at SLAC National Accelerator [33].

Neutrino detection experiments commonly use ice due to its long attenuation length (approximately 1000 m at 300 MHz [12]). Attenuation length is the travel distance that decreases

signal power by 1/*e*. Long attenuation lengths allow for neutrino detection in large volumes. Antarctica's large quantity of ice creates a large volume for neutrino interactions to occur, making it an ideal location for neutrino experiments.

1.3.3 The Antarctic Impulsive Transient Antenna

The Antarctic Impulsive Transient Antenna (ANITA) is an airborne neutrino detection experiment. ANITA flies 35-37 km above the Antarctic continent carried by a NASA long-duration balloon. ANITA observes signals using vertically and horizontally polarized horn antennas operating from 200 MHz to 1200 MHz. ANITA observes approximately 1.5×10^6 km^3 of ice [11]. Antarctic weather conditions limit ANITA's flight duration to approximately 30 days per year.

1.3.4 The Askaryan Radio Array

The Askaryan Radio Array (ARA) is a neutrino detection experiment sensitive to cosmogenic neutrinos located near the South Pole. ARA37, the completed ARA detector, will consist of 37 stations in a triangular array with a 2 km separation distance, shown in Figure 1-3. There are currently five ARA stations deployed as of January 2018.

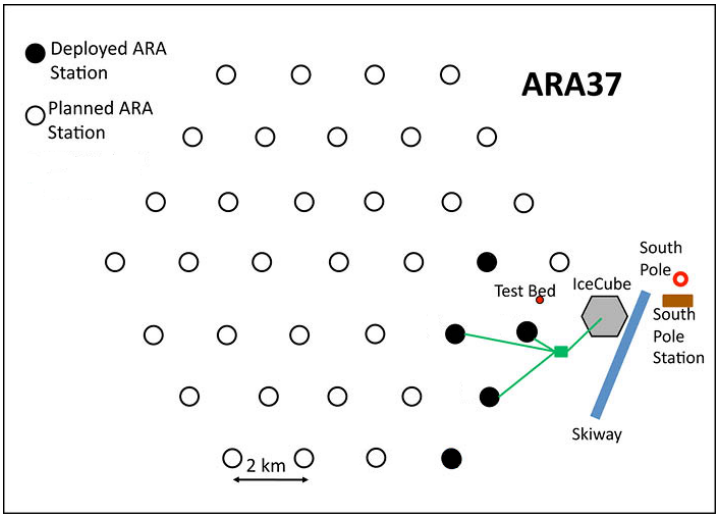

Figure 1-3: Deployed and planned ARA station locations, January 2018 [2].

Each ARA station consists of 16 antennas distributed among four "strings". Each string contains four antennas in a 10 cm diameter borehole located 200 m below the ice surface. Figure 1-4 shows the string arrangement. Antenna pairs are vertically and horizontally polarized. Each station has two additional strings of antennas for calibration. Fiber optic cable transmits antenna signals to trigger and data recording electronics. A typical trigger requires a received power threshold 5 times greater than the mean power in 3 or more of the 16 station antennas [11]. When this criterion is met, a trigger will signal a digitizer to record data. Measured Askaryan radiation can be traced to determine neutrino event location and ultimately the neutrino's propagation direction. This project optimizes the vertically polarized antennas radiation patterns to improve neutrino detection. The baseline design for a single ARA station is shown in Figure 1-4.

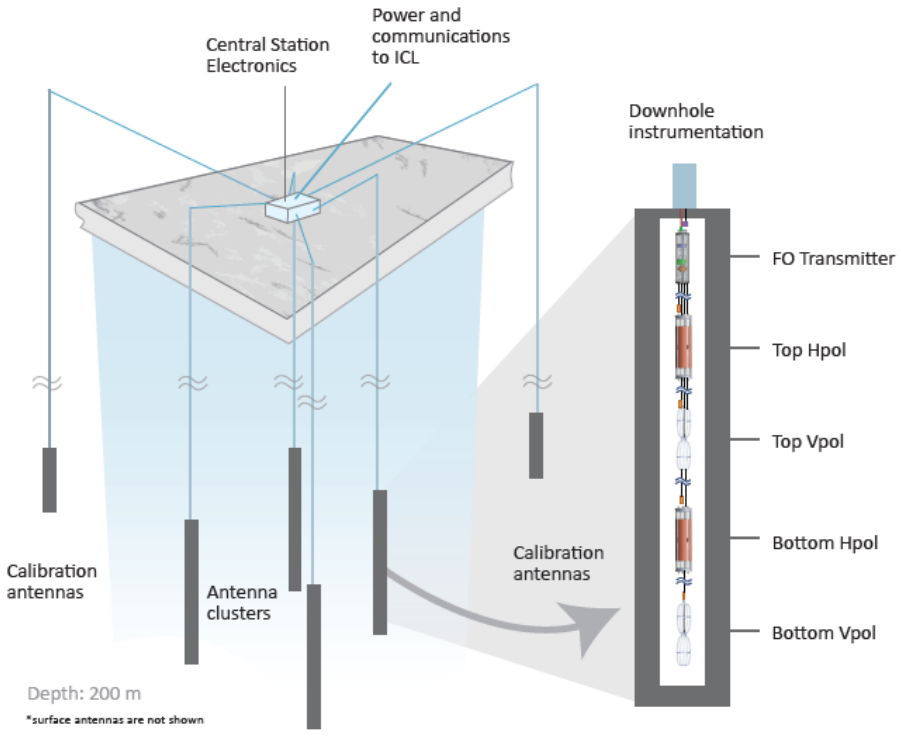

Figure 1-4: Baseline design of an ARA station [2].

ARA and IceCube are sensitive to neutrinos with different energies: 10^{15} to 10^{17} eV [8] and 10^{16} to 10^{19} eV [2], respectively. Each experiment attempts to determine the neutrino flux at their respective energy ranges. ARA covers less ice than ANITA but ANITA is limited to 30-100 day flights due to extreme weather in Antarctica. In a 30-day flight, ANITA-II was expected to detect 5.8 neutrinos while ARA37 is expected to detect 48.7 neutrinos in a year, according to the most optimistic models [2].

1.4 In-Ice Neutrino Detection Geometry

The Antarctic ice sheet and Askaryan radiation create a specific geometry for in-ice neutrino detection. Neutrinos traveling upward with respect to the ice surface are expected to interact with the Earth prior to reaching the Antarctic ice sheet. Neutrinos traveling downward with respect to the ice surface are unlikely to interact with the ice sheet prior to passing the detector (Figure 1-5).

These conditions decrease the likelihood of detecting neutrinos coming from below and above the detector, respectively.

Askaryan Radiation is emitted in a cone defined by a medium-dependent angle. The Cherenkov angle defines this cone,

$$\cos(\theta_{cone}) = \frac{1}{n\beta} \tag{1-1}$$

where θ_{cone} is the Cherenkov angle, n is the medium's refractive index, and β is the particle shower's velocity relative to the speed of light in vacuum [7]. Antarctic ice refractive index decreases with depth [12], which computationally intensifies radio propagation modeling.

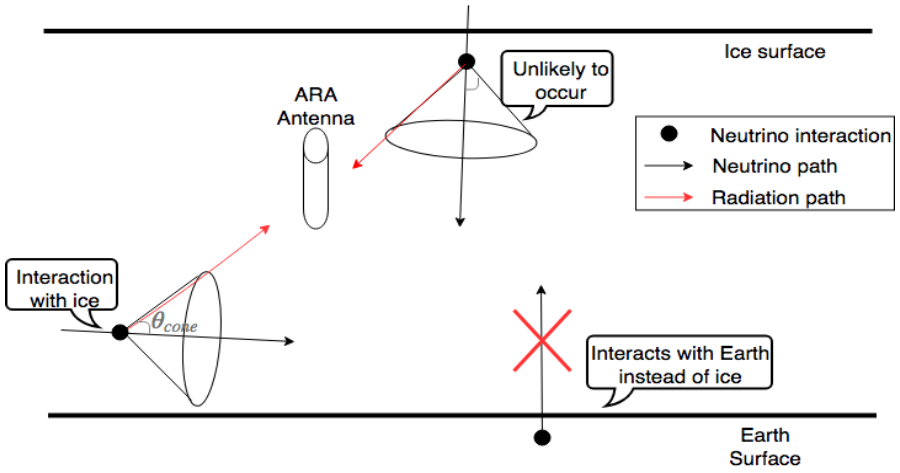

Figure 1-5: Multiple neutrino entry locations and interactions.

Figure 1-5 shows three types of neutrino interactions. The interaction location is depicted with a bold dot. The interaction with the horizontal neutrino path is detectable because it interacts with the ice and the radiation travels to the antenna. A neutrino produces radiation when it interacts with a proton or neutron. Hence, interactions occur within the volume of a material rather than the surface. Askaryan radiation from the interaction with the Earth is not detected (red X) because the radio signal is absorbed in rock. The downward traveling neutrino is unlikely to be detected

because it will likely pass the detector before interacting with the ice. This project aims to match the in-ice neutrino geometry to the ARA detector design.

1.5 Why apply optimization?

Neutrino detection is improved by maximizing ARA antenna gain at neutrino incident angles. When analytical solutions do not exist, optimizing radiation patterns enhances antenna performance.

2 Optimization and the Genetic Algorithm

2.1 Introduction to Optimization and Mathematical Optimization

Optimization is the process of selecting the best solution to a problem among a set of potential solutions [13]. The goodness of a particular solution is described by an objective function or performance index [14]. Mathematically, optimization is the process of solving the problem

$$minimize\ or\ maximize\ f(\vec{x}), \qquad \vec{x} = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n \tag{2-1}$$

while satisfying equality or inequality constraints on the column vector $\vec{x}$ [15]. Each element x_i of $\vec{x}$ is a design variable, $f(\vec{x})$ is the objective function, and the optimal solution $\vec{x}^*$ is the vector that maximizes/minimizes the objective function while satisfying all constraints. The objective function $f(\vec{x})$ can be analytically defined, a simulation result, or experimental data.

For example, consider the problem

$$minimize\ f(\vec{x}) = x_1^2 + x_2^2, \qquad \vec{x} = [x_1, x_2]^T \in \mathbb{R}^2 \tag{2-2}$$

subject to constraint

$$\boldsymbol{A}\vec{x} \geq 1, \qquad \boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2-3}$$

The constraint can be rewritten

$$x_2 \geq 1 - x_1 \tag{2-4}$$

The objective function has a global minimum $f(\vec{x} = 0) = 0$ but this solution does not satisfy equation 2-3. The optimal solution is $\vec{x}^* = [0.5, 0.5]^T$ because it minimizes the objective function while satisfying constraints. Figure 2-1 shows the objective function contour map, the boundary set by the constraints, and the optimal solution location.

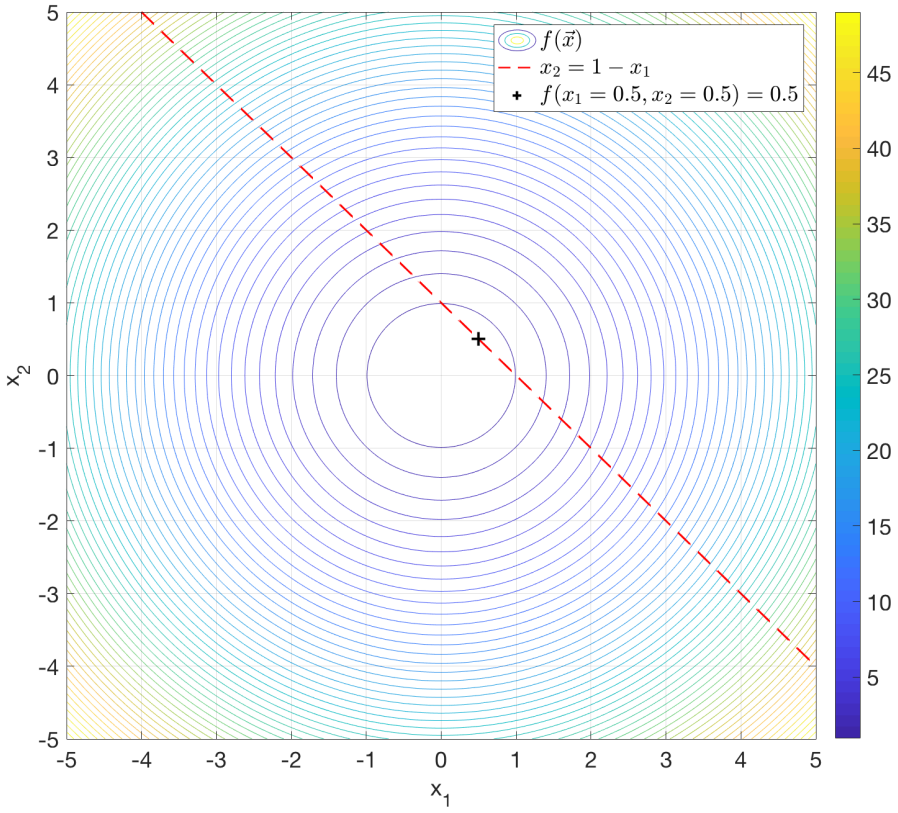

Figure 2-1: Contour map of the objective function, the constraints boundary (red line), and the optimal solution (black cross).

Many mathematical techniques and algorithms help find solutions to optimization problems. Optimization methods generally fall into two categories: gradient-based and gradient-free. Gradient-based optimization uses derivative information from the objective function to find minima or maxima. Gradient-free optimization uses an algorithm or probabilistic model to

find solutions. An example of gradient optimization in an unconstrained problem is the method of steepest descent [16]. The gradient of the objective function $\boldsymbol{\nabla} f$ is calculated at a point. The gradient $\boldsymbol{\nabla} f$ is in the direction of steepest ascent normal to a contour. A new point is chosen in the direction of $\boldsymbol{\nabla} f$ or $-\boldsymbol{\nabla} f$ to approach a maximum or minimum, respectively. The method of steepest descent is iterative, with each iteration approaching a local solution. Convergence to a global solution is not guaranteed unless the search begins in the neighborhood of the global solution [16]. The method of Lagrange multipliers is commonly used in constrained gradient optimization [13, 15, 16].

Gradient search methods can converge quickly, are computationally efficient, and guarantee a local solution is found [15]. However, the derivative of the objective function is not always available or continuous. In the case of a simulation, a potential solution vector $\vec{x}_j$ is mapped to $f(\vec{x}_j)$, but the closed-form expression for the objective function $f(\vec{x})$ is not explicitly defined. Noisy objective functions can have derivatives with large discontinuities. In such cases, gradient-free optimization is preferred over gradient-based searches [14]. Gradient-free optimization methods leverage the speed of modern computing to test potential solutions on $f(\vec{x})$. The simplest optimization method is a brute-force approach, in which $f(\vec{x})$ is evaluated at every potential solution on a user-defined domain. For high-dimensionality problems, brute force is impractical due to the large number of potential solutions. Other types of gradient-free optimization include random searches, pattern searches, and Bayesian optimization [17].

2.2 Optimization Algorithms Inspired by Real-World Processes

Several optimization algorithms exist that replicate nature's ability to find optimal states in complex systems. Examples of optimization algorithms inspired by real-world processes include swarm algorithms, simulated annealing, and evolutionary algorithms. Each algorithm relies on

techniques and probabilities to find optimal solutions instead of gradient information. This allows for optimization with noisy or multi-modal objective functions. Furthermore, optimization is possible when a closed-form expression for $f(\vec{x})$ does not exist [18].

The biological dynamics such as bird flocking, fish schooling, and insect swarming inspire swarm algorithms [19]. Particle swarm optimization is a successful implementation of swarm optimization. A set of potential solutions P (called a population) is randomly created. Each potential solution $\vec{x}_i$ (called an agent) in P is evaluated on the objective function $f(\vec{x})$. The best fitness in the entire group is recorded, and each agent's personal fitness is recorded. A velocity vector $\vec{v}$ is calculated from each agent to determine its location in the next iteration. Each agent moves towards its personal best location and the best location in P. Movement allows for exploration of the parameter space. Each agent will eventually occupy the same location, indicating convergence. This location is the optimum solution $\vec{x}^*$ [19].

Metallurgical annealing inspires the simulated annealing algorithm. In metallurgical annealing, samples are repeatedly heated and cooled to reduce defects in a material [20]. Simulated annealing algorithms are useful in problems with high-dimensionality and noisy objective functions. The algorithm is unlikely to find a global solution but has a high probability of finding a high-quality solution [18]. Simulated annealing creates a potential solution $\vec{x}_i$ (called the sample) and evaluates this solution on the objective function. The sample starts at simulation temperature T. The solution is randomly modified and evaluated on the objective function. It will improve or diminish in quality compared to the previous state, which is summarized by the quantity ΔD. ΔD is negative for an improved solution, and positive for a diminished solution. The algorithm will accept the change in state if $e^{\frac{\Delta D}{T}} > R(0,1)$, where $R(0,1)$ is a random number on the interval $[0, 1]$. The algorithm is more likely to accept a diminished solution at high temperature and reject

one at low temperature [18]. Over time the sample is cooled, forcing the algorithm to only accept changes in state that increase solution quality. This forces the algorithm to converge to a solution. The initial state at high temperature allows the algorithm to search the solution space, and convergence is guaranteed as the sample cools [20].

Evolutionary algorithms are a class of gradient-free optimization algorithms that are inspired by biological evolution [13, 17]. Evolutionary algorithms create a set of potential solutions P called a population and perform operations on P to improve the solution quality for the entire set. These operations generate new solutions by combining features of higher quality solutions in P. This simulates the mating of fit individuals in a biological population. The entire population evolves over several generations, each time improving solution quality [21]. NASA successfully used evolutionary algorithms to generate unique antenna designs subject to radiation pattern constraints [22].

2.3 Introduction to Genetic Algorithms

A genetic algorithm is an optimization technique modelled after modern genetic theory. In a genetic algorithm, a random set of potential solutions to a problem is produced. This set is called a population P. Each potential solution in P is a chromosome. The genetic algorithm evaluates each chromosome on the objective function and assigns a fitness score. Some chromosomes are eliminated in favor of replacements. The remaining members are then selected using a decision-making process called a selection method, which pairs chromosomes together for reproduction. Paired chromosomes are called parents. Through another process called the reproduction method, the parents create one or two new chromosomes, called children. These children are evaluated on the objective function, assigned a fitness score, and inserted into the population. Children are produced until the original size of the population is restored. This new population represents the

next generation of potential solutions. The process repeats until a user-defined convergence condition is met.

Genetic algorithms model natural selection by simulating 'survival of the fittest' evolution. Chromosomes with low fitness scores are eliminated and better-performing chromosomes have increased probability for parent selection. Children may be randomly mutated prior to evaluation, resulting in features contained in neither parent. This process introduces new genetic material to the population and promotes exploration of the parameter space. Over several generations, the population quality will increase because highly-fit chromosomes share their genetic information with the rest of the population.

Genetic algorithms have successfully solved optimization problems in real-world applications [23]. Haupt and Haupt [21] summarize the genetic algorithm's advantages as an optimization tool, including:

- Derivative information is unnecessary
- A wide domain of the objective function's surface is searched and sampled
- Optimizes surfaces where concavity is not guaranteed
- Provides several potential solutions to the objective function
- Works with constrained search parameters
- Works with numerically generated data, experimental data, or external simulation tools

Genetic algorithms are a useful tool for optimization where only the relative quality of a solution is known. However, they require many objective function evaluations to assign fitness scores to chromosomes [24]. This can result in long optimization times for problems with computationally involved fitness assignments, such as external simulation tools. Fitness assignments are parallelizable, reducing optimization time on parallel-capable machines.

2.4 The Chromosome Genotype and Phenotype

Optimization variables must be compatible with the genetic algorithm. In binary string genetic algorithms, all optimization parameters are represented by a single binary string. The genetic algorithm manipulates the binary string rather than the optimization variables directly. The programmer decides how this binary string maps to the optimization problem. The binary string is the chromosome genotype. The chromosome phenotype are the optimization variables evaluated on the objective function. Genetic algorithms are not constrained to binary string genotypes. Continuous variable genotypes are often used because they map to the optimization problem more directly [21]. Binary string and continuous variable genotypes are both referred to as $\vec{x}$ hereafter.

2.4.1 Binary String Genotypes

Consider the problem of maximizing a function $f(\vec{x})$, $\vec{x} = [x_1, x_2]^T \in \mathbb{R}^2$ constrained to the interval $\vec{x} \in [-1, 1]$. In a binary string genetic algorithm, optimization parameters x_1 and x_2 are encoded into a binary string input for the genetic algorithm [21]. Parameter resolution is determined by

$$resolution = \frac{domain\ size}{2^n} \tag{2-5}$$

where n is the number of bits used to represent a parameter. If both parameters are encoded into an 8-bit binary string, the parameter resolution is $\frac{2}{2^8} = 7.8125 \times 10^{-3}$. The parameters map to binary strings:

$$x_1 = -1.0 \quad \rightarrow \quad x_1' = 0b00000000$$

$$x_2 = +0.9921875 \quad \rightarrow \quad x_2' = 0b11111111$$

The entire genotype is the concatenation of the two binary strings $[x_1' \; x_2']$. A point is then mapped to a binary string:

$$\vec{x} = [-1.0, +0.9921875]^{\mathrm{T}} \rightarrow 0b0000000011111111$$

In this example, the string '$000000011111111'$ is the genetic algorithm genotype, but the vector $\vec{x}$ is the phenotype. The objective function evaluates the phenotype, but the genetic algorithm operates on the binary string. Potential solutions are discretized in 7.8125×10^{-3} steps and the parameter space is limited to $2^{16} = 65536$ unique solutions. Longer binary strings are used for greater precision.

2.4.2 Continuous Variable Genotypes

In continuous variable genotypes, the optimization parameters map to continuous variables. Considering the same problem as before, the optimization parameters now map to continuous variables:

$$x = -1 \rightarrow x' = -1.0$$

$$y = +1 \rightarrow y' = +1.0$$

The parameter resolution is limited by the computer's resolution. One benefit of the continuous variable genotype is close or even exact mapping between the genotype and phenotype [23]. However, the parameter space becomes much larger. In binary strings, the number of possible solutions is limited by the length of the genotype, but there are nearly infinite potential solutions with a continuous variable genotype. Continuous variable genotypes allow for wider variety of crossover and mutation processes, discussed in sections 2.6 and 2.7.

2.5 Selection Methods

The selection method is responsible for choosing parents that will pass on genes to offspring. Selection methods should, on average, favor better-performing individuals in a population. However, it is important to maintain diversity in a population to prevent premature convergence to local solutions [25]. Selection methods can choose less fit solutions to prevent one particular solution from dominating the entire population.

Selection methods are separated into two categories: fitness proportionate and elitist selection. In a fitness proportionate selection scheme, the probability that an individual is selected to become a parent is proportional to its fitness score. Better-performing individuals are more likely to be selected than other members of the population. If the fitness of a high-performing individual is not suitably large, the probability that this individual is not selected for reproduction increases. This could result in a low-quality final solution. The elitist selection scheme attempts to remedy this problem by placing more pressure on the selection method to choose fit individuals and prevent unfit individuals from passing on genes. Intentional bias towards better solutions is referred to as elitism. Two selection methods are discussed: the roulette method, a fitness-proportionate scheme, and tournament selection, an elitist scheme.

2.5.1 Roulette Selection

The roulette selection method imitates the random nature of a roulette wheel. Each individual in a population is assigned a section of a roulette wheel with size directly proportional to the individual's fitness score, and the wheel is "spun" to randomly select an individual. The roulette selection method is illustrated in Figure 2-2.

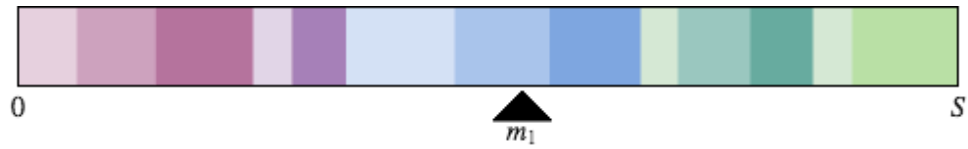

Figure 2-2: Roulette selection of an individual from a population.

The number S is equal to the sum of the population's fitness. Each member is allocated a portion equal to the size of its fitness score on the interval [0, S]. A marker m_1 is chosen from a real uniform distribution on [0, S]. m_1 selects the individual. In genetic algorithms that use two parents for reproduction, the roulette method is implemented as follows [26]:

1. F_n is the fitness of the n^{th} individual in the population. In a population of N individuals, the sum of fitness scores S is

$$S = \sum_{n=1}^{N} F_n \tag{2-6}$$

2. Random markers m_1 and m_2 are chosen from a real uniform distribution on the interval [0, S], where $m_1 < m_2$.
3. The first selected individual is the i^{th} member of the population, where i is the first integer index to satisfy the criteria

$$\sum_{n=1}^{i} F_n \geq m_1 \tag{2-7}$$

4. The second selected individual is the j^{th} member of the population, where j is the first integer index to satisfy the criteria

$$\sum_{n=1}^{j} F_n \geq m_2 \tag{2-8}$$

The probability that the i^{th} individual in a population is selected P_i is

$$P_i = \frac{F_i}{S} \tag{2-9}$$

P_i is proportional to an individual's fitness F_i, making this selection method fitness proportionate [27]. This process is equivalent to placing two markers randomly on a roulette wheel and spinning the wheel once. It is convenient to implement in software because only one running sum must be performed to select both individuals. Once the first individual is selected, the program continues adding fitness scores until the second marker has been reached. A potential downside is the case where $m_1 + m_2 < F_i$. In this case, one individual could meet both criteria, and be selected as both parents. This resulting offspring would be identical to the parent. If one individual is particularly fit, this might lead to loss of diversity early in the algorithm, where each offspring is a copy of this parent. This would result in premature convergence without adequately exploring the solution space. To prevent this from occurring, selection should be performed without replacement [21]. That is, once the first parent is chosen, it is temporarily removed from the population until a second parent has been chosen, preventing one parent from dominating the selection process.

2.5.2 Tournament Selection

The tournament selection method compares individuals against one another. In the simplest tournament, two members of the population are selected at random with equal probability. The fitness scores of these individuals are directly compared, and the individual with the larger fitness score is selected to reproduce [21]. Guaranteed selection of the fittest individual classifies the tournament method as elitist. The benefit of an elitist scheme is that it prevents loss of good genetic material. A single tournament is called a k-way tournament, where k is the number of individuals

competing in the tournament. The tournament described above is a 2-way tournament; only two individuals compete. A k-way tournament can be implemented in software as follows [26]:

1. Generate an array A of all potential individual indices for population of N individuals.

$$A = \{0, 1, 2, \dots, N-1\}$$

2. Shuffle the array A.
3. Read the first k indices from the shuffled array and place the individuals of the population at these indices in a tournament.
4. Compare the fitness scores of each individual in the tournament and choose the fittest individual for reproduction.

Multiple tournaments can be held with or without replacement to choose several individuals for reproduction. The probability that an individual is selected depends only on the individual's rank in comparison to the other members of the tournament pool. In general, the probability that an individual is selected to enter the tournament pool is k/N. This means the probability that the fittest individual in the entire population is selected for reproduction is k/N. This allows the programmer to control the level of diversity within the population. In optimization problems with lengthy fitness computation times or where a local maximum/minimum is desired rather than a global maximum/minimum, set k sufficiently large to promote rapid convergence to a local solution [27]. An excessive k value can lead to loss of population diversity because it increases the probability that only the most fit individual chosen.

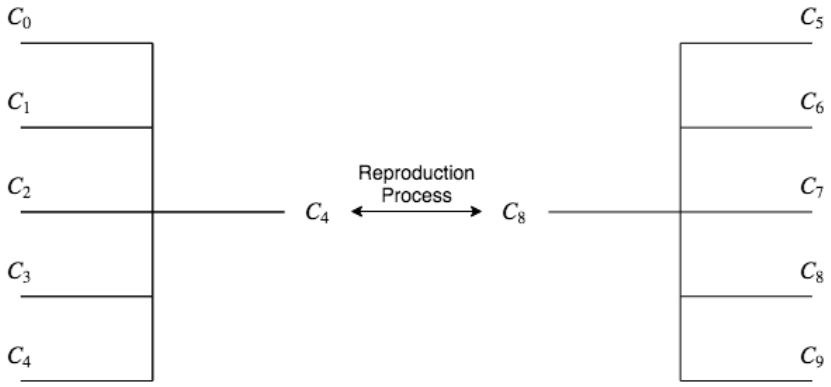

Figure 2-3: Two 5-way tournaments select two individuals for reproduction.

Figure 2-3 shows an example selection of two parents with two 5-way tournaments. Ten chromosomes are chosen randomly from the population without replacement. The fittest chromosomes in each tournament are selected for reproduction.

2.6 Reproduction Methods

Reproduction methods control how genes are passed from parents to offspring. Genetic algorithms typically use only two parents for reproduction, but high quality solutions can be produced by using more than two parents to produce a single offspring [23]. For this project, crossover methods are limited to two parents.

2.6.1 Single-Point Crossover

Single-point crossover generates at most two different children from two different parents. A random integer marker is chosen from a real uniform distribution on [0, L_x]. Let $p_1$and p_2 represent the parents, and c_1 and c_2 represent the children. Before the marker location, the genotype of c_1 will match the genotype of p_1, and the genotype of c_2 will match the genotype of p_2. After the marker, the genotype of c_1 will match the genotype of p_2, and the genotype of c_2 will match the genotype of p_1 [23]. Figure 2-4 shows this process.

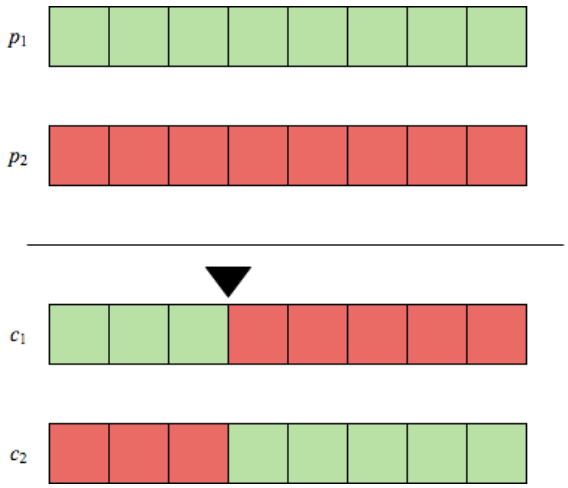

Figure 2-4: Single-point crossover with crossover point at position 3.

2.6.2 Multi-Point Crossover

Multi-point crossover expands upon the same process used in single-point crossover, but with an arbitrary number of crossover points. Children receive genetic material from a unique parent. Each time a crossover marker is reached, a child receives its genetic material from the other parent [23]. This is implemented by generating an array A of all potential marker indices

$$A = \{0, 1, 2, \dots, L_x - 1\}$$

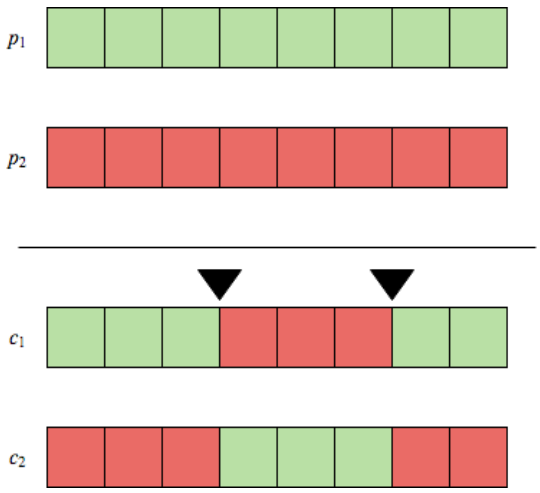

Figure 2-5: Multipoint crossover with crossover points at poistions 3 and 6.

where L_x is the genotype length. The array is shuffled. If *n* crossover points are desired, the first *n* values are read from the shuffled array. Figure 2-5 illustrates a two-point crossover with crossover points at position 3 and 6.

2.6.3 Uniform Crossover

Uniform crossover interleaves the parents' genetic information randomly among the two children. At each gene location, a random number is drawn from a real uniform distribution on the interval [0, 1]. If the number is greater than a threshold t = 0.5, the child c_1 inherits the gene from parent p_1, and c_2 receives the gene from p_2. If the number is below the threshold, c_1 inherits from p_2, and c_2 inherits from p_1 [23]. With a threshold of 0.5, genes from both parents are randomly and uniformly interleaved within the children. The threshold is adjusted to promote more inheritance from a single parent. Figure 2-6 shows an example uniform crossover with $t = 0.5$.

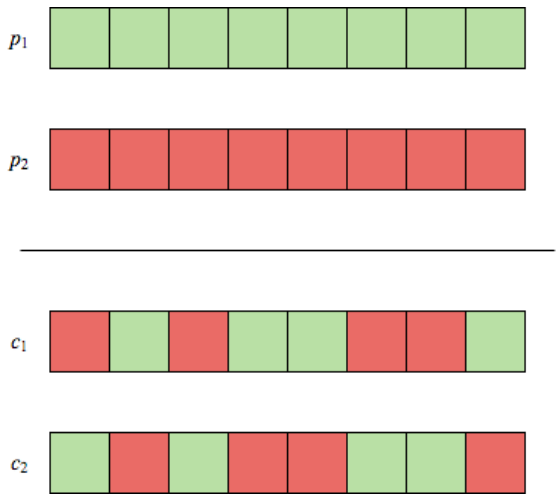

Figure 2-6: Example uniform crossover with t = 0.5.

2.6.4 Continuous Variable Average Crossover

This crossover type is unique to continuous variable genotypes. In continuous variable average crossover, each gene is averaged between the two parents to create one or two children [21]. For each gene, the weighted average of the parent genes is calculated and is assigned to the child. The new gene is biased towards a particular parent with a coefficient a which lies on the interval [0.5, 1]. The genotype of two children produced through floating point average crossover is

$$\vec{x}_{c_1} = a\vec{x}_{p_1} + (1-a)\vec{x}_{p_2}$$

$$\vec{x}_{c_2} = (1-a)\vec{x}_{p_1} + a\vec{x}_{p_2}$$

If $a = 0.5$, the children are identical and located halfway between each parent. If $a = 1$, c_1 is identical to p_1 and c_2 is identical to p_2.

2.7 Mutations

A mutation is an operator that acts on a chromosome's genotype. It randomly introduces new genetic material to the population, encouraging exploration of the parameter space. Each gene in the genotype has probability p_m of being mutated. In many genetic algorithms, p_m is equal to $1/L_x$, where L_x is the genotype length. This encourages mutation of only one gene at a time [21]. If p_m is too large, it can change too much of the genotype at once, potentially moving a solution away from a maximum/minimum. In binary string genotype representations, the mutation inverts a bit at the mutation location, shown in Figure 2-7.

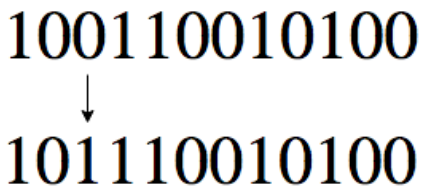

Figure 2-7: Binary string mutation selects the 3rd bit in the genotype for mutation.

Continuous variable genotypes enable more complex mutations. Rather than inverting a bit, a new gene is chosen at random from a Gaussian distribution. If the gene x_i has been chosen for mutation, the new gene is chosen at random from a Gaussian distribution with mean $\mu = x_i$ and variance σ^2, where σ is the standard deviation. The variance is user-defined and depends on the optimization problem [23]. The variance can be increased or decreased to promote searching different sizes of the parameter space.

2.8 Genetic Algorithm Implementation

A genetic algorithm written in C++ and is used for optimization. The continuous variable genotype is chosen for ease of implementation. The first iteration of the genetic algorithm, referred to as GA1, is outlined below. The genetic algorithm operates on fixed population size of N chromosomes for G generations, where N and G are user inputs.

1. Generate a population of N random chromosomes.
2. Score the population on the objective function.
3. Generate $N/2$ new chromosomes by selecting two parents using roulette selection and continuous variable average crossover with crossover parameter $a = 0.8$.
4. Generate $N/6$ new chromosomes by selecting one individual from roulette selection and Gaussian mutation with parameters $p_m = 1/L_x$ and $\sigma^2 = 1.0$.

5. Generate $N/6$ new chromosomes by selecting two parents from two, 6-way tournaments and continuous variable average crossover with crossover parameter $a = 0.8$.
6. Generate $N/6$ new chromosomes by selecting one individual from a single, 6-way tournament selection and Gaussian mutation with parameters $p_m = 1/L_x$ and $\sigma^2 = 1.0$.
7. Score population on the objective function.
8. If G generations have been completed, terminate the program. Else, return to step 3.

The size of the chromosome genotype and the objective function is application dependent and are input by the user at runtime. The continuous variable average crossover is chosen to encourage exploration of the parameter space between high-performing solutions. Mutation parameter p_m is chosen as $1/L_x$ to discourage mutations from occurring in multiple dimensions at once.

2.9 Genetic Algorithm Verification

The genetic algorithm is tested on multiple objective functions with known solutions to verify performance.

2.9.1 Two-Dimensional Test Case

Consider the problem of finding the ordered pair (x^*, y^*) that maximizes the function

$$f(x, y) = 19.81 - x\sin(4x) - 1.1y\sin(2y) \tag{2-12}$$

subject to the constraints $x, y \in [0, 10]$. The function has a global maximum

$$f(x^* = 9.03899,\ y^* = 8.66819) = 38.3645 \tag{2-13}$$

and many local maxima and minima. The genetic algorithm is tested on this objective function, with fitness assigned by evaluating the objective function. Any solution created that does not fall within the interval $x, y \in [0, 10]$ is assigned a fitness of 0. The genotype length is $L_x = 2$, with $x \rightarrow x_1$ and $y \rightarrow x_2$. The algorithm operates on $N = 80$ individuals with termination criterion of $G = 30$ generations. Figure 2-8 shows the objective function surface and optimization summary. The blue line in the optimization summary shows the maximum fitness in the population at each generation. The green line shows the population's average fitness at each generation.

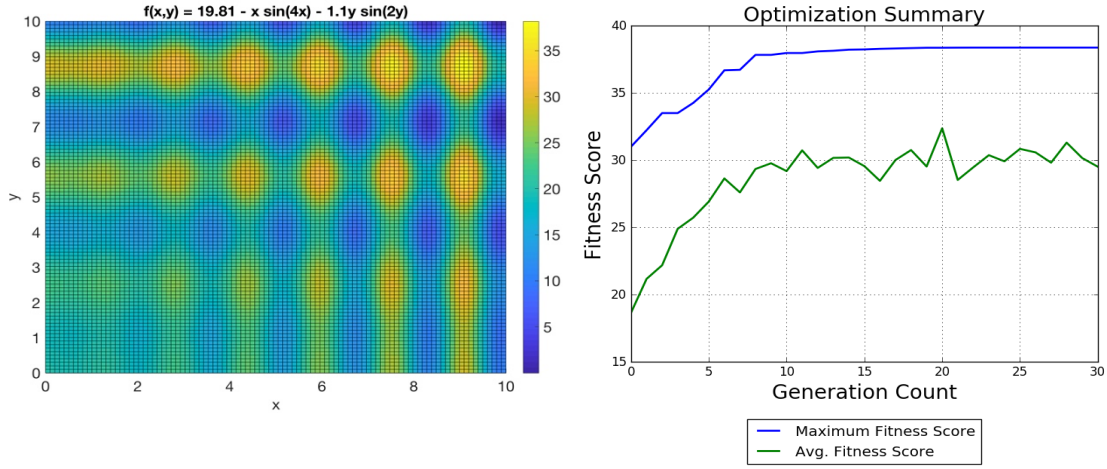

Figure 2-8: Objective function surface (left) and optimization summary (right).

The maximum fitness stops increasing, indicating convergence. The genetic algorithm finds $\vec{x}^* = [9.03899,\ 8.66806]^{\mathrm{T}}$ after 30 generations. This solution has a fitness of 38.3653 and a distance $d = 0.00759$ from the known global maximum. Convergence could be improved by allowing the genetic algorithm to optimize for a larger generation count.

2.9.2 Omnidirectional Radiation Pattern Test Case

The genetic algorithm is tested on radiation pattern optimization. An azimuthally symmetric radiation pattern $G(\theta)$ can be expressed as a weighted sum of spherical harmonics

$$G(\theta) = a_0 Y_0^0(\theta) + a_1 Y_1^0(\theta) + \cdots a_n Y_n^0(\theta), \qquad \theta \in [0, \pi] \tag{2-14}$$

and is subject to conservation of energy and non-negative constraints. Spherical harmonics are chosen to model antenna gain because they enforce conservation of energy by fixing $a_0 = 2\sqrt{\pi}$. Enforcing constraints and using the first 13 harmonics simplifies the model to

$$G(\theta, \vec{a}) \cong 2\sqrt{\pi}\, Y_0^0(\theta) + a_1 Y_1^0(\theta) + \cdots a_{12} Y_{12}^0(\theta), \qquad \theta \in [0, \pi] \tag{2-15}$$

$$\vec{a} = [a_1, a_2, \ldots, a_{12}]^T \tag{2-16}$$

An in-depth discussion of spherical harmonics as models for radiation patterns and necessary constraints is found in Appendix A.

An expansion vector $\vec{a}$ can be chosen to model low-gain, azimuthally symmetric radiation patterns. To maintain consistent vocabulary, let $\vec{a} \to \vec{x}$, where the $\vec{x}$ is a continuous variable genotype in a genetic algorithm. The function

$$f[G(\theta, \vec{x})] = \frac{1}{\pi \max[G(\theta, \vec{x})]} \int_0^{\pi} G(\theta, \vec{x}) d\theta \tag{2-17}$$

is maximized by a radiation pattern $G(\theta, \vec{x}) = C$, or a radiation pattern with no directivity. Thus, a vector $\vec{x}^*$ exists that maximizes $f[G(\theta, \vec{x})]$ such that the radiation pattern $G(\theta, \vec{x}^*)$ is

omnidirectional. $f[G(\theta,\vec{x})]$ is bounded on the range [0, 1] and the optimal solution is known to be $\vec{x}^* = 0$. Furthermore, the optimal solution that satisfies conservation of energy constraints is $G(\theta,\vec{x}) = 1.0$.

The genetic algorithm is configured with genotype length $L_x = 12$. The first generation is sampled from a uniform distribution with $\vec{x} \in [0,10]$. Furthermore, to enforce conservation of energy on radiation patterns, a constraints function is added to the algorithm that corrects solution vectors

$$constrain(\vec{x}) = \begin{cases} \dfrac{\vec{x}}{1 - \min[G(\theta,\vec{x})]} & G(\theta,\vec{x}) < 0 \\ \vec{x} & G(\theta,\vec{x}) \geq 0 \end{cases} \tag{2-18}$$

The new algorithm, GA2, is a modification of GA1 that supports radiation pattern constraints. GA2 uses the following steps:

1. Generate a population of N random chromosomes.
2. Score population on the objective function.
3. Generate $N/2$ new chromosomes by selecting two parents from roulette selection and continuous variable average crossover with crossover parameter $a = 0.8$.
4. Generate $N/6$ new chromosomes by selecting one individual from roulette selection and Gaussian mutation with parameters $p_m = 1/L_x$ and $\sigma^2 = 1.0$.
5. Generate $N/6$ new chromosomes by selecting two parents from two, 6-way tournaments and continuous variable average crossover with crossover parameter $a = 0.8$.
6. Generate $N/6$ new chromosomes by selecting one individual from a single, 6-way tournament selection and Gaussian mutation with parameters $p_m = 1/L_x$ and $\sigma^2 = 1.0$.
7. Constrain each new chromosome according to equation 2-18.

8. Score population on the objective function.
9. If stopping criteria is reached, terminate the program. Else, return to step 3.

The genetic algorithm, GA2, will find $\vec{x}$ that maximizes $f[G(\theta,\vec{x})]$. In this problem, the genetic algorithm phenotype is a radiation pattern $G(\theta,\vec{x})$ and the genotype is $\vec{x}$. Figure 2-9 shows a summary of the genetic algorithm optimization and the radiation pattern of the most fit individual of the final generation. The genetic algorithm finds $\vec{x}^*$ shown in Table 2-1 after 100 generations. This solution has a fitness of 0.99 and gain averaged over all polar angles of 0 dBi.

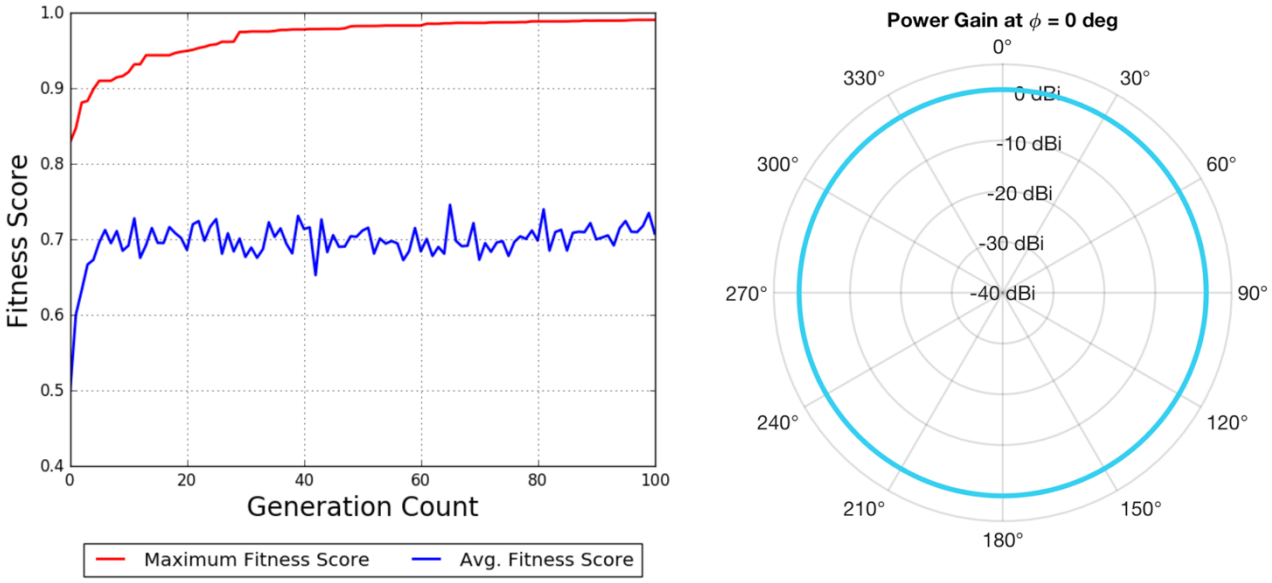

Figure 2-9: Optimization summary (left) and radiation pattern of the fittest individual (right) for omnidirectional test case..

Table 2-1: Genotype of fittest solution in omnidirectional test case

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-0.0036	-0.016	-0.0093	-0.0034	-0.0052	0.0047	-0.0043	-0.0034	0.0022	-0.0095	-0.0013	-0.0027

The decibel magnitude plot of the solution resembles an isotropic radiator, which is the expected solution for an omnidirectional radiation pattern that obeys conservation of energy.

2.9.3 Directional Radiation Pattern Test Case

The genetic algorithm is tested on its ability to find directional radiation patterns. The phenotype $G(\theta, \vec{x})$ and genotype $\vec{x}$ used in section 2.9.2 is used again. The new fitness function is

$$f[G(\theta, \vec{x})] = \frac{\int_{\theta_1}^{\theta_2} G(\theta, \vec{x}) d\theta}{\int_0^{\pi} G(\theta, \vec{x}) d\theta}, \qquad 0 < \theta_1 < \theta_2 < \pi \tag{2-19}$$

where the range of $f[G(\theta, \vec{x})]$ falls within [0, 1] for non-negative $G(\theta, \vec{x})$. An optimal radiation pattern is a rectangular function $\Pi[\frac{1}{\theta_2 - \theta_1}(\theta - \frac{\theta_2 + \theta_1}{2})]$ with a fitness of 1.0. This solution is unattainable due to the use of a finite number of spherical harmonics in the radiation pattern model.

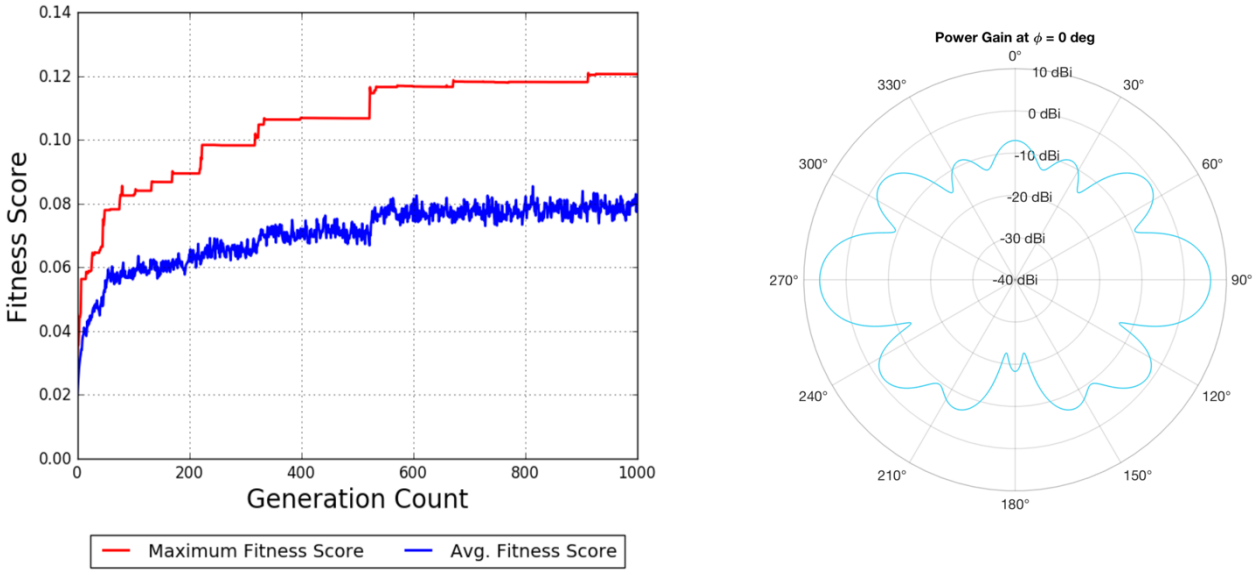

Figure 2-10: Optimization summary (left) and radiation pattern of the fittest individual (right) for directional test case.

The genetic algorithm attempts to find $\vec{x}$ that maximizes $f[G(\theta, \vec{x})]$ with $\theta_1 = 85°$ and $\theta_2 = 95°$. Figure 2-10 shows a summary of the genetic algorithm optimization and the radiation pattern of the most fit individual of the final generation. The first generation is randomly distributed on $\vec{x} \in [-5, 5]$. The genetic algorithm finds $\vec{x}^*$ shown in Table 2-2 after

1000 generations. This solution has a fitness of 0.120. The associated radiation pattern has maximum gain of 6.23 dBi at 90° with half-power beam width of 20°.

Table 2-2: Genotype of fittest solution in directional test case

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-0.0032	-2.76	-0.0365	1.91	-0.0316	-1.76	-0.0839	1.76	-0.0459	-1.37	-0.0522	0.53

Fitness could be further improved by using higher-order spherical harmonics. This would increase the dimensionality of the optimization problem, resulting in longer optimization times.

2.9.4 Discussion of Genetic Algorithm Performance

The genetic algorithm finds optimal solutions to the two-dimensional and omnidirectional test cases. Furthermore, it finds a sub-optimal solution to the directional test case. The sub-optimal solution is acceptable because it outperforms the initial generation of randomly selected solutions, indicating the final solution is better than randomly sampled solutions. The genetic algorithm's ability to produce directional and omnidirectional solutions implies it can create a wide range of low-gain radiation patterns. Note spherical harmonics do not allow discontinuities, making the global maxima of the directional test case unobtainable.

3 Simulation Tools

3.1 **AraSim**

The fitness score assignment in this optimization utilizes an in-ice Monte Carlo simulation software, AraSim. It simulates the ARA detector performance and was developed by the ARA collaboration. The software simulates high-energy neutrino (~10^{18} eV) interactions in the Antarctic ice sheet that produce electromagnetic (EM) showers resulting in radio frequency radiation. The radiation from in-ice EM showers are modeled by Askaryan radiation described

in Ref. [6]. The interactions are evenly distributed and confined to a 3 km-radius cylindrical volume centered around the detector [28]. Figure 3-1 illustrates this volume.

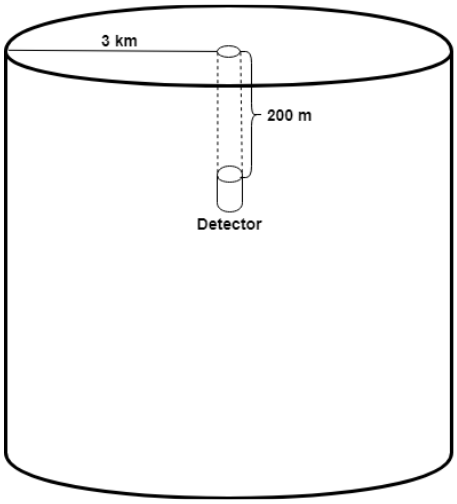

Figure 3-1: Neutrino interactions are uniformly distributed within a cylindrical volume.

The incoming neutrino travel direction randomly distributed over a 4π solid angle. Radio emission propagation is modeled using ray tracing methods that determine the path length from the interaction to the detector. The ray tracing method models the depth dependent index of refraction of Antarctic ice as a starting at $n = 1.3$ at the ice surface and reaching $n = 1.8$ at 200 m beneath the surface. The emitted EM waves bend through the ice from the interaction site to the antenna. AraSim calculates the viewing angle, EM wave polarization at the antenna, travel time to the antenna, ice attenuation factor, and Fresnel refraction factor [28]. AraSim also models the system electronics, noise waveforms, and the time-domain trigger discussed in depth in Ref. [28].

The detector antennas are gain values over the 83.33 MHz to 1083.53 MHz band, in a Numerical Electromagnetics Code (NEC) simulation file. The NEC files describe an antenna model's gain and phase. Each file is simulated in AraSim at 60 frequencies with 16.67 MHz

frequency increments, starting at 83.33 MHz. AraSim calculates the detector's fitness score, denoted as the effective volume. The effective volume quantifies the volume of ice monitored by the neutrino detector (described explicitly in section 3.2). This effective volume is appended to the NEC file to clearly associate its value with the radiation pattern. The effective volume is then used in the genetic algorithm as an associated fitness metric.

3.2 Effective Volume

The effective volume quantifies the volume of ice monitored by a neutrino detector. It accounts for trigger threshold, geometry, neutrino-nucleon interaction cross-section (or neutrino interaction probability), signal strength, and EM propagation in the ice. To first order, the volume monitored by an ARA station is a cylinder centered on a station, but the directional dependence of the interaction probability and arrival direction detection probability modify this volume. The effective area of a detector is defined as,

$$A_{eff}(E) = \frac{V_{gen}(E)}{N_{gen}(E)} \frac{1}{L_{int}(E)} \sum_{i,trig} w_i \tag{3-1}$$

where $V_{gen}(E)$ is AraSim's neutrino interaction scan volume, $N_{gen}(E)$ is the number of neutrino interactions, $L_{int}(E)$ is the neutrino interaction length within the ice, and w_i is the weight (probability) of a neutrino interaction at a given detection trigger threshold summed over all possible neutrino interactions, i [1]. $L_{int}(E)$ is defined as,

$$L_{int}(E) = \frac{m_{nucleon}}{\rho_i \sigma} \tag{3-2}$$

where $m_{nucleon}$ is the nucleon mass, ρ_i is the Earth layer i mass density, and σ is the neutrino-nucleon cross-section [7]. $L_{int}(E)$ is calculated at the user-defined neutrino interaction energy.

The trigger threshold defines the required EM wave voltage to trigger the detector. Weight is defined on as

$$w_i(E) = \prod_j e^{\frac{-l_j}{L_{int,j}(E)}} \tag{3-3}$$

where l_j is the path length through material j. Weight is limited to [0, 1] and is directly proportional to interaction probability over the path the neutrino takes through the Earth. The expected mean weight of all simulated interactions is 0.50. A neutrino and its associated weight is only included in the summation if it triggers the detector in the simulation. This definition is valid if the effective volume is defined as,

$$V_{eff}(E) = \frac{V_{gen}(E)}{N_{gen}(E)} \sum_{i,trig} w_i. \tag{3-4}$$

Detector effective volume is the chosen fitness function because it defines the sensitivity of a given neutrino experiment. Effective volume is not calculated as a function of the number of neutrino interactions. However, increased interactions enhance effective volume precision and decreases uncertainty. Figure 3-2 shows increasing number of simulated neutrinos corresponds to decreasing error bars.

AraSim reads the NEC file gain values line-by-line until the end of the file. Files varying from NEC file specific formatting cause file input errors. Therefore, it is necessary to modify the AraSim file input code to only read the NEC file gain values and not the appended fitness score. Figure 3-2 demonstrates that modified AraSim and original AraSim assign similar effective volumes using the standard ARA Vpol and Hpol antenna radiation patterns, as expected. Changing the method for reading files should not affect the calculated effective volume.

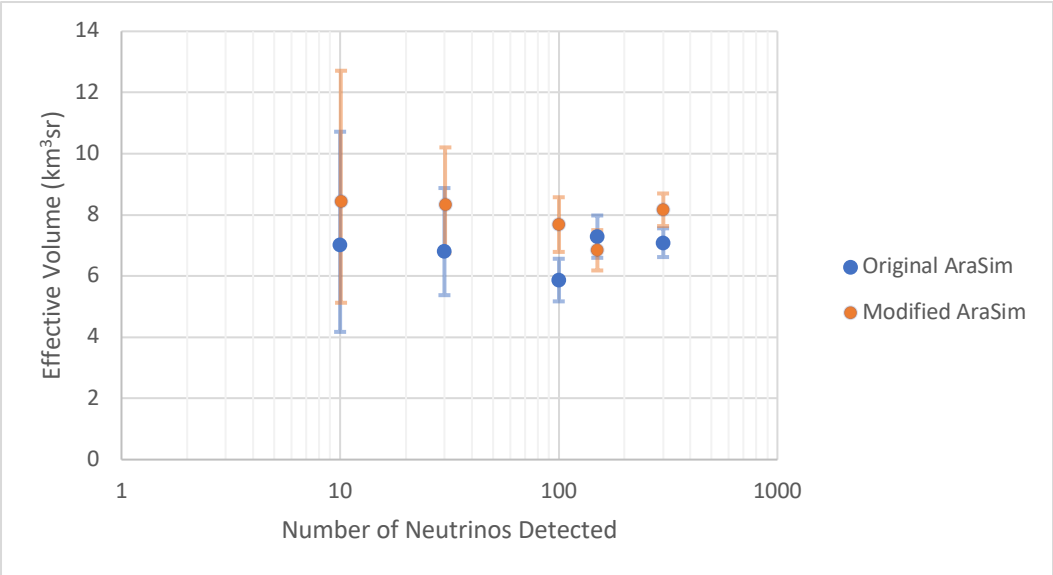

Figure 3-2: Effective volume and error bars vs. number of detected neutrinos, using Vpol bicone and Hpol quad-slot antennas.

3.3 Directional Beam Pattern Testing

ARA's vertically and horizontally polarized antennas (Vpol and Hpol, respectively) are tested in AraSim to determine the most likely neutrino arrival directions. Only directional beam radiation patterns are used for one polarization at a time. The angular test range is 0° to 180° with respect to the vertical in 30° increments, where 0° corresponds to a neutrino path direction originating from the ice surface, directly above the antenna. These 7 unique beam patterns are executed in AraSim. The same beam pattern is used at all 60 frequencies. This test benchmarks which angles of θ correspond to the largest number of neutrino interactions. Each pattern is symmetric about the z-axis (azimuthally symmetric) and has approximately 9 dBi of gain in the specified direction: 0°, 30°, 60°, … , 180°. Each directional pattern has an approximate first null beam width (FNBW) of 30°. Figure 3-3 shows two sample beam patterns used in this test focused at 30° and 120°. Figures 3-4 and 3-5 indicate antennas with main beam lobes between 60° and 120°

yield higher effective volumes. This indicates an optimized radiation pattern's main beam is directed between 60º and 120º.

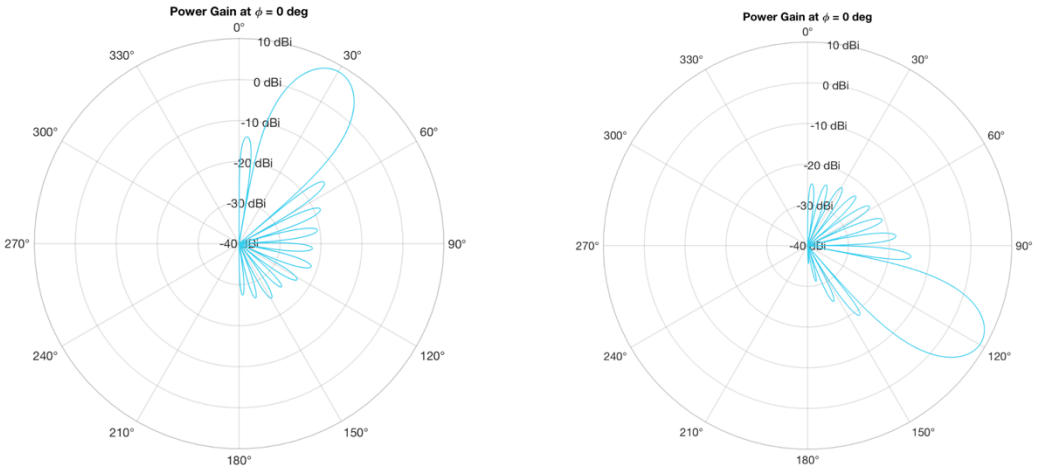

Figure 3-3: Sample directional beam patterns with main beam centered about 30° (left) and 120° (right) with respect to vertical and approximate first null beam width (FNBW) = 30º.

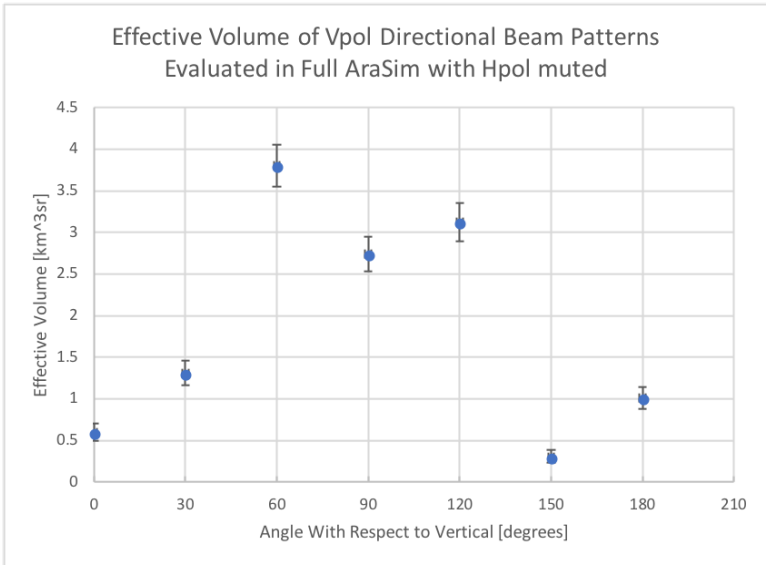

Figure 3-4: Effective volume of Vpol directional beam patterns with respect to vertical. Hpol radiation pattern gain is set to zero at all angles. Patterns incremented in 30º steps, approximate FNBW = 30º.

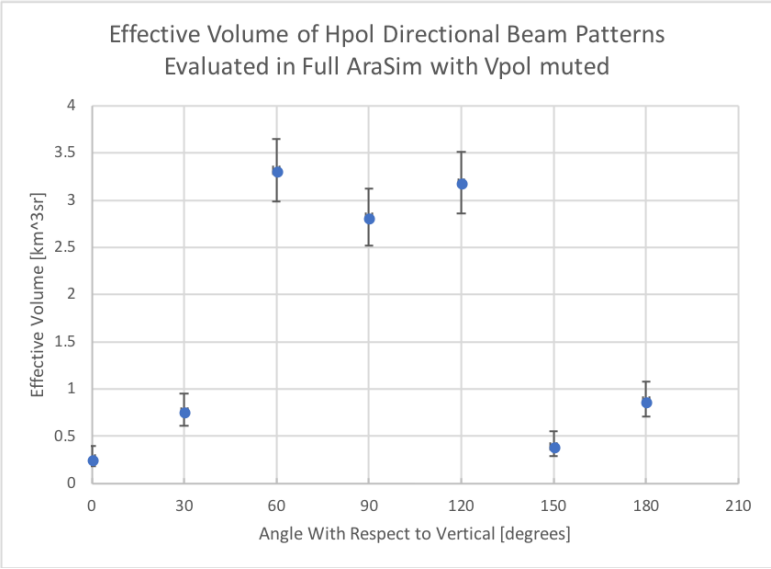

Figure 3-5: Effective volume of Hpol directional beam patterns with respect to vertical. Vpol radiation pattern gain is set to zero at all angles. Patterns incremented in 30º steps, approximate FNBW = 30º.

3.4 **Simplified AraSim**

AraSim requires approximately 30-45 minutes per radiation pattern. This extrapolates to multi-day full program simulations when executing multiple genetic algorithm generations. To reduce simulation time, a simplified version of AraSim is created, AraSimLite. The neutrino energy used in AraSimLite is 10^{18} eV. The redesigned simulation tool, created in collaboration with Ohio State University (OSU), simplifies the fitness assignment by omitting the simulation's ray-tracing, noise waveforms, signal polarization, and ice modeling. AraSimLite is created by simulating the neutrino-nucleon interaction cross-sections and samples possible neutrino interaction points within the ice [29]. It then randomly distributes neutrino directions over 4πsr and produces corresponding weights and neutrino path directions.

AraSim requires approximately 30 minutes to simulate 10,000 neutrino interactions and calculate fitness, while AraSimLite requires approximately 0.3 seconds. Therefore, AraSimLite evaluates radiation pattern fitness approximately 6000 times faster than AraSim. Executing the

program loop with AraSimLite in place of AraSim serves as an intermediate test of GA performance with a neutrino-based fitness function. A full GA test is the program loop executed with AraSim as the neutrino interaction simulator. AraSimLite results identify potentially optimal detector radiation patterns that serve as initial radiation patterns for GA-interfaced AraSim. Two versions of AraSimLite are constructed for testing: AraSimLite and AraSimLite2.

3.5 AraSimLite

A block diagram of AraSimLite is shown below in Figure 3-6.

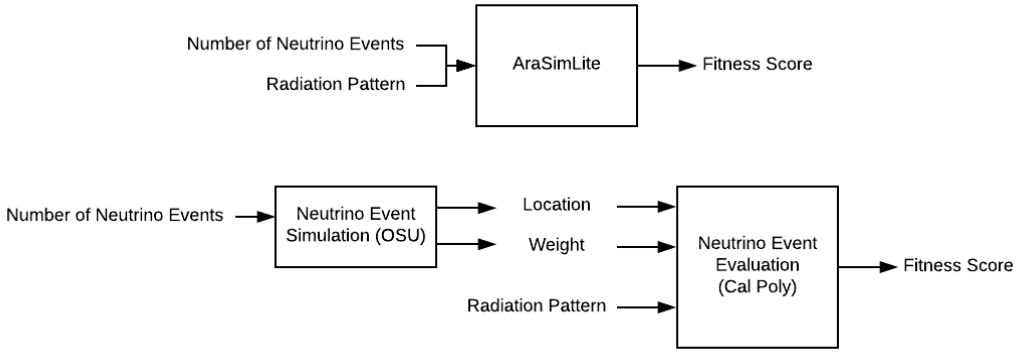

Figure 3-6: AraSimLite Block Diagram condensed (top) and expanded (bottom).

The Neutrino Event Simulation block in Figure 3-6 generates the neutrino event 'location' relative to the ARA detector location, and the 'weight.' 'Location' is the event's three-dimensional position in rectangular coordinates. 'Weight' is the probability that the neutrino event occurs. This is the probability of neutrino-ice interaction vs. Earth absorption. The Monte Carlo simulation evaluates a user-defined number of events and returns an array of weights and locations.

AraSimLite uses the data array to calculate a fitness score for each antenna radiation pattern. The Neutrino Event Evaluation converts the location to polar coordinates, R_i and θ_i,

shown in equations 3-5 and 3-6, respectively. Antenna gain $g(\theta_i)$ is computed from the radiation pattern at angle θ_i. Fitness score is calculated as

$$fitness\ score = \sum_{i=1}^{N} \begin{cases} \omega_i, & if\ \frac{g(\theta_i)}{R_i^2} > r_{th} \\ 0, & otherwise \end{cases} \tag{3-5}$$

where R_i, denoted as R in Figure 3-7, is

$$R_i = \sqrt{x^2 + y^2 + z^2} \tag{3-6}$$

and θ_i, denoted as θ in Figure 3-7, is

$$\theta_i = \cos^{-1}\left(\frac{z + 200m}{R}\right) \tag{3-7}$$

z is the vertical location of the interaction ($z < 0$ below the ice surface), N is the number of neutrino events, and w_i is weight. The minimum detectable signal for ARA electronics is the threshold value r_{th} in units of $\frac{1}{m^2}$ representing $\frac{1}{R^2}$ EM wave power loss. Figure 3-7 shows the AraSimLite geometry. Note z is positive above the ice; therefore, the z location of neutrino events occurring within the ice are negative.

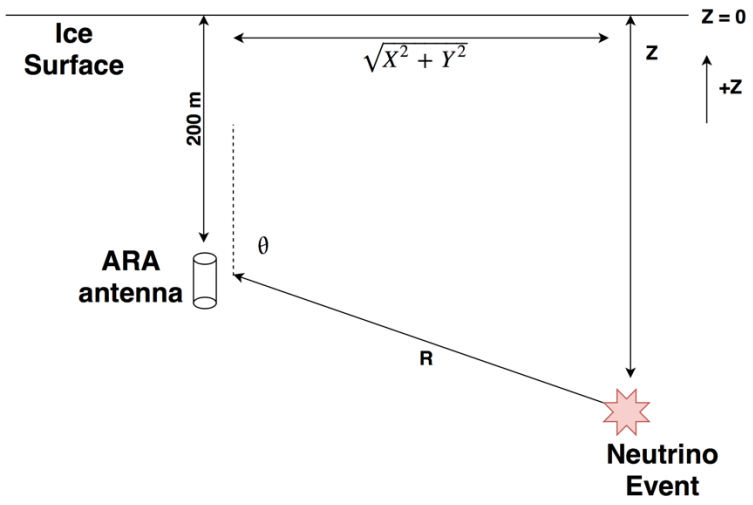

Figure 3-7: AraSimLite geometry.

Equation 3-5 is characterized by varying the threshold value, r_{th}. Figure 3-7 shows the threshold. In simulation, r_{th} is adjusted to increase fitness score sensitivity to radiation patterns. Using an isotropic radiation pattern in AraSimLite, the threshold value r_{th} is swept to determine its effect on fitness score. A threshold value that is too low allows many radiation patterns to achieve high fitness score. However, if the threshold value is too high, then minimal neutrino events will be detected. Therefore, it is necessary to choose a threshold value that does not saturate the population with high fitness radiation patterns, but still allows for sufficient event triggering. The threshold value corresponding to 50% of the maximum fitness score for an isotropic radiator in Figure 3-8 is $r_{th} = 150 \frac{1}{m^2}$.

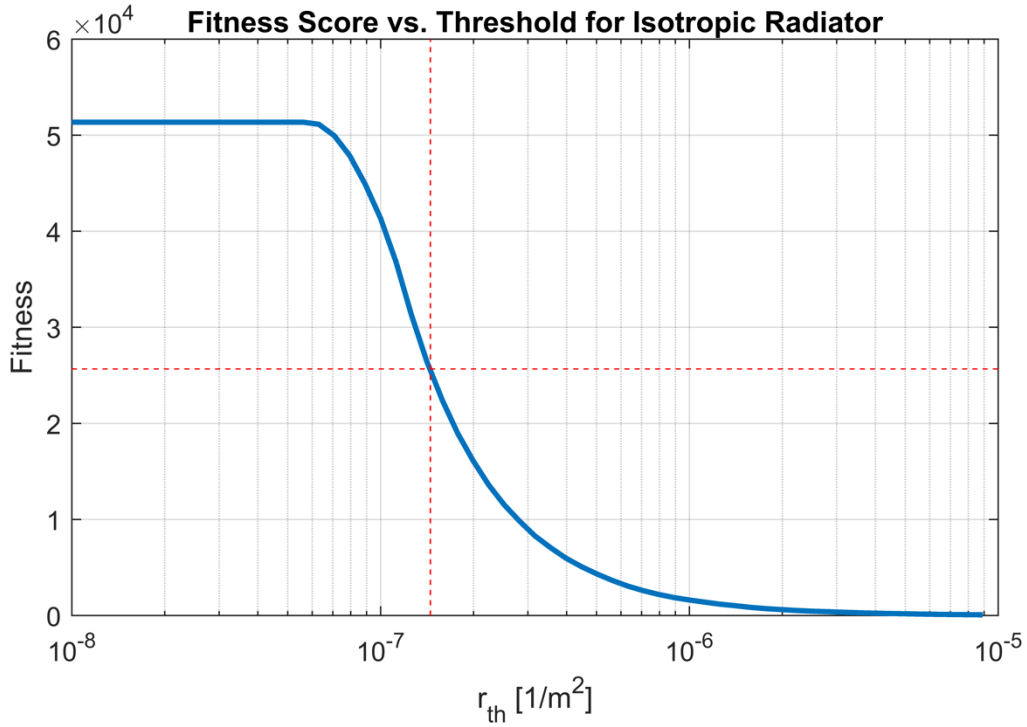

Figure 3-8: AraSimLite Fitness score vs. r_{th} threshold, isotropic radiator. Red dotted line intersection corresponds to 50% fitness point.

Decreased computation time sacrifices neutrino simulation accuracy. Table 3-1 shows AraSimLite and AraSimLite2 computations compared to AraSim.

Table 3-1: Computation Comparison: AraSim, AraSimLite, AraSimLite2. Red boxes indicate included computations.

Computation	AraSim	AraSimLite	AraSimLite2
Models earth absorption			
Accounts for signal spreading loss			
Evaluates single frequency			
Accounts for neutrino travel direction and Cherenkov Cone			
Models ice characteristics			
Accounts for noise			
Accounts for signal polarization			
Evaluates full frequency range: $f = 83.3\ MHz$ - $1083.5\ MHz$			

Note that the radiation path bending due to varying index of refraction in ice as a function of depth is not accounted for in AraSimLite and AraSimLite2. AraSim has computationally intensive ice models that calculate through-ice radiation propagation. Omission of ray-tracing and noise generation are the greatest contributors to increased computational speed in AraSimLite and AraSimLite2.

3.6 **AraSimLite2**

An updated AraSimLite version (AraSimLite2) accounts for neutrino trajectory and the Cherenkov cone. Table 3-1 above shows the included computations of AraSimLite and AraSimLite2 compared to AraSim.

The fitness function described in equation 3-5 is updated to account for the Cherenkov cone. θ_{view} is defined as the angle between the velocity vector $\vec{v}$ and the vector from the neutrino event to the detector, $\vec{R}$. If θ_{view} is within the Cherenkov cone angle, θ_{cone}, and the received power density is greater than r_{th}, the event weight is added to the fitness score. The revised fitness score is

$$fitness\ score = \sum_{i=1}^{N} \begin{cases} \omega_i, & if\ \frac{g(\theta_i)}{R_i^2} > r_{th}\ and\ 55.3^o < \theta_{view} < 56.3^o \\ 0, & otherwise \end{cases} \tag{3-8}$$

where $\theta_{view}\ is$

$$\theta_{view} = \cos^{-1}\left(\frac{\vec{\boldsymbol{v}} \cdot \vec{\boldsymbol{R}}}{|\vec{\boldsymbol{v}}| \cdot |\vec{\boldsymbol{R}}|}\right) \tag{3-9}$$

Glacial ice's index of refraction is approximately 1.76 [30]. Therefore, using equation 1-1, the Cherenkov cone angle in glacial ice is 56.8^o relative to the neutrino's velocity vector $\vec{v}$. The cone has a width of approximately 1^o. That is

$$55.3^o < \theta_{cone} < 56.3^o \tag{3-10}$$

Figure 3-9 illustrates an event within the viewing angle. X, Y, and Z represent the location of the neutrino event in the ice. Events that meet the viewing angle criteria are counted in the fitness score.

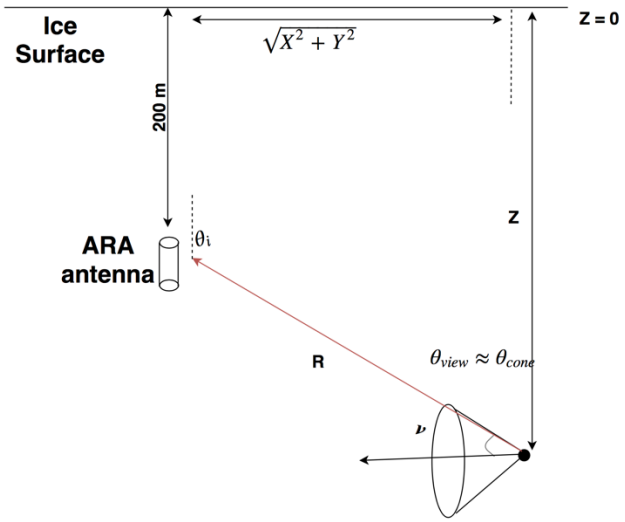

Figure 3-9: Neutrino interaction with θ_{view} meeting viewing angle criteria described in equation 3-8.

The trigger threshold, r_{th}, is re-characterized to account for changes to the neutrino simulation model. Figure 3-10 below shows the fitness score, equation 3-8, as a function of r_{th} for an isotropic radiator. Table 3-2 shows critical threshold values and corresponding fitness scores.

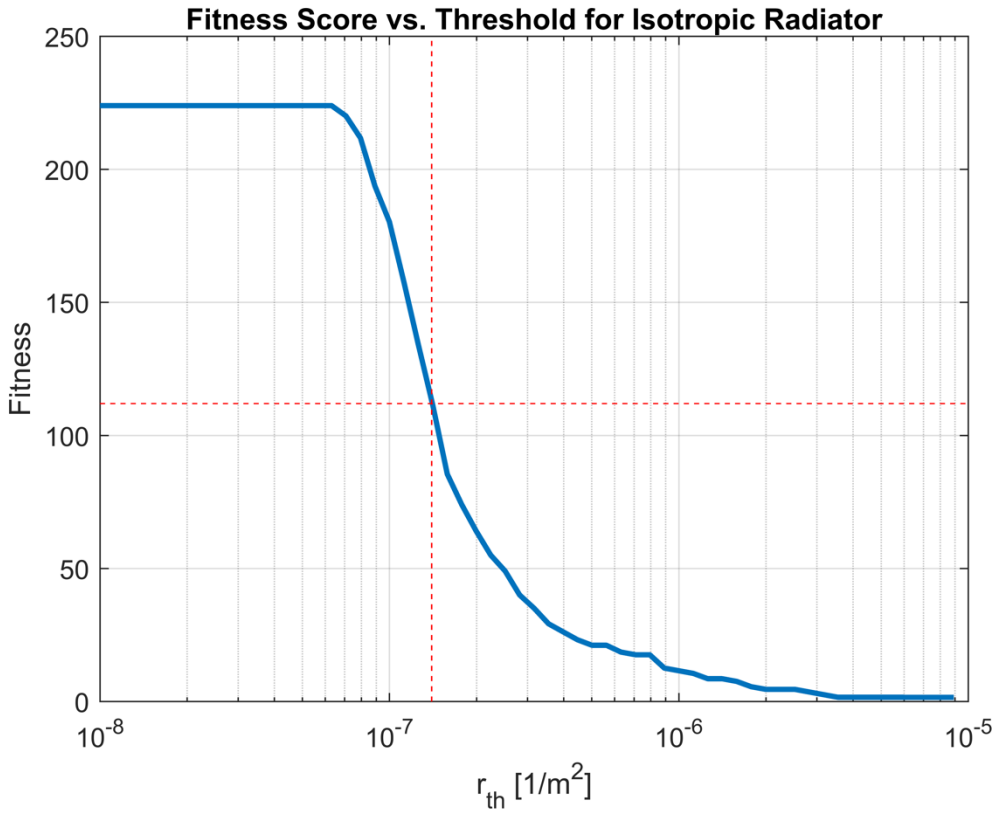

Figure 3-10: AraSimLite2 Fitness score vs. threshold, isotropic radiator. Red dotted line intersection corresponds to 50% fitness point.

Table 3-2: Critical threshold values for AraSimLite2

Percent of Maximum	Fitness	r_{th} [1/m^2]
75%	167	1.2×10^{-7}
50%	112	1.4×10^{-7}
25%	56	2.2×10^{-7}

AraSimLite2 is characterized further to determine the distribution of events that pass the θ_{view} criteria as a function of θ. AraSimLite2 processes antenna radiation patterns using a pre-produced set of events. AraSimLite2 processes antenna radiation patterns from neutrino-ice interaction events computed by full AraSim. N = 100,000 events is selected to ensure the

simulation accounts for a sufficient number of neutrino paths. A histogram of events that pass θ_{view} criteria vs. θ is shown in Figure 3-11. θ is the angle between the z-axis and event location, depicted as θ_i in Figure 3-9. Only 693 events (0.693%) pass the viewing angle criteria, and they cluster around $90^o < \theta < 160^o$. Hence, an optimal radiation pattern's main beam is in this θ range. Figure 3-12 is the Figure 3-11 histogram scaled by weight. The total weight in each angular bin is plotted on the y-axis.

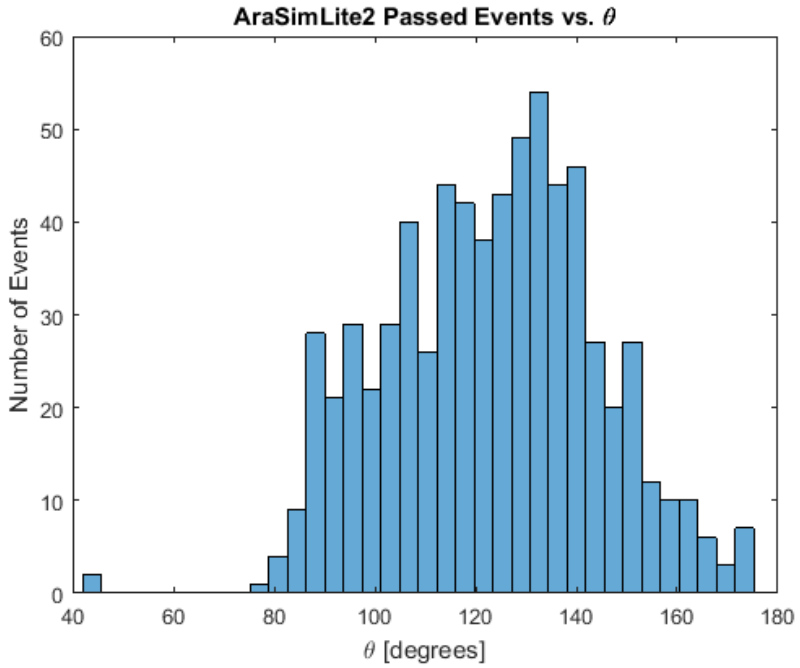

Figure 3-11: AraSimLite2 Passed Events Distribution, Viewing Angle within Cherenkov cone.

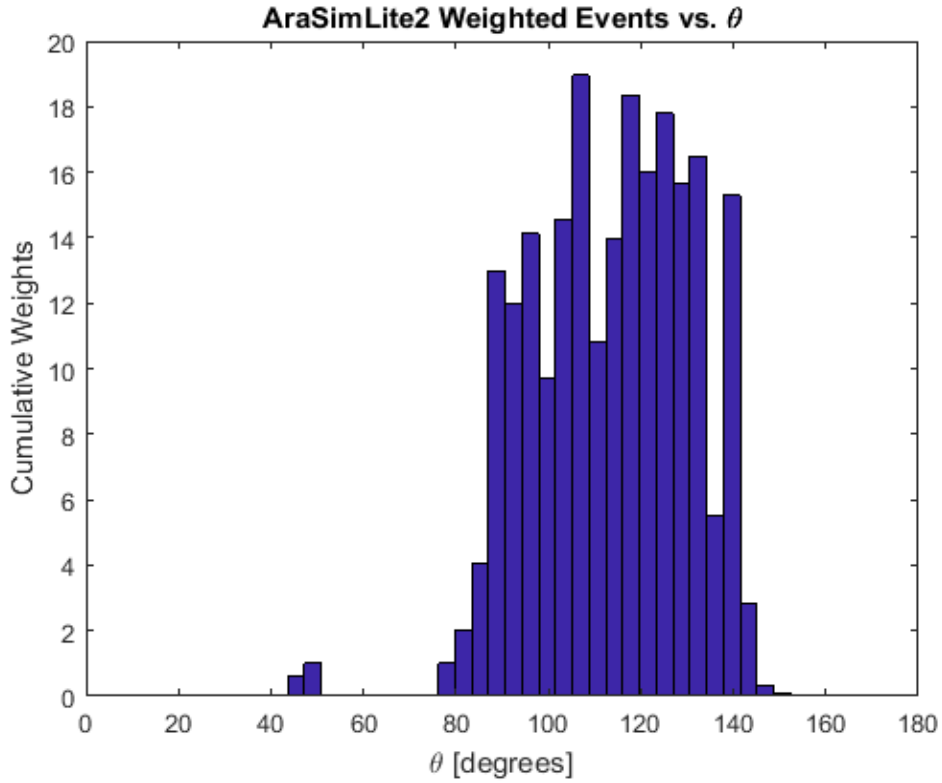

Figure 3-12: AraSimLite2 Passed Event Distribution, scaled by weight.

4 *Radiation Pattern Optimization with Neutrino Simulations*

This section introduces AraSimLite and AraSimLite2 optimization results. The genetic algorithm finds several unique solutions with similar fitness, leading to multiple optimization results. Analyzing a single result is insufficient. The shared features between multiple optimization results inform which radiation pattern characteristics lead to high fitness. Sections 4.1 and 4.2 show each result's radiation pattern, spherical harmonic coefficients, and optimization summary. Section 4.3 analyzes radiation patterns and compares each to the bicone antenna currently implemented in the ARA detector.

4.1 Optimization Results Using AraSimLite

GA2 optimizes radiation patterns through $G(\theta,\vec{x})$ and genotype $\vec{x}$ described in section 2.9.2. Each optimization trial's population size is N=60. The fitness function is

$$f[G(\theta,\vec{x})] = \sum_{i=1}^{N} \begin{cases} \omega_i, & if\ \dfrac{g(\theta_i)}{R_i^2} > r_{th} \\ 0, & otherwise \end{cases} \tag{4-1}$$

and is derived in section 3.5. A selection of three candidate solutions are included.

4.1.1 Candidate Solution 1

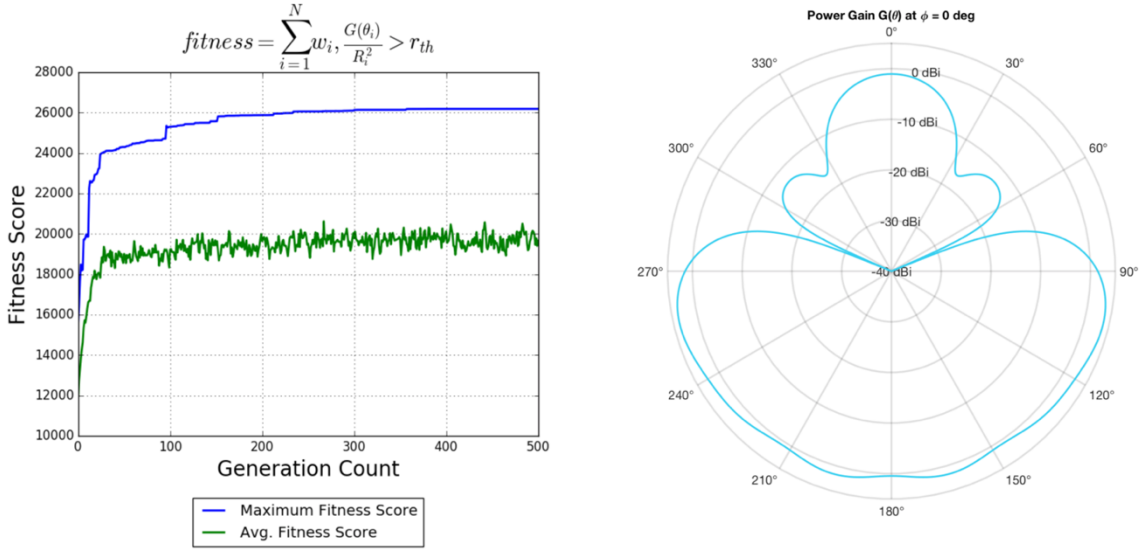

Figure 4-1: Optimization summary (left) and radiation pattern (right) of C1.

Figure 4-1 shows the optimization summary and radiation pattern of candidate solution one (C1). C1 is the fittest individual in the population after 500 generations. The solution vector $\vec{x}$ of C1 is included in Table 4-1. The C1 radiation pattern has maximum gain of 3.81 dBi at 105° and a fitness of 26186.0.

Table 4-1: Spherical harmonic coefficients of candidate solution 1

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-2.275	-0.751	1.491	0.502	-0.825	0.045	0.543	0.105	-0.207	-0.038	0.199	-0.091

4.1.2 Candidate Solution 2

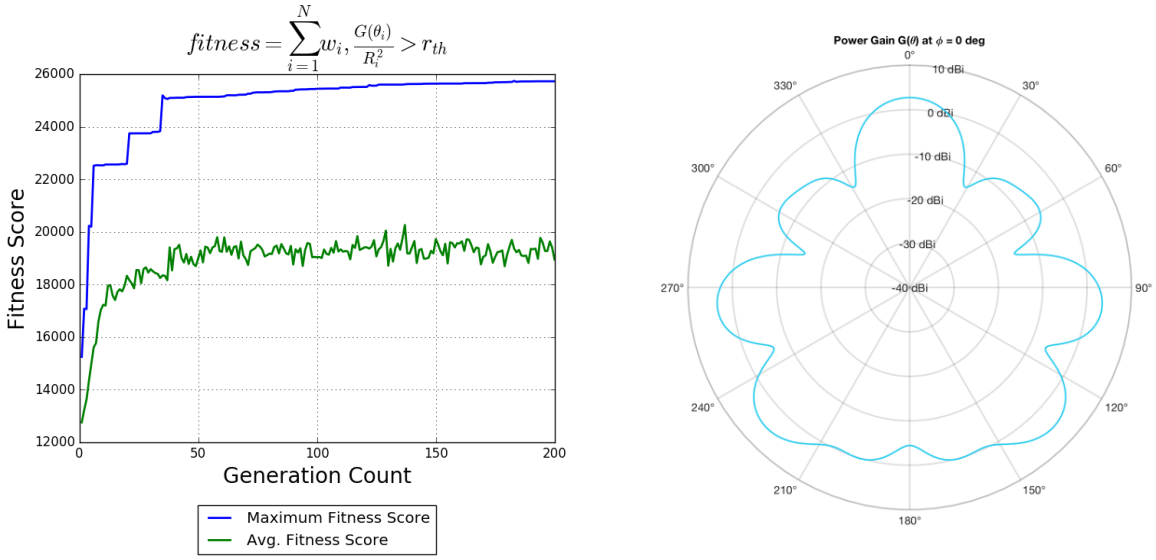

Figure 4-2: Optimization summary (left) and radiation pattern (right) of C2.

Figure 4-2 shows the optimization summary and radiation pattern of candidate solution two (C2). C2 is the fittest individual in the population after 200 generations. The solution vector $\vec{x}$ of C2 is included in Table 4-2. The C2 radiation pattern has maximum gain of 5.13 dBi at 135° and a fitness of 25731.5.

Table 4-2: Genotype of candidate solution 2

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-2.097	-0.124	0.989	-0.002	0.780	-0.977	0.369	1.573	-0.963	-0.247	0.807	-0.228

4.1.3 Candidate Solution 3

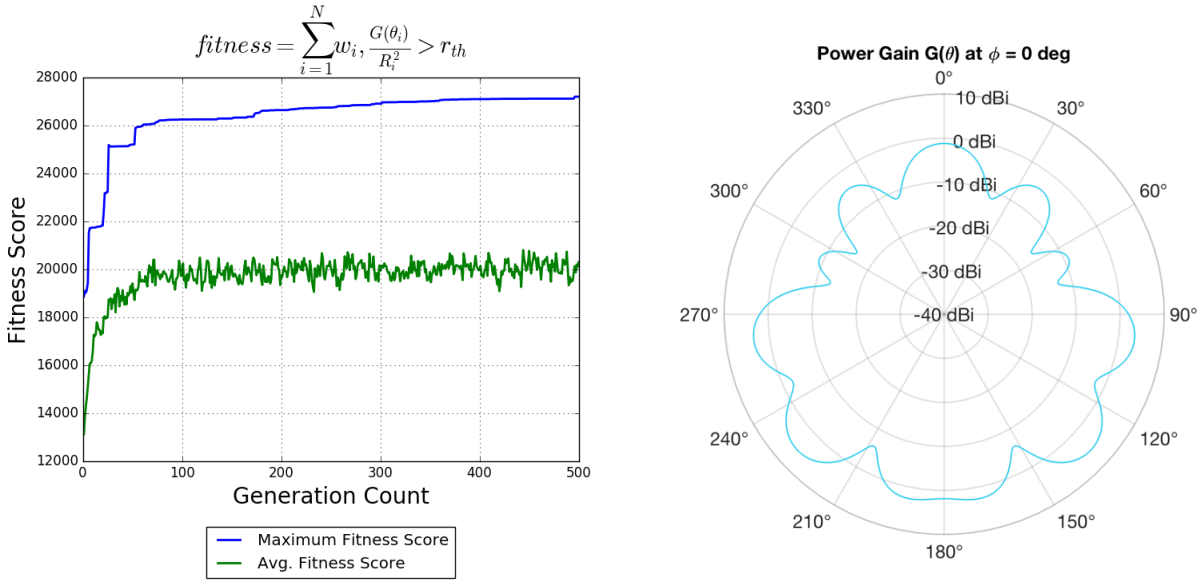

Figure 4-3: Optimization summary (left) and radiation pattern (right) of C3.

Figure 4-3 shows the optimization summary and radiation pattern of candidate solution three (C3). C3 is the fittest individual in the population after 500 generations. The solution vector $\vec{x}$ of C3 is included in Table 4-3. The C3 radiation pattern has maximum gain of 5.42 dBi at 133° and a fitness of 27119.9.

Table 4-3: Genotype of candidate solution 3

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-2.291	-0.287	1.265	-0.210	0.019	-0.497	-0.265	1.556	-1.141	-0.087	1.077	-0.487

4.2 Optimization Results Using AraSimLite2

GA2 optimizes radiation patterns in AraSimLite2. The phenotype $G(\theta, \vec{x})$ and genotype $\vec{x}$ described in section 2.9.2. Each optimization trial size is N=60. The fitness function is,

$$f[G(\theta,\vec{x})] = \sum_{i=1}^{N} \begin{cases} \omega_i, & if\ \frac{g(\theta_i)}{R_i^2} > r_{th}\ and\ 55.3^o < \theta_{view} < 56.3^o \\ 0, & otherwise \end{cases} \quad (4\text{-}2)$$

and is derived in section 3.6. The genetic algorithm finds several solutions over different optimization trials. A selection of three candidate solutions are included.

4.2.1 Candidate Solution 4

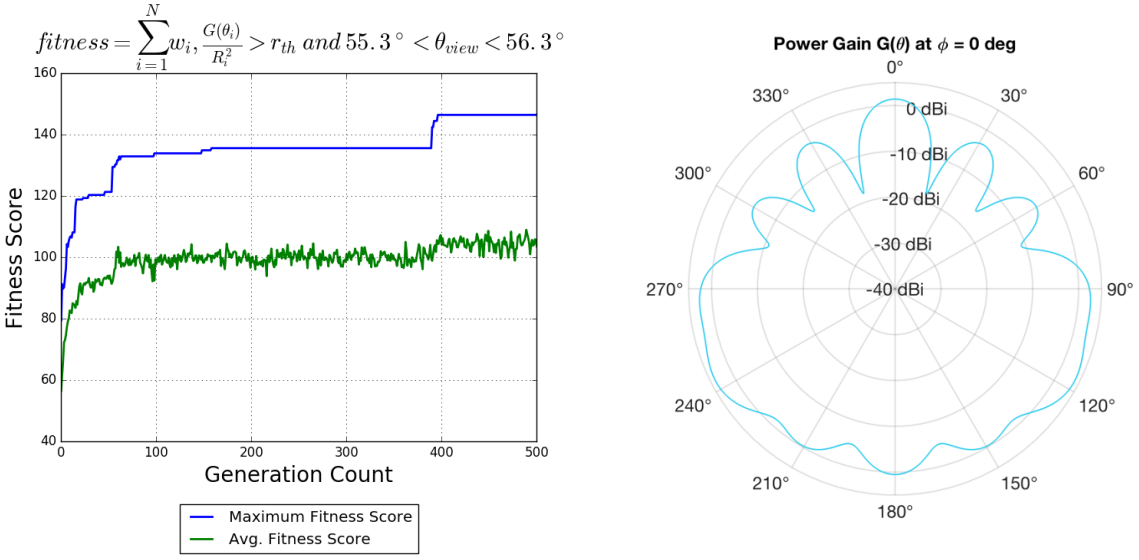

Figure 4-4: Optimization summary (left) and radiation pattern (right) of C4.

Figure 4-4 shows the optimization summary and radiation pattern of candidate solution four (C4). C4 is the fittest individual in the population after 500 generations. The solution vector $\vec{x}$ of C4 is included in

Table 4-4. The C4 radiation pattern has maximum gain of 3.98 dBi at 117.3° and a fitness of 146.4.

Table 4-4: Genotype of candidate solution 4

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-1.580	-1.261	1.646	0.237	-0.778	0.186	0.150	0.009	0.226	-0.253	-0.035	0.682

4.2.2 Candidate Solution 5

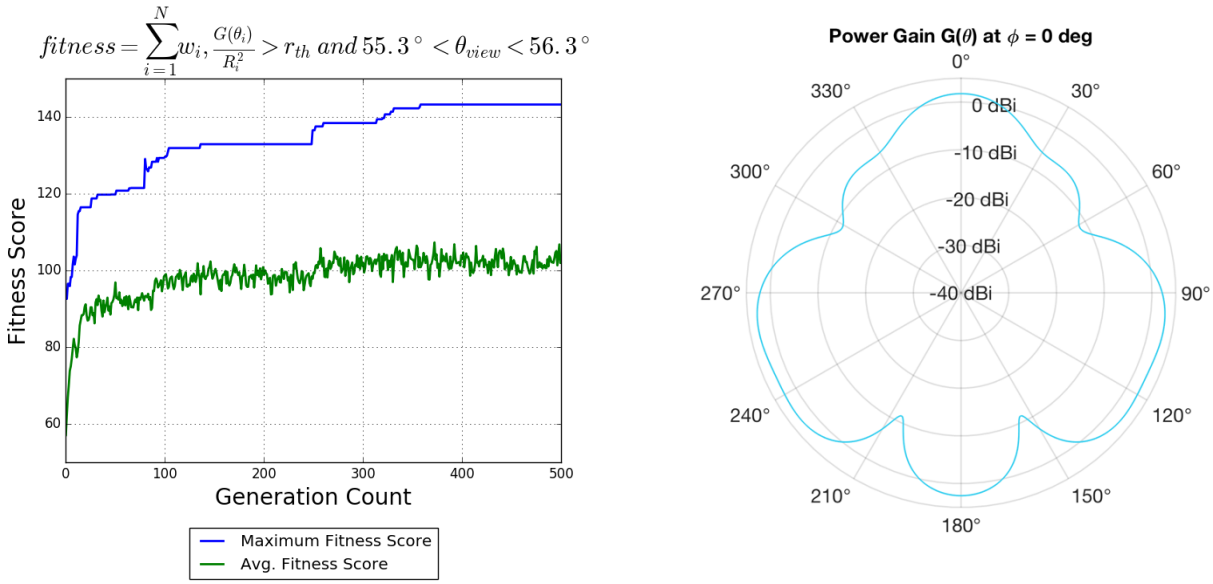

Figure 4-5: Optimization summary (left) and radiation pattern (right) of C5.

Figure 4-5 shows the optimization summary and radiation pattern of candidate solution five (C5). C5 is the fittest individual in the population after 500 generations. The solution vector $\vec{x}$ of C5 is included in Table 4-5. The C5 radiation pattern has maximum gain of 3.12 dBi at 101° and a fitness of 143.2.

Table 4-5: Genotype of candidate solution 5

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-1.472	-1.093	1.708	0.427	-0.441	0.162	-0.125	0.706	-0.261	0.019	0.125	-0.031

4.2.3 *Candidate Solution 6*

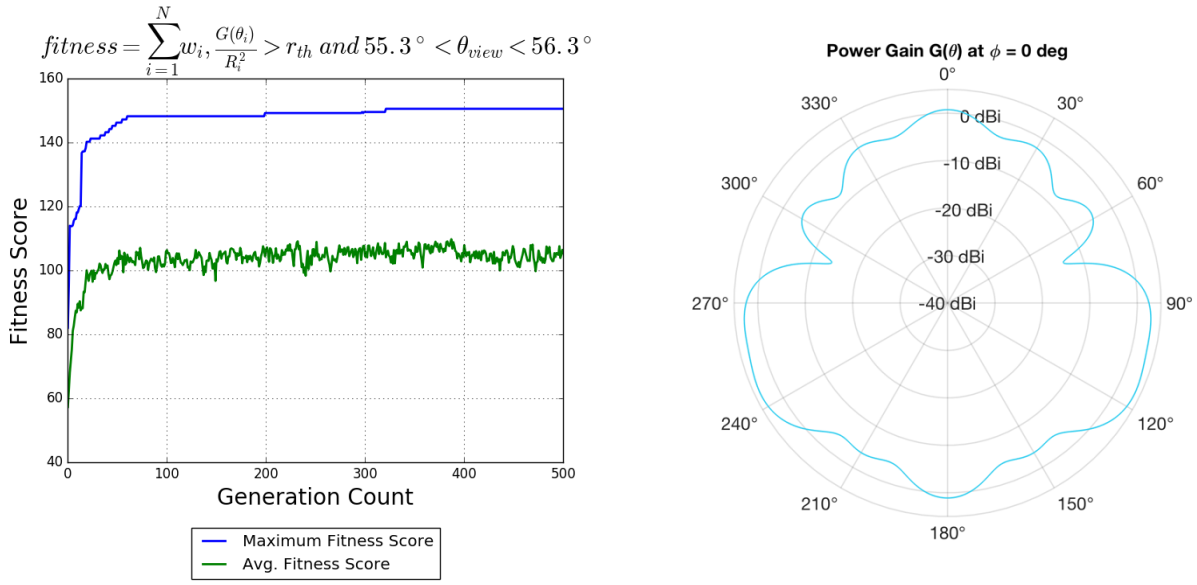

Figure 4-6: Optimization summary (left) and radiation pattern (right) of C6.

Figure 4-6 shows the optimization summary and radiation pattern of candidate solution six (C6). C6 is the fittest individual in the population after 500 generations. The solution vector $\vec{x}$ of C6 is included in Table 4-6. The C6 radiation pattern has maximum gain of 3.82 dBi at 116° and a fitness of 150.5.

Table 4-6: Genotype of candidate solution 6

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
-1.211	-1.327	1.810	0.456	-1.043	0.308	0.108	0.111	0.071	-0.332	-0.030	0.476

4.3 Optimization Results Summary and Comparison

AraSimLite and AraSimLite2 optimization results yield radiation patterns with suppression in the $0° - 90°$ range and gain in the $90° - 180°$ range. C3 is the best candidate from the AraSimLite optimization and has a maximum gain of 5.42 dBi at $133°$. C6 is the best candidate from the AraSimLite2 optimization and has a maximum gain of 3.82 dBi at $116°$.

Each candidate solution from AraSimLite2 is downward directed, with main beam direction $90° < \theta < 120°$. This common feature yields high fitness in AraSimLite2. Table 4-7 shows main beam direction and half power beam width for all candidates.

Table 4-7: Radiation Pattern Summary, Candidates C1-C6

	Origin	Main Beam			Second Side Lobe		
		Direction (degrees)	Half power beam width (degrees)	Gain (dBi)	Direction (degrees)	Half power beam width (degrees)	Gain (dBi)
C1	AraSimLite	104.9	56.0	3.81	N/A	N/A	N/A
C2	AraSimLite	134.4	10.8	5.13	95.9	20.6	3.58
C3	AraSimLite	133.0	10.3	5.42	97.8	20.3	3.64
C4	AraSimLite2	117.3	45.4	3.98	N/A	N/A	N/A
C5	AraSimLite2	101.1	58.2	3.12	180	27.0	2.51
C6	AraSimLite2	116.8	45.2	3.82	180	23.8	1.09

Table 4-8: Fitness Evaluation and Comparison

	AraSim Fitness [km^3sr]	AraSimLite2 Fitness
Bicone	6.23 ± 0.45	91.0
Candidate 6	4.98 ± 0.45	150.5

Table 4-8 compares the best performing candidates to the in-ice bicone antenna. Because AraSimLite and AraSimLite2 are not frequency dependent, the bicone antenna beam pattern at the center of its operating range (400 MHz) is used.

Table 4-8 shows discrepancies between AraSimLite2 and AraSim results. AraSim evaluates the bicone to be 20.1% better than Candidate 6. However, AraSimLite2 evaluates Candidate 6 to have 39.7% better fitness than the bicone. This suggests that AraSimLite2 does not adequately represent AraSim well enough to be optimize an antenna that out performs the bicone. Future work (section 5.2) should implement the GA with AraSim to best optimize an antenna for ARA neutrino detection.

5 Conclusions

5.1 Antenna Recommendations

The ARA application calls for wideband antennas operating in the 150-850 MHz frequency range [1]. A discone antenna is low-gain with 10:1 bandwidth [31]; well-suited for the ARA application. An XFdtd ARA frequency band discone model is shown in Figure 5-1 The discone has dipole radiation characteristics at 250-500 MHz and increases directivity from 0 dBi to 5 dBi in the 90° - 180° range as frequency increases to 1000 MHz, see Figure 5-2. The standing-wave ratio is less than 2.0 across the 220-1000 MHz band, see Figure 5-3.

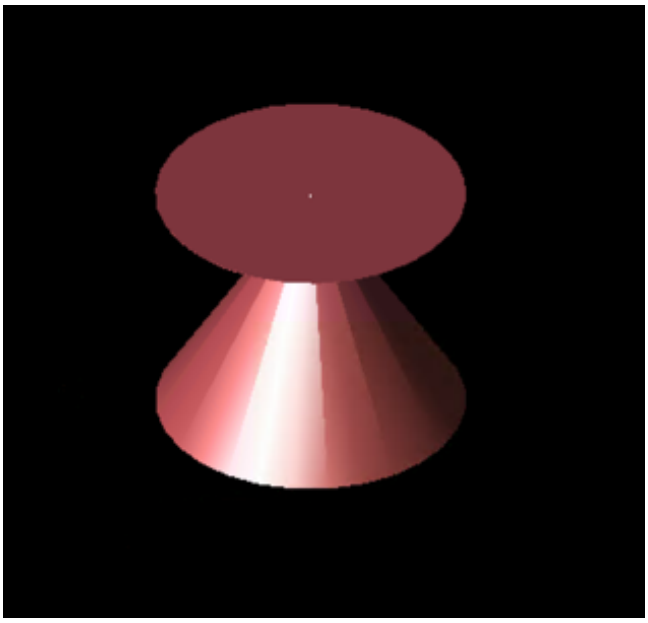

Figure 5-1: Discone model constructed in XFdtd.

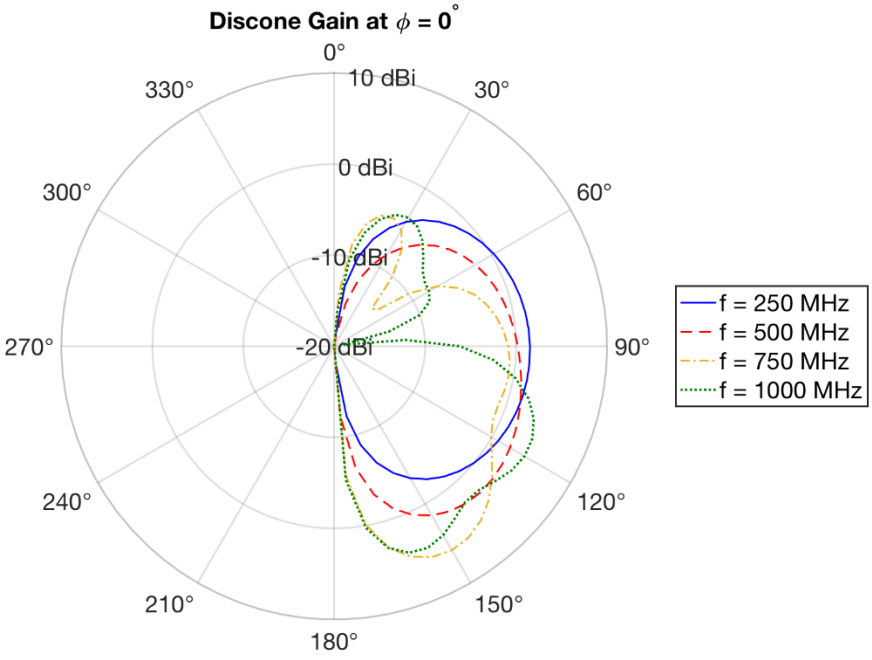

Figure 5-2: XFdtd Discone Gain, 250-1000 MHz

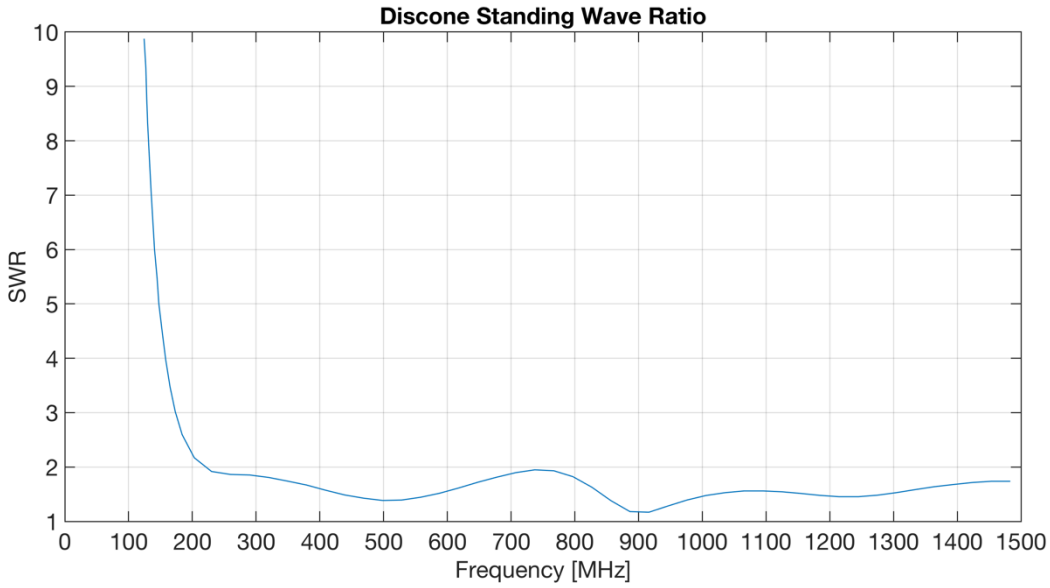

Figure 5-3: XFdtd Discone Standing Wave Ratio vs. Frequency

The discone antenna meets ARA project frequency requirements and the radiation pattern is directed between 90° and 180° for frequencies greater than 500 MHz. The discone diameter is 40 cm, which exceeds ARA's 10 cm diameter borehole.

5.2 Future Work

5.2.1 AraSim Integration

Excessive simulation times (30 minutes per radiation pattern) prevented AraSim objective function optimization. Genetic algorithm and AraSim implementation on a high-performance computing cluster allows parallelization. AraSim objective function optimization yields candidate solutions with all ARA parameters (Table 3-1).

5.3 Concluding Remarks

A genetic algorithm that optimizes radiation patterns was developed. It is interfaced with neutrino simulation tools AraSimLite and AraSimLite2 to produce candidate radiation patterns for neutrino detection. The candidate solutions focus radiation in the $90° < \theta < 180°$ range,

suggesting that downward directed antennas should be used in the ARA experiment. ARA antennas must be broadband; hence, a discone antenna is recommended. However, further analysis is required to develop a discone antenna that meets ARA frequency band and size constraints.

Appendix A: Spherical Harmonic Representations of Antenna Power Patterns

An antenna power pattern $G(\theta, \phi)$ is modelled as the weighted sum of spherical harmonics. $Y_l^m(\theta, \phi)$ is the l, m-order complex spherical harmonic. Under the condition that $m = 0$, spherical harmonics $Y_l^0(\theta, \phi)$ are real and ϕ-independent, or azimuthally symmetric. An azimuthally symmetric power pattern can be written

$$G(\theta) = a_0 Y_0^0(\theta) + a_1 Y_1^0(\theta) + \cdots a_n Y_n^0(\theta), \qquad \theta \in [0, \pi] \tag{A-1}$$

where $\vec{a} = [a_0, a_1, \ldots, a_n]^T$ is the expansion vector and each element of $\vec{a}$ is an expansion coefficient. Antenna power patterns are subject to conservation of energy, placing constraints on expansion coefficients. An energy conserving power pattern should have non-negative gain and obey the relation

$$\Delta\Omega = \frac{4\pi}{D_{max}} \tag{A-2}$$

where $\Delta\Omega$ is the beam solid angle and D_{max} is maximum directivity of the power pattern $G(\theta)$ [31]. The beam solid angle of a power pattern is defined

$$\Delta\Omega = \int_{\Omega} g(\theta, \phi)\, d\Omega = \frac{1}{\max\left[G(\theta, \phi)\right]} \int_{\Omega} G(\theta, \phi)\, d\Omega \tag{A-3}$$

and the maximum directivity is defined

$$D_{max} = \frac{\max\left[G(\theta, \phi)\right]}{\epsilon_r}, \qquad 0 \leq \epsilon_r \leq 1 \tag{A-4}$$

where ϵ_r is the radiation efficiency of the antenna.

For an azimuthally symmetric radiation pattern represented by spherical harmonics

$$\Delta\Omega = \frac{1}{\max[G(\theta)]}\int_{\Omega} G(\theta)\, d\Omega = \frac{1}{\epsilon_r D_{max}}\int_{\Omega}[a_0 Y_0^0(\theta) + a_1 Y_1^0(\theta) + \cdots a_n Y_n^0(\theta)]d\Omega$$

$$= \frac{1}{\epsilon_r D_{max}}\left[\int_{\Omega} a_0 Y_0^0(\theta)\, d\Omega + \int_{\Omega} a_1 Y_1^0(\theta)\, d\Omega + \cdots + \int_{\Omega} a_n Y_n^0(\theta)\, d\Omega\right]$$

$$= \frac{1}{\epsilon_r D_{max}}\left[a_0\int_{\Omega} Y_0^0(\theta)\, d\Omega + a_1\int_{\Omega} Y_1^0(\theta)\, d\Omega + \cdots + a_n\int_{\Omega} Y_n^0(\theta)\, d\Omega\right], \qquad \text{(A-5)}$$

and letting,

$$s_n = \int_{\Omega} Y_n^0(\theta)\, d\Omega, \qquad \vec{s} = [s_0, s_1, \ldots, s_n]^T. \qquad \text{(A-6)}$$

Equations A-5 and A-6 lead to

$$\Delta\Omega = \frac{1}{\epsilon_r D_{max}}[a_0 s_0 + a_1 s_1 + \cdots + a_n s_n] = \frac{1}{\epsilon_r D_{max}}\vec{a}^T\vec{s}. \qquad \text{(A-7)}$$

The vector $\vec{s}$ is evaluated numerically as,

$$\vec{s} = \left[2\sqrt{\pi}, 0, \ldots, 0\right]^T$$

leading to,

$$\Delta\Omega = \frac{1}{\epsilon_r D_{max}}\vec{a}^T\vec{s} = \frac{a_0 s_0}{\epsilon_r D_{max}} \qquad \text{(A-8)}$$

Enforcing equation A-2 on equation A-8,

$$\Delta\Omega = \frac{a_0 s_0}{\epsilon_r D_{max}} = \frac{4\pi}{D_{max}}$$

and solving for a_0,

$$a_0 = \frac{4\pi\epsilon_r}{s_0} = \frac{4\pi\epsilon_r}{2\sqrt{\pi}} = 2\sqrt{\pi}\epsilon_r \cong 2\sqrt{\pi}$$

Because ϵ_r is bounded on [0, 1], realistic antenna power pattern must have $a_0 < 2\sqrt{\pi}$ and non-negative gain. For a well-designed antenna, $\epsilon_r \cong 1$, leading to the constraints

$$G(\theta) = a_0 Y_0^0(\theta) + a_1 Y_1^0(\theta) + \cdots a_n Y_n^0(\theta) \geq 0\,, \qquad \theta \in [0, \pi]$$

and

$$a_0 = 2\sqrt{\pi}$$

If an $\vec{a}$ leads to a negative $G(\theta)$, a new vector $\vec{a}'$ is made to preserve the features of the original vector $\vec{a}$ while obeying the above constraints. Under the condition $\min[G(\theta)] < 0$, an offset can be introduced to $G(\theta)$ by subtracting $\min[G(\theta)]$, scaled by $2\sqrt{\pi}$ from the a_0 element

$$a_0' = a_0 - 2\sqrt{\pi}\min[G(\theta)]. \tag{A-9}$$

This operation removes all negative values from the power pattern, but no longer obeys conservation of energy if $a_0 > 2\sqrt{\pi}$. That is, the power radiated by an antenna with this pattern would be greater than the power input to the antenna. To preserve the characteristics and positivity of the power pattern, each coefficient should be scaled by a positive constant C that restores a_o to $2\sqrt{\pi}$.

$$a_0'' = Ca_0' = C\left(a_0 - 2\sqrt{\pi}\min[G(\theta)]\right) = 2\sqrt{\pi} \tag{A-10}$$

Solving for C,

$$C = \frac{2\sqrt{\pi}}{a_0 - 2\sqrt{\pi}\min\left[G(\theta)\right]}, \tag{A-11}$$

and if $a_0 = 2\sqrt{\pi}$

$$C = \frac{1}{1 - \min\left[G(\theta)\right]}. \tag{A-12}$$

$\vec{a}'$ is then,

$$\vec{a}' = \frac{1}{1 - \min[G(\theta)]}\begin{bmatrix} a_0 - 2\sqrt{\pi}\min[G(\theta)] \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_0 \\ \dfrac{a_1}{1 - \min\left[G(\theta)\right]} \\ \vdots \\ \dfrac{a_n}{1 - \min\left[G(\theta)\right]} \end{bmatrix} \tag{A-13}$$

In the context of optimization, it is beneficial to reduce the dimensionality of the optimization problem, decreasing the size of the parameter space. Because the radiation efficiency $\epsilon_r \cong 1$, it is a valid assumption to set $a_0 = 2\sqrt{\pi}$. Thus, a power pattern becomes

$$G(\theta) = 2\sqrt{\pi}Y_0^0(\theta) + \ a_1 Y_1^0(\theta) + \cdots \ a_n Y_n^0(\theta)\,, \qquad \theta \in [0, \pi] \tag{A-14}$$

and $\vec{a} = [a_1, a_2, \ldots, a_n]^T$. If A results in $G(\theta) < 0$ on $\theta \in [0, \pi]$, the following transformation will remove any negativity while preserving the characteristics of the power pattern

$$\vec{a}' = \frac{1}{1 - \min\left[G(\theta)\right]}\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} \tag{A-15}$$

Works Cited

[1] P. Allison *et al.*, "Performance of two Askaryan Radio Array stations and first results in the search for ultra-high energy neutrinos," *Physical Review D,* vol. 93, no. 8, 2016.

[2] P. Allison *et al.*, "Design and Initial Performance of the Askaryan Radio Array Prototype EeV Neutrino Detector at the South Pole," *Astroparticle Physics,* vol. 35, no. 7, 2012.

[3] A. Markus and H. Francis, "Opening a New Window onto the Universe with IceCube," In press *Progress in Particle and Nuclear Physics,* arXiv: 1805.11112, 2018.

[4] J. Alvarez-Muñiz, "Ultra-high energy neutrinos: status and prospects," in *35th International Cosmic Ray Confrence*, Bexco, 2017.

[5] P. Meszaros, "Astrophysical Sources of High Energy Neutrinos in the IceCube Era," *Annual Review of Nuclear and Particle Science,* vol. 67, pp. 45-67, 2017.

[6] A.-M. J. and Z. E., "Cherenkov radio pulses from EeV neutrino interactions: the LPM effect," *Physics Letters B,* vol. 411, no. 1-2, 1997.

[7] E. Hong, "Searching for Ultra-high Energy Neutrinos with Data from a Prototype Station of the Askaryan Radio Array," Ph.D. dissertation, The Ohio State University, 2014.

[8] S. R. Klein, "Recent Highlights from IceCube," in *33rd International Cosmic Ray Conference*, Rio de Janeiro, 2013.

[9] M. Aaboud *et al.*, "Measurement of the Inelastic Proton-Proton Cross Section at $\sqrt{s} = 13$ TeV with the ATLAS Detector at the LHC," *Physical Review Letters,* vol. 117, no. 18, 27 June 2016.

[10] M. Aartsen *et al.*, "Probing the origin of cosmic-rays with extremely high energy neutrinos using the IceCube Observatory," *Physical Review D,* vol. 88, no. 11, 2013.

[11] A. Connolly and A. Vieregg, "Radio Detection of High Energy Neutrinos," *World Scientific Review,* 2016.

[12] C. Deaconu *et al.*, "Measurements and Modeling of Near-Surface Radio Propagation in Glacial Ice and Implications for Neutrino Experiments," Submitted to *Physical Review D,* arXiv: 1805.12576, 31 May 2018.

[13] R. Fletcher, Practical Methods of Optimization, New York: John Wily & Sons, Inc., 1987.

[14] E. Chong and Z. Stanislaw, An Introduction to Optimization, New York: John Wiley & Sons, Inc., 1996.

[15] J. Snyman, Practical Matehmatical Optimization, An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Alorithms, New York: Springer Science + Business Media, Inc., 2005.

[16] B. Bunday, Basic Optimisation Methods, Victoria: Edwin Arnold (Publishers) Ltd, 1984.

[17] A. Conn, K. Scheinberg and L. Vicente, Introduction to Derivative Free Optimization, Philidelphia: Society for Industrial and Applied Mathematics, 2009.

[18] R. Carr, "Simulated Annealing," MathWorld--A Wolfram Web Resource, [Online]. Available: http://mathworld.wolfram.com/SimulatedAnnealing.html.

[19] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," IEEE, 1995.

[20] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing," *Science,* vol. 220, no. 4598, pp. 671-680, 1893.

[21] R. Haupt and S. Haupt, Practical Genetic Algorithms, New York: John Wiley & Sons, Inc., 1998.

[22] G. S. Hornby, A. Globus, D. S. Linden and J. D. Lohn, "Automated Antenna Design with Evolutionary Algorithms," *American Institute of Aeronautics and Astronautics,* 2006.

[23] L. Davis, Handbook of Genetic Algorithms, New York: Van Nostrand Reinhold, 1991.

[24] R. Hassan, B. Cohanim, O. de Weck and G. Venter, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," American Institute of Aeronautics and Astronautics, 2004.

[25] B. Miller and D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise," *Complex Systems,* vol. 9, pp. 193-212, 1995.

[26] Tutorials Point (I) Pvt. Ltd., "Genetic Algorithms - Parent Selection," 2016. [Online]. Available: https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_tutorial.pdf. [Accessed 6 April 2018].

[27] D. E. Goldberg and K. Deb, "A Comparative Analysis of Selection Schemes Used in Genetic Algorithms," in *Foundations of Genetic Algorithms*, San Mateo, Morgan Kaufmann Publishers, 1991, pp. 70-92.

[28] E. Hong, A. Connolly and C. Pfendner, "Simulation of the ARA Experiment of the Detection of Ultrahigh Energy Neutrinos," in *33rd International Cosmic Ray Conference*, Rio De Janeiro, 2013.

[29] A. Connoly, R. S. Thorne and D. Waters, "Calculation of high energy neutrino-nucleon cross sections and uncertainties using the Martin-Stirling-Thorne-Watt parton distribution functions and implications for future experiments," *Phys. Rev. D,* vol. 83, no. 11, p. 113009, 2011.

[30] I. Kravchenko, B. David and M. Josh, "In situ index-of-refraction measurements of the South Polar firn with the RICE detector," *Journal of Glaciology,* vol. 50, no. 171, 2004.

[31] W. L. Stutzman and G. A. Thiele, Antenna Theory and Design, Third Edition, New York: John Wiley & Sons, Inc, 2013.

[32] K. Bernlohr, "Simulation of imaging atmospheric Cherenkov telescopes with CORSIKA and sim_telarray," *Astroparticle Physics,* vol. 30, no. 3, 2008.

[33] D. Saltzberg *et al.*, "Observation of the Askaryan Effect: Coherent Microwave Cherenkov Emission from Charge Asymmetry in High Energy Particle Cascades," *Physical Review Letters,* vol. 86, no. 13, 2008.