IPN Progress Report 42-246 • August 15, 2026

Antenna Array Geometry Optimization Using a Genetic Algorithm

Katherine G. Skocelas,[*] Jacob Weiler,[†] Amy Connolly,[†] Max Foreback,[*] Emily Dolson,[*] Aman Hafez,[‡] Charles Ofria,[*] Anselmo C. Pontes,[§] Rajiv Ramnath,[‡] Bryan Reynolds,[†] Joey Wagner,[*] and Julie Rolla.[¶]

ABSTRACT. — Antenna array geometry optimization is a design problem in which the physical arrangement of antennas must be defined to meet specified electromagnetic performance objectives. This report describes the application of the Nebulous evolutionary software platform to antenna array geometry optimization. Candidate array designs are encoded using real-valued geometric parameters, simulated in XFdtd to determine their radiation patterns, and evaluated by applying selected fitness functions to those simulated patterns. Two optimization studies are presented for arrays evaluated at a single frequency of 428 MHz. First, a Euclidean distance-based fitness function is used to evolve nine-element dipole arrays toward a prescribed complex radiation pattern. Second, a directional gain fitness function is used to evolve an array composed of nine Yagi-Uda antennas to maximize gain in a target direction while reducing gain in off-target directions. These studies extend previous evolutionary antenna-design work from individual antenna geometries to array layouts and provide a basis for future work in which array designs are optimized directly against mission-relevant science performance metrics.

---

[*]Department of Computer Science and Engineering, Michigan State University.

[†]Department of Physics, Center for Cosmology and AstroParticle Physics, The Ohio State University.

[‡]Department of Computer Science and Engineering, The Ohio State University.

[§]Autogenetics Research Lab.

[¶]Tracking System and Applications Section.

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

I. Introduction

Antenna array geometry optimization is the problem of defining a physically realizable layout of antennas such that the resulting array exhibits the desired electromagnetic performance when detecting signals. This problem is especially important for astrophysics experiments because antenna arrays are often used to detect weak signals in noisy, difficult to access environments.

This report presents the application of the Nebulous evolutionary software platform to antenna array geometry optimization. Two classes of array optimization experiments are presented. In the first, Nebulous evolves an array in an attempt to match a target radiation pattern. In the second, Nebulous evolves an array using a directional fitness function that rewards high gain in the user-specified target direction while penalizing gain in off-target directions. Both classes of experiments evaluate candidate array designs using XFdtd, an electromagnetic simulation software developed by Remcom Inc. [1].

This work expands upon the previous August 2025 Interplanetary Network Progress Report "Designing Optimized Antennas for Science Applications Using Evolutionary Algorithms" [2]. The 2025 report extended earlier primitive-based antenna evolution work [3] by optimizing dipole-like antenna designs according to performance metrics from science simulation software. The present work expands those efforts from individual antenna design to array design by evolving the physical placement of identical copies of a simple dipole antenna. These arrays are evaluated according to their similarity to predefined target gain and phase patterns.

In the second experiment, Nebulous evolves a Yagi-Uda antenna array using a directional gain fitness function to optimize its geometry. Candidate array designs are evaluated for sensitivity in a specified target direction; the fitness function rewards high gain in the target direction and penalizes gain sampled from off-target directions, encouraging arrays with stronger directional response and reduced sensitivity in undesirable spatial regions. Future versions of this approach will integrate science simulation software so that array designs can be optimized directly against mission-relevant science objectives.

Evolutionary optimization methods have been widely applied to antenna array design, including genetic algorithms (GAs), particle swarm optimization, differential evolution, and ant colony optimization [4]. Prior studies have used GAs to optimize antenna radiation pattern characteristics such as main-beam direction, null placement, sidelobe level, half-power beamwidth, and element spacing [5, 6, 7, 8, 9]. GAs have also been used to optimize antenna geometry and array element placement, including Yagi-Uda antenna design and radio astronomy array layouts [10, 11]. More recent evolutionary algorithm (EA)-based array optimization work has examined advanced array optimization strategies, thinned-array problems, encoding strategies, fitness-function design, and constraint handling [12, 13, 14].

Classical antenna-array design is often developed from array-factor analysis, in which the total radiation pattern is represented in terms of the element pattern, element locations, and complex excitation weights [15, 16, 17, 18]. Within this framework, first-principles synthesis methods commonly select element spacing, amplitude taper, or phase excitation to satisfy beam-level requirements such as main-lobe direction, beamwidth, null placement, and sidelobe level. Examples include Dolph-Chebyshev and Taylor distributions for beamwidth and sidelobe control, as well as Woodward-Lawson sampling methods for shaped-beam synthesis [19, 20, 21]. These methods provide analytical guidance and efficient design baselines, but they generally assume prescribed array topologies, simplified element models, or separable control of geometry and excitation. The GA approach used here is complementary: It searches directly over physically realizable array geometries, evaluates each candidate with full-wave electromagnetic simulation, and can incorporate geometric constraints or mission-specific fitness metrics that are difficult to express in closed form. Related JPL work on Deep Space Network array concepts also illustrates that practical array layout involves tradeoffs among gain, sidelobe behavior, baseline length, shadowing, calibration, and cost [22, 23]. The present work therefore uses evolutionary search as an early-stage design-space exploration tool that builds on first-principles array theory while expanding the class of array layouts that can be explored.

The Genetically Evolving Neutrino Telescopes (GENETIS) collaboration has demonstrated that GAs can be combined with science simulation software to design antennas optimized for neutrino detection [24, 25, 26, 27, 28]. The present work builds on this body of research by extending EA-based optimization from individual antenna design to array geometry design.

The intended science applications of Nebulous include antenna arrays for Antarctic ultra-high-energy neutrino observatories, lunar radio instruments for heliophysical studies, and beyond. High-quality array design is critical in these settings, as these systems must measure weak signals in the presence of substantial background noise, are costly to deploy, and are difficult or impossible to modify after installation. Although these applications motivate the present work, the same optimization framework may also be extended to a broader portfolio of radio-observation instruments via the Evolutionary Computation Library for Instrumentation Prototyping in Scientific Engineering (ECLIPSE) [29], a tool developed for Nebulous. ECLIPSE enables Nebulous to apply standard evolutionary workflows to a wide range of applications, such as passive sounding and low-frequency radio astronomy—domains with well-established design challenges ([30, 31, 32, 33]). Using ECLIPSE, Nebulous is able to rapidly extend existing software to new mission concepts while accommodating problem-specific constraints and requirements. By improving early-stage array optimization capability, this work paves the way for future astrophysics experiments across a range of mission classes.

The remainder of this report is organized as follows. Section II describes a detailed overview of the GA, including how array individuals are constructed, simulated, evaluated, and replicated. Section II.B describes the two fitness functions used in this analysis: Euclidean distance and directional gain. The results of evolving array geometries using each fitness function are given in Section III. Finally, Section IV discusses the significance of the results and the next steps in evolving array geometries toward specified science outcomes.

II. Genetic Algorithm

The workflow discussed in this report uses an age-layered population structure (ALPS) [34] to promote continued exploration, avoiding premature convergence to suboptimal solutions. ALPS regularly introduces new random individuals and gives them the ability to compete by restricting competition based on lineage age. In this way, younger lineages are given time to improve before competing with older, more refined solutions. Prior studies have shown that ALPS produces better solutions than other multi-start EAs and diversity-maintenance schemes [34, 35].

A flowchart summarizing the GA workflow used in this analysis is provided in Figure 1; each stage is described in the subsections that follow. The algorithm is initialized by constructing a starting population of candidate antenna array layout designs, each initialized with a lineage age of zero. Individual designs are generated using the procedure detailed in Section II.A and are then modeled within XFdtd. Electromagnetic simulations performed with XFdtd yield the phase and gain response patterns for each individual. These response patterns are then used to derive fitness scores based on the selected goals. These fitness scores are used in the parent selection process, along with other factors such as lineage age, to determine which individuals produce offspring for the next generation. Each offspring is mutated, guaranteeing that it is different from its parent, and has its lineage age incremented. These mutated individuals are added to the population, replacing a random set of individuals from the previous generation with the caveat that the highest fitness individual is always preserved. After every 1000 evaluation steps, new, randomly generated offspring are introduced into the population (starting with a lineage age of zero). A complete list of the GA's configurable parameters is provided in Appendix I.

A. Array Individuals

Array individuals are defined by a genome of four real-valued parameters: the inter-row spacing (d_1), the column offset distance (d_2), the column offset angle (ψ), and a single azimuth angle (α) that is applied to all antennas. In addition to these evolved parameters, a collection of static parameters is set by the user at the beginning of each experiment: the number of nodes (fixed), the minimum ($d_{\min}$) distance between nodes, and the maximum grid rows and columns. An individual's node coordinates are centered on the x-y plane before being passed to XFdtd. An example array individual is shown in Figure 2.

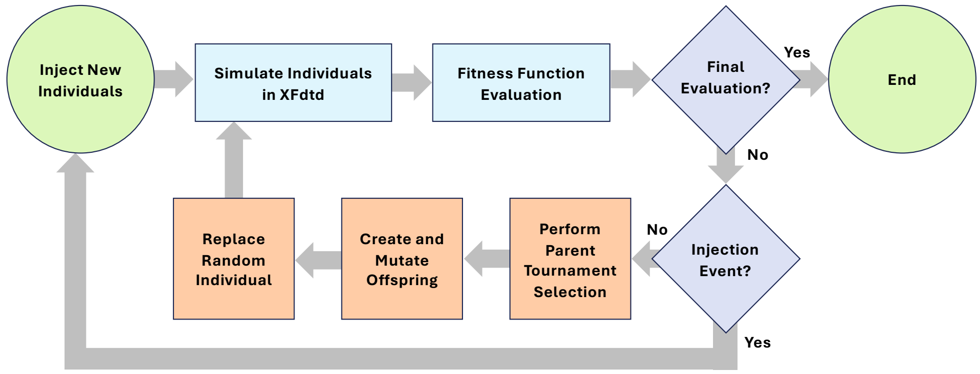
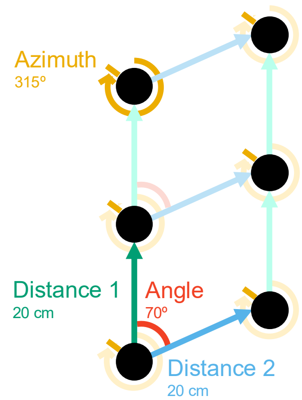


Figure 1. Flowchart of the GA loop used for antenna-array optimization. The loop begins with the injection of new candidate individuals, followed by electromagnetic simulation in XFdtd and fitness-function evaluation. If the final evaluation step has not been reached, the algorithm either injects new individuals at the scheduled injection interval or performs parent tournament selection, offspring creation and mutation, random-individual replacement, and re-simulation. Green circles denote injection and termination stages, blue rectangles represent the fitness function evaluation steps, orange squares correspond to the steps involved in generating new individuals, and lavender diamonds denote conditional decision nodes.

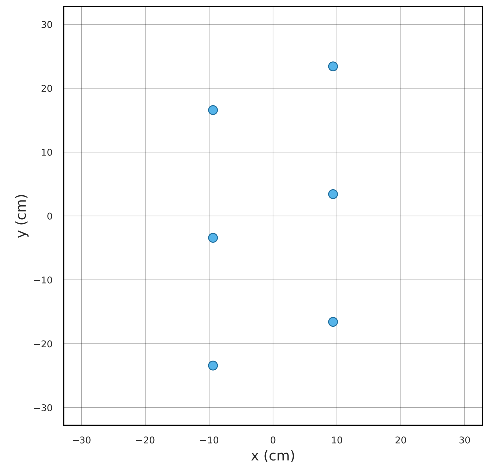
The parameter d_1 defines the candidate spacing in centimeters between successive positions within each column (i.e., the row-to-row spacing along the column direction). The parameter d_2 defines the column offset distance of the candidate grid in centimeters, such that successive columns are offset by $d_2 \sin\psi$ horizontally and $d_2 \cos\psi$ vertically, where ψ is the evolved column offset angle. When $\psi = 90°$, the two evolved spacing parameters (d_1 and d_2) define a perfect grid with columns d_2 centimeters apart, and rows d_1 centimeters apart.

Candidate positions are accepted into the array only if they satisfy a minimum Euclidean separation distance of d_{min} from all previously placed nodes. As a result, d_1 and d_2 do not directly determine the final node positions, but instead define the resolution of the grid searched during placement. Candidate positions are evaluated in column-major order, proceeding from row zero to the maximum row count within each column before advancing to the next, such that earlier-placed nodes may occlude subsequent candidate positions that fall within d_{min}. When moving to the next column, the algorithm starts at the lowest row in range given the offset from the previous column. The spacing parameters d_1 and d_2, and the column offset angle, ψ, are subject to evolutionary pressure, allowing the algorithm to explore a continuous space of candidate array geometries.

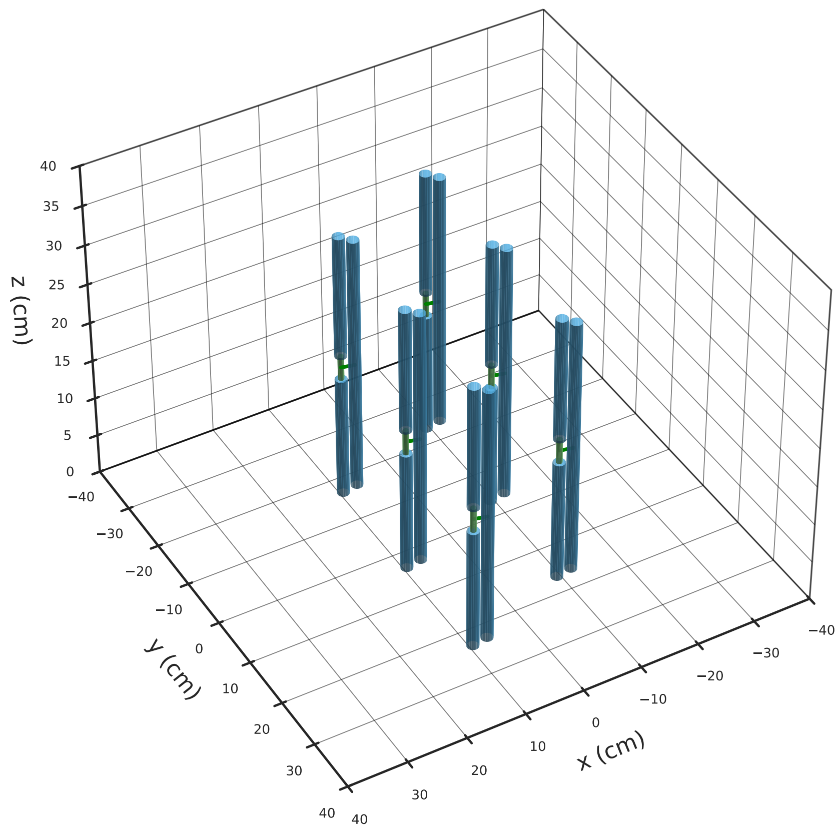
The parameter α defines the azimuth, which is the orientation of the antennas in the array. The azimuth is applied uniformly across all nodes of a given individual.



(a) Genome encoding



(b) Resulting node coordinates



(c) Simulated antenna array

Figure 2. Mapping from an example candidate genome to its simulated antenna array. (a) Each individual's genome encodes four values: two inter-node distances, a column offset angle, and a shared antenna azimuth. The GA also uses three static, user-defined parameters: a fixed number of nodes and maximum numbers of rows and columns. This example uses 6 nodes, 3 rows, and 2 columns. (b) The genome values, together with these node, row, and column limits, determine the array's node coordinates (centered at the origin). (c) The resulting coordinates are combined with a user-defined antenna to produce the three-dimensional antenna array used in the electromagnetic simulation.

B. Fitness Function Calculation

Once constructed, an array individual is simulated in XFdtd and the resulting gain and phase patterns are used to determine the individual's fitness. In this analysis, two fitness functions were utilized for the studies presented in Section III: Euclidean distance, F_E, and directional gain, F_D.

1. Euclidean Distance Fitness Function

The Euclidean distance-based fitness function is used to compare the radiation pattern of an evolved antenna array to that of a target design. A similar Euclidean distance-based fitness function was also discussed in detail in a previous IPNPR [3].

Euclidean distance is used to quantify the agreement between N-dimensional vectors by summing corresponding elements of each vector in quadrature. A Euclidean distance of zero represents exact agreement between vectors, with larger values indicating more dissimilarity. The Euclidean distance between two generalized N-dimensional vectors, n and m, is given by Equation 1:

$$d(\vec{n}, \vec{m}) = \sqrt{\sum_{i=1}^{N} (n_i - m_i)^2} \tag{1}$$

To extend the Euclidean distance calculation to determine similarity between antenna radiation patterns, complex values encoding gain and phase in both the zenith, θ, and azimuthal, ϕ, polarizations are defined as shown in Equations 2 and 3.

$$Z_\theta = G_\theta \left[\cos(\Phi_\theta) + i \sin(\Phi_\theta)\right] \tag{2}$$

$$Z_\phi = G_\phi \left[\cos(\Phi_\phi) + i \sin(\Phi_\phi)\right] \tag{3}$$

Here, G_θ and G_ϕ are the theta-polarized and phi-polarized gain values, respectively, converted from dBi to linear gain, and Φ_θ and Φ_ϕ are the corresponding phases in radians.

The complex values Z_θ and Z_ϕ are evaluated at each sampled point in (θ,ϕ), i, for both the evolved and target antenna patterns. Accounting for both polarizations independently and weighting their contributions, the Euclidean distance-based fitness metric assumes the form shown in Equation 4.

$$F_E = \sqrt{\left(\sum_{i=1}^{N} \left[w_\theta \left|Z_{\theta_i,\text{evolved}} - Z_{\theta_i,\text{target}}\right|^2 + w_\phi \left|Z_{\phi_i,\text{evolved}} - Z_{\phi_i,\text{target}}\right|^2\right]\right)} \tag{4}$$

Contributions from each polarization are assigned weights, w_θ and w_ϕ, which are required to obey $w_\theta + w_\phi = 1$. For all experiments using the Euclidean distance fitness function discussed in this report, equal weighting was assigned to each contribution, yielding the weights $w_\theta = w_\phi = 0.5$.

The Euclidean distance metric given in Equation 4 is used as a fitness function when comparing the performance of an evolved array to that of a target design. Smaller values of the Euclidean distance fitness metric, F_E, indicate stronger agreement between the radiation patterns of the evolved and target designs, with a value of $F_E = 0$ representing perfectly identical performance.

2. Directional Gain Fitness Function

An important consideration for antennas and antenna arrays is their directional sensitivity. To evolve toward specified directional sensitivity goals, a fitness function was created to maximize and/or minimize gain in user-defined directions. This directional gain fitness function calculates the average gain values in the specified angular regions for sensitivity maximization and minimization, and subtracts the values, yielding a fitness score that rewards individuals with larger average gain in the maximization region and penalizes individuals with larger average gain in the minimization region. The directional gain fitness function is given by Equation 5.

$$F_D = \bar{G}_{\max} - \bar{G}_{\min} \tag{5}$$

where:

$$\bar{G}_{\max} = \frac{1}{|M|} \sum_{(\theta,\phi)\in M} 10^{g_\theta(\theta,\phi)/10} \tag{6}$$

$$\bar{G}_{\min} = \frac{1}{|m|} \sum_{(\theta,\phi)\in m} 10^{g_\theta(\theta,\phi)/10} \tag{7}$$

Here, M represents the set of points where gain is evaluated in the maximization region and m represents the set of points where gain is evaluated in the minimization region. The term $g_\theta(\theta, \phi)$ represents the theta-polarized gain in dBi evaluated in the direction of (θ, ϕ) in spherical coordinates, where θ and ϕ are the zenith and azimuth angles, respectively. The values $\bar{G}_{\max}$ and $\bar{G}_{\min}$ therefore represent the average linear gain over sampled directions in the specified maximization and minimization regions, respectively. Better performing individuals will yield larger values of F_D upon evaluation.

Only theta-polarized gain values were used in the fitness calculation for this iteration of the analysis. This simplification is justified by the geometry of the array and the antenna configuration, for which the phi-polarized gain is approximately $-38\,\mathrm{dBi}$, which is effectively negligible. Consequently, computing fitness from the total gain magnitude (combining both polarization components) would yield results equivalent to those obtained using theta polarization alone, and the added complexity is not warranted.

C. Parent Selection

This analysis relied exclusively on tournament selection with ALPS to select parent individuals. In tournament selection, a subset of T individuals compete and the individual with the highest fitness score is selected to be a parent [36, 37, 38, 39]. In ALPS, each tournament is limited to prospective parents within a single age-layer, ensuring that well established lineages do not compete directly against newly injected individuals. This procedure biases reproduction toward higher-performing candidate solutions within each layer.

The tournament size T determines the strength of selection pressure. Larger tournaments increase the probability that high-fitness individuals are part of the selected subset, which can accelerate convergence but may reduce population diversity. Smaller tournaments introduce more stochasticity into parent selection, allowing lower-fitness individuals to occasionally reproduce, which helps preserve diversity.

Preliminary tuning trials with a larger tournament size of $T = 10$ showed rapid convergence of the population, suggesting that the selection pressure was too strong for this setup. The tournament size was therefore reduced to $T = 3$, corresponding to 2% of the 150-individual population, to better preserve population diversity while still favoring higher-fitness individuals.

D. Genetic Operators

This section describes the two genetic operators used to build new array individuals: mutation and injection.

Array individuals reproduce via asexual reproduction, in which each offspring has only one parent. When a parent individual is selected for replication, each of the four genes in its genome has a 30% chance of mutating. When a gene mutates, it is modified by adding zero-mean Gaussian noise with a configurable standard deviation to its current value, then rounding the result to the nearest whole number and clamping it to the allowable range (in the case of inter-row spacing, column offset distance, and column offset angle) or wrapping it (in the case of the circular azimuth value).

Before a new offspring individual is accepted into the population, it must pass two verification steps. First, the algorithm confirms that it is not a clone of its parent. Second, the algorithm verifies that it is a valid array individual satisfying the user's constraints. If either of these steps fails, the offspring individual is discarded and a new one is generated from the same parent. The cycle continues until an offspring individual is generated that is both unique and valid.

New array individuals are also inserted into the population via injection. Injected individuals are created using the same technique as new individuals in the first generation. For each gene, a random value is generated between the minimum and maximum ranges for that gene. The minimum and maximum values for inter-row spacing (d_1) and column offset distance (d_2) are defined by the user; column offset angle (ψ) must be between 1 and 90 degrees, inclusive, and azimuth (α) must be between 0 and 359 degrees, inclusive. A prospective individual is rejected and regenerated if its gene encoding does not result in the user-defined number of nodes.

All accepted new array individuals are incorporated using the asynchronous steady-state evolutionary framework described in our previous work [3]. In this framework, new individuals are inserted into the population immediately after they are generated rather than waiting for a full generational replacement step. For comparison with generational genetic algorithms, one generation is defined as the creation of a number of new individuals equal to the population size.

III. Results and Discussion

For this analysis, the array geometry optimization was tested under two experimental conditions. First, we used the Euclidean distance fitness function to evolve an array of nine dipole antennas to a target radiation pattern. Second, we used the directional fitness function to evolve an array composed of nine Yagi-Uda antennas to maximize gain in a target direction. In both conditions, the array geometries were evolved for a single frequency of 428 MHz and used the same core GA parameters (see Appendix I).

The total number of evaluation steps for each run was not fixed in advance by an automated stopping criterion. The current Nebulous implementation does not include a convergence-based termination condition. Instead, a run ends when it is manually halted or when the compute scheduler reaches its wall-time limit.

After each run, we execute a post-processing script to assess whether the search has converged or should be extended. The script reports the best-performing individual, the evaluation step at which that individual was first evaluated, and the evaluation step at which its genome first appeared in the population. It also reports the best-performing individuals within each ALPS layer and generates a scatter plot of fitness as a function of evaluation step.

These outputs are used to determine whether additional evaluation steps are likely to yield further improvement. A run is extended when the best-performing individual appeared recently, when the leading individuals across ALPS layers still have distinct fitness scores, or when the fitness scatter plot shows continued improvement. A run is not extended when the best-performing individual appeared many evaluation steps earlier with no subsequent improvement, when the leading individuals across ALPS layers converge to the same fitness score, or when there is little room for further improvement. In the Euclidean-distance run (minimizing distance from the target pattern), the best individual (a near-perfect match to the target) was found at evaluation step 3,484 of 3,615 total steps, at which point the run was stopped. In the directional-gain run (maximizing fitness), the best individual was found at evaluation step 5,602 of 7,280 total steps. As the run continued, more individuals converged to the same max fitness score, indicating model convergence and the run was stopped.

Future versions of Nebulous will support automated termination based on user-defined convergence criteria, such as no improvement over a fixed number of evaluations or convergence of the best-performing individuals across ALPS layers.

A. Euclidean Distance Target Radiation Pattern Evolution

To build on our previous work, we first evolved an array of nine dipole antennas to a target radiation pattern. The target radiation pattern was generated by selecting the antenna coordinates of the target array and simulating it with XFdtd. An individual's fitness was calculated via the Euclidean distance fitness function as described in Section II.B.1.

Figure 3 shows the results of evolving a population of 150 individuals evaluated at a single frequency of 428 MHz. Maximum rows and columns were each set to three, and number of antennas was fixed at nine. Each point represents one evaluated individual, plotted according to its global evaluation step and Euclidean distance fitness value. Point color denotes the ALPS age layer to which the individual belongs, and the red star marks the highest-fitness individual identified during the run. After 3,615 evaluations, the best evolved array individual's Euclidean distance from the target radiation pattern was 0.025. By comparison, we evaluated a hill climber algorithm, a local search heuristic that iteratively perturbs a single candidate solution and retains each change only if it improves fitness. Across five replicate runs, the average lowest Euclidean distance achieved by this algorithm was 47.143, and the lowest Euclidean distance reached by any single run was 43.837 (see Appendix II).

Figure 4 shows the 2D node coordinates of the final best-individual geometry from optimization using the Euclidean distance fitness function. Figure 5 shows the gain and phase patterns of the most fit individual at the final evaluation step compared to the overlaid gain and phase patterns of the target array. These plots show that Nebulous was able to evolve a close match to the target array using the Euclidean distance fitness function.

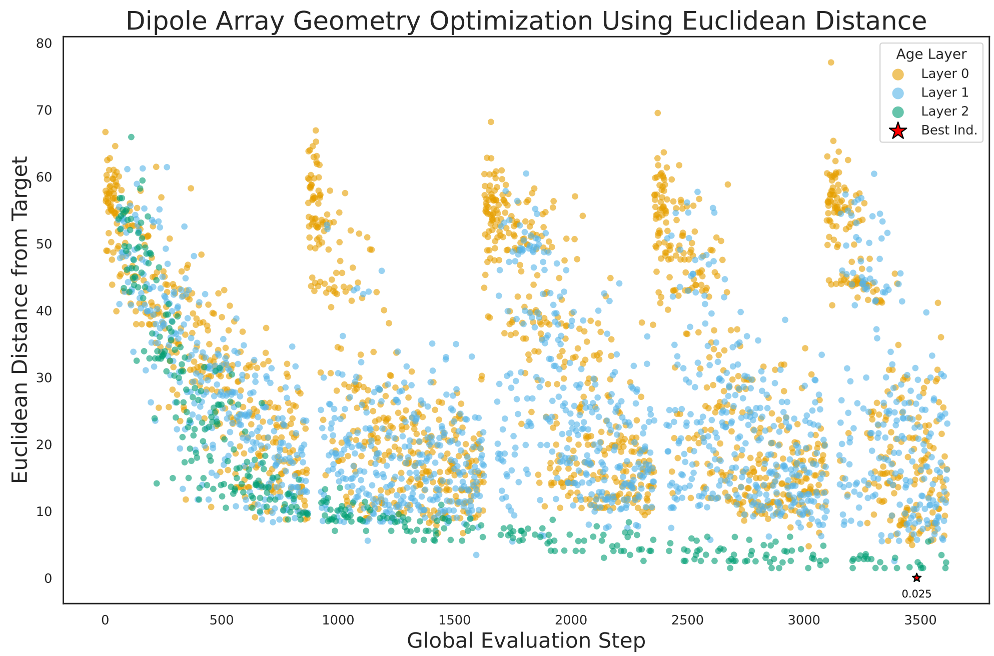


Figure 3. The Euclidean distance from target values obtained during a dipole antenna array optimization run using the Euclidean distance fitness function, maximum three rows and columns, and nine antennas. Each point corresponds to one evaluated individual and is plotted against the global evaluation step. Point color denotes the ALPS age layer to which the individual belongs, and the red star marks the best-matching individual identified during the run. The text below the red star indicates the top individual's Euclidean distance from the target (0.025). Lower distance values indicate closer agreement between the simulated and target single-frequency radiation patterns.

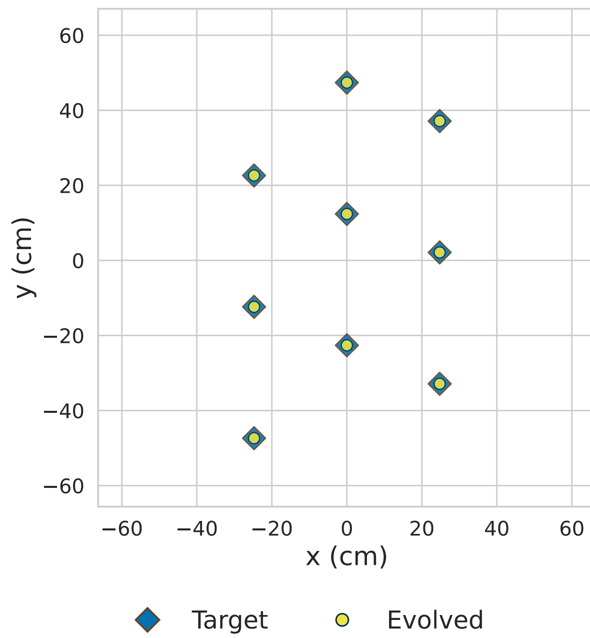


Figure 4. Final best-individual 2D geometry of a dipole array after optimization using the Euclidean distance fitness function, maximum three rows and columns, and nine nodes. Evolved individual node coordinates (yellow circles) are overlaid against the fixed target array geometry (blue diamonds).

B. Directional Gain Fitness Function Results

In our second experiment, we used the directional gain fitness function to optimize the geometry of a nine-element Yagi-Uda array. The directional fitness function maximizes gain in the target direction and minimizes gain in all other directions as described in Section II.B.2. While we did not incorporate phase into this experiment, it can be added as needed in future analyses based on project requirements.

We evolved the geometry of an array of nine Yagi-Uda antennas at a single frequency of 428 MHz toward a broadside target (i.e., a direction perpendicular to the array's x-axis; see Figure 6). Maximum rows and columns were both set to three, and number of nodes was fixed at nine. Gain was maximized within a 30° sector centered at $\theta = 90°$ and $\phi_{\text{target}} = 0°$, with gain suppressed outside that region. After evolving for 7,280 evaluations, the most fit individual achieved a fitness score of 13.886, 2.404 higher than its human-designed counterpart, which used basic design principles and achieved a fitness of 11.482 (see Appendix III).

Figure 7 shows representative slices of the gain pattern of the best Yagi-Uda array individual in the azimuth and zenith directions. These results show that the Nebulous algorithm successfully shaped the antenna array's radiation pattern to favor the target angular region in both cuts. In azimuth, the best-overall individual concentrates gain into the maximization region near 0°, with gain dropping in the minimization region.

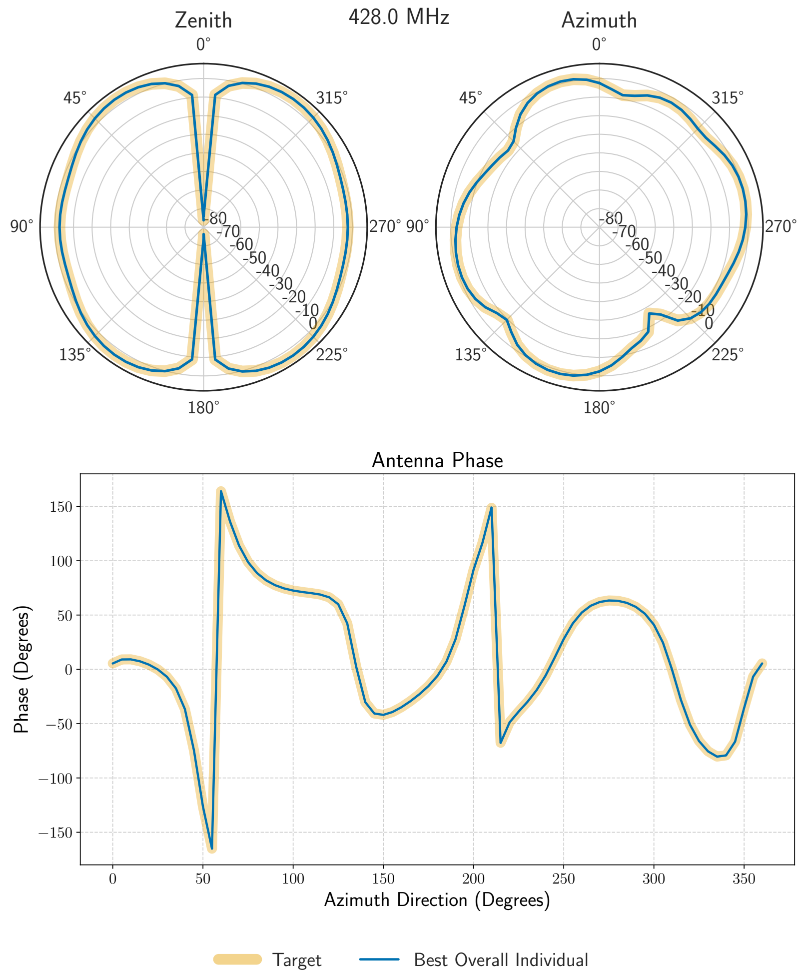


Figure 5. The gain (top) and phase (bottom) patterns of the most fit individual as of the final evaluation step of the single-frequency dipole array evolution toward a target radiation pattern using the Euclidean distance fitness function, maximum three rows and columns, and nine nodes. Also overlaid on each is the corresponding pattern of the target. The target's line is enlarged for visibility.

The zenith cut shows the same desired behavior: Gain is elevated throughout the maximization region centered at 90° and drops off in the surrounding minimization region. Both cuts show jagged, multi-lobed structure outside the target region, indicative of constructive/destructive interference between array elements.

IV. Conclusions

This report presents a GA for antenna array geometry optimization that is capable of evolving to a target radiation pattern or to maximize and minimize sensitivity in user-defined directions. With a Euclidean distance fitness function, the algorithm was able to evolve the geometry of an array of nine dipole antennas to match a target radiation pattern, showing strong agreement with the target in terms of both gain and

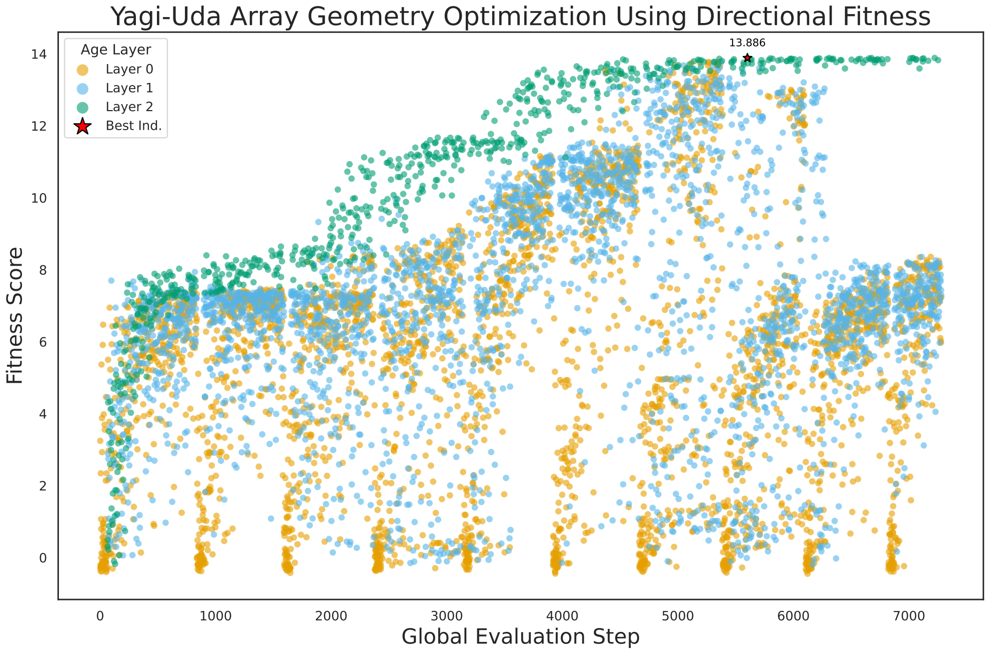


Figure 6. Fitness scores from a Yagi-Uda antenna array geometry optimization run using the directional fitness function, maximum three rows and columns, and nine nodes. Each point corresponds to one evaluated individual, plotted against the global evaluation step. Point color denotes the ALPS age layer to which the individual belongs, and the red star marks the best individual identified during the run, with its fitness score labeled above. Higher fitness scores indicate better directional performance.

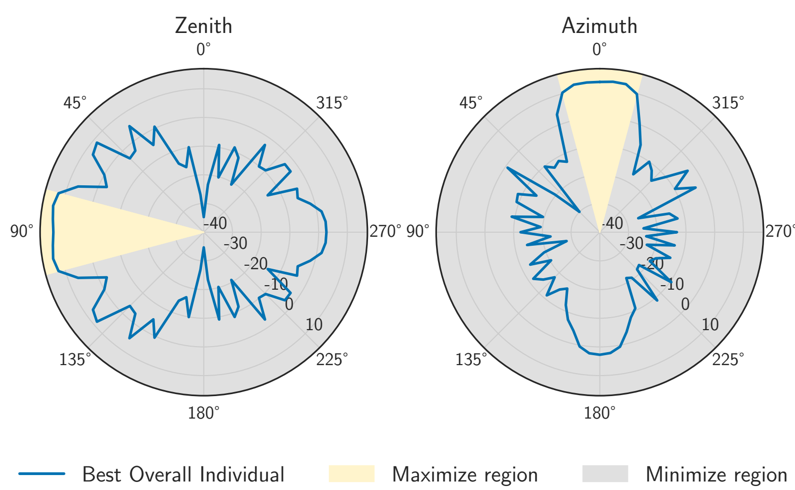


Figure 7. Zenith (left) and azimuth (right) gain patterns for the best-overall individual from the Yagi-Uda antenna array geometry optimization run using the directional fitness function, maximum three rows and columns, and nine nodes. The radial axis shows gain in dBi (gridlines from -40 dBi at center to 20 dBi at the outer edge) as a function of angle. The blue trace is the simulated gain pattern of the best individual; the yellow wedge marks the angular region where gain was rewarded (maximize region), and the gray region marks where gain was penalized (minimize region).

phase. With a directional gain fitness function, the algorithm was able to evolve the geometry of an antenna array to maximize gain within a 30° sector centered on ϕ_{target} in the horizontal plane ($\theta = 90°$), while suppressing gain outside that region.

A future goal of this research is to connect antenna array geometry optimization to science outcomes by using fitness functions integrated with science simulation software. Here we have demonstrated a proof of concept that these techniques work for optimizing antenna array geometries. Moving forward, we will be able to build on these efforts to improve our techniques and design deployable arrays for science goals.

A. Future Work

Going forward, we will explore improving the capabilities of Nebulous and expanding the diversity of its designs. Increasing the number of user-configurable parameters is an area of active development, including implementing additional antenna types, node-generation schemes, mutation operators, and multi-objective selection schemes. Individuals will also have increased user-defined constraints to satisfy real-world requirements.

Future work will evaluate additional optimization approaches toward the goal of improving the computational efficiency of the array-design search. Improvements to the GA are currently in development to reduce array geometry evaluation times using evaluation caching [40, 41, 42]. Techniques from the GA literature for decreasing convergence time, such as adaptive mutation rates [43, 44], are also being assessed. To reduce the occurrence of extended periods of stagnation during evolutionary runs, we are updating the timing of ALPS injections to be responsive to current evolutionary conditions.

Hybrid model approaches being considered would incorporate particle swarm optimization and related heuristic search techniques [45, 46, 47, 48], as well as neural-network-based surrogate models [49, 50]. Particle swarm methods may provide a more directed exploration of the design space, reducing the number of low-quality candidate array geometries generated during the search. Neural-network surrogate models could be trained using previously evaluated individuals to estimate candidate fitness, thereby reducing the number of full-array performance simulations required during optimization.

The work presented in this report demonstrates the application of a genetic algorithm to antenna array geometry optimization. Further development of this approach will improve the efficiency of array geometry design-space exploration and support the identification of candidate configurations that meet mission-specific performance requirements.

Acknowledgments

The authors are grateful to the entire GENETIS collaboration for their prior work and continued assistance, especially Prof. Wolfgang Banzhaf, Bill Tozier, and the DevoLab at Michigan State University. We would also like to thank Remcom for their support, especially Tarun Chawla and Walter Janusz.

References

[1] R. Luebbers, "XFDTD and beyond-from classroom to corporation," *2006 IEEE Antennas and Propagation Society International Symposium*, pp. 119–122, 2006.

[2] J. Rolla, B. Reynolds, J. Weiler, D. Wells, M. Foreback, A. Connolly, E. Dolson, and C. Ofria, "Designing optimized antennas for science applications using evolutionary algorithms," Jet Propulsion Laboratory, Pasadena, California, Tech. Rep. 42-242, August 2025.

[3] J. Rolla, B. Reynolds, D. Wells, J. Weiler, A. Connolly, and R. Debolt, "Design of antennas from primitive shapes using genetic algorithms," Jet Propulsion Laboratory, Pasadena, California, Tech. Rep. 42-237, May 2024.

[4] S. K. Goudos, D. E. Anagnostou, C. Kalialakis, P. Vasant, and S. Nikolaou, "Evolutionary algorithms applied to antennas and propagation: Emerging trends and applications," *International Journal of Antennas and Propagation*, vol. 2016, no. 1, 2016. https://doi.org/10.1155/2016/5279647

[5] D. Marcano and F. Durán, "Synthesis of antenna arrays using genetic algorithms," *IEEE Antennas and Propagation Magazine*, vol. 42, no. 3, pp. 12–20, 2002.

[6] V. R. Mognon, W. Artuzi, and J. R. Descardeci, "Tilt angle and sidelobe level control of array antennas by using genetic algorithm," in *Proceedings of the 2001 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference.(Cat. No. 01TH8568)*, vol. 1, pp. 299–301, IEEE, 2001.

[7] A. Hussein, H. Abdullah, A. Salem, S. Khamis, and M. Nasr, "Optimum design of linear antenna arrays using a hybrid MoM/GA algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 1232–1235, 2011.

[8] Y. Li, W. Hong, M. Li, and G. Lu, "Unequally spaced linear antenna arrays synthesis based on genetic algorithm," in *2018 International Workshop on Antenna Technology (iWAT)*, pp. 1–4, IEEE, 2018.

[9] A. Bhargav and N. Gupta, "Multiobjective genetic optimization of nonuniform linear array with low sidelobes and beamwidth," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 1547–1549, 2013.

[10] E. A. Jones and W. T. Joines, "Design of Yagi-Uda antennas using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 9, pp. 1386–1392, 1997.

[11] S. Kiehbadroudinezhad, N. K. Noordin, A. Sali, and Z. Z. Abidin, "Optimization of an antenna array using genetic algorithms," *The Astronomical Journal*, vol. 147, no. 6, p. 147, 2014.

[12] G. Oliveri, P. Rocca, L. Poli, M. Carlin, E. T. Bekele, A. De Matteis, and A. Massa, "Evolutionary strategies for advanced array optimization," *2011 IEEE International Symposium on Antennas and Propagation (APSURSI)*, pp. 2441–2444, 2011.

[13] F. Grimaccia, M. Mussetta, A. Niccolai, and R. E. Zich, “Comparison of binary evolutionary algorithms for optimization of thinned array antennas,” *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, 2018.

[14] Q. Xu, S. Zeng, F. Zhao, R. Jiao, and C. Li, “On formulating and designing antenna arrays by evolutionary algorithms,” *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 2, pp. 1118–1129, 2021.

[15] C. A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed. Hoboken, NJ: John Wiley & Sons, 2016.

[16] R. J. Mailloux, *Phased Array Antenna Handbook*, 3rd ed. Norwood, MA: Artech House, 2018.

[17] R. C. Hansen, *Phased Array Antennas*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2009.

[18] H. L. Van Trees, *Optimum Array Processing: Part IV of Detection, Estimation, and Modulation Theory*. New York, NY: John Wiley & Sons, 2002.

[19] C. L. Dolph, “A current distribution for broadside arrays which optimizes the relationship between beam width and side-lobe level,” *Proceedings of the IRE*, vol. 34, no. 6, pp. 335–348, June 1946.

[20] T. T. Taylor, “Design of line-source antennas for narrow beamwidth and low side lobes,” *Transactions of the IRE Professional Group on Antennas and Propagation*, vol. 3, no. 1, pp. 16–28, January 1955.

[21] P. M. Woodward and J. D. Lawson, “The theoretical precision with which an arbitrary radiation-pattern may be obtained from a source of finite size,” *Journal of the Institution of Electrical Engineers*, vol. 95, no. 37, pp. 363–370, 1948.

[22] D. L. Jones, “Geometric configurations for large spacecraft-tracking arrays,” in *Proceedings of the 2003 IEEE Aerospace Conference*, vol. 2, pp. 2–997–2–1002, 2003.

[23] C. H. Lee, V. Vilnrotter, E. Satorius, Z. Ye, D. Fort, and K.-M. Cheung, “Large-array signal processing for deep-space applications,” *The Interplanetary Network Progress Report*, vol. 42-150, Jet Propulsion Laboratory, California Institute of Technology, pp. 1–21, 2002.

[24] J. Rolla, A. Machtay, A. Patton, W. Banzhaf, A. Connolly, R. Debolt, L. Deer, E. Fahimi, E. Ferstl, P. Kuzma, C. Pfendner, B. Sipe, K. Staats, and S. A. Wissel, “Using evolutionary algorithms to design antennas with greater sensitivity to ultra high energy neutrinos,” *Phys. Rev. D*, vol. 108, p. 102002, Nov 2023. https://link.aps.org/doi/10.1103/PhysRevD.108.102002

[25] B. Reynolds, J. Rolla, A. Machtay, W. Banzhaf, D. Calderon, C. Chen, A. Connolly, R. Debolt, E. Fahimi, N. King, E. Melotti, A. Patton, B. Sipe, K. Staats, D. Wells, S. Wissel, and A. Zinn, “Design of antennas with greater sensitivity to ultra-high energy neutrinos,” *National Radio Science Meeting*, January 2023.

[26] A. Machtay, J. Rolla, and A. Patton, “Using genetic algorithms to optimize antenna designs for improved sensitivity to ultra-high energy neutrinos,” *38th International Cosmic Ray Conference*, 2023.

[27] J. Rolla, D. Arakaki, M. Clowdus, A. Connolly, R. Debolt, L. Deer, E. Fahimi, E. Ferstl, S. Gourapura, C. Harris, L. Letwin, A. Machtay, A. Patton, C. Pfendner, C. Sbrocco, T. Sinha, B. Sipe, K. Staats, J. Trevithick, and S. Wissel, “Evolving antennas for ultra-high energy neutrino detection,” *37th International Cosmic Ray Conference*, 2021.

[28] J. Rolla, A. Connolly, K. Staats, S. Wissel, D. Arakaki, I. Best, A. Blenk, B. Clark, M. Clowdus, S. Gourapura, C. Harris, H. Hasan, L. Letwin, D. Liu, C. Pfendner, J. Potter, C. Sbrocco, T. Sinha, and J. Trevithick, "Evolving antennas for ultra-high energy neutrino detection," *36th International Cosmic Ray Conference*, 2019.

[29] M. Foreback, E. Imata, V. Ragusa, J. Weiler, J. Sy, C. Shao, J. Wagner, D. Wells, R. Marcusen, K. G. Skocelas, et al., "Eclipse: An evolutionary computation library for instrumentation prototyping in scientific engineering," *arXiv preprint arXiv:2601.05098*, 2026.

[30] S. T. Peters, K. Nessly, T. M. Roberts, D. M. Schroeder, and A. Romero-Wolf, "Spatial coherence constraints on passive radar sounding with radio-astronomical sources," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–12, 2024.

[31] A. Romero-Wolf, G. Steinbruegge, J. Castillo-Rogez, C. J. Cochrane, T. A. Nordheim, K. L. Mitchell, N. S. Wolfenbarger, D. M. Schroeder, and S. T. Peters, "Feasibility of Passive Sounding of Uranian Moons Using Uranian Kilometric Radiation," *Earth and Space Science*, vol. 11, no. 2, 2024.

[32] S. Jester and H. Falcke, "Science with a lunar low-frequency array: From the dark ages of the universe to nearby exoplanets," *New Astronomy Reviews*, vol. 53, no. 1–2, pp. 1–26, 2009.

[33] J. O. Burns, R. MacDowall, S. Bale, G. Hallinan, N. Bassett, and A. Hegedus, "Low radio frequency observations from the moon enabled by NASA landed payload missions," *The Planetary Science Journal*, vol. 2, no. 2, p. 44, 2021.

[34] G. S. Hornby, "Alps: The age-layered population structure for reducing the problem of premature convergence," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 815–822, Association for Computing Machinery, 2006. https://doi.org/10.1145/1143997.1144142

[35] ——, *A Steady-State Version of the Age-Layered Population Structure EA*. Boston, MA: Springer, 2010. https://doi.org/10.1007/978-1-4419-1626-6_6

[36] J. Zhong, X. Hu, J. Zhang, and M. Gu, "Comparison of performance between different selection strategies on simple genetic algorithms," *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 2, pp. 1115–1121, 2005.

[37] A. Shukla, H. M. Pandey, and D. Mehrotra, "Comparative review of selection techniques in genetic algorithm." *2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE)*, pp. 515–519, 2015.

[38] B. L. Miller, D. E. Goldberg, et al., "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, no. 3, pp. 193–212, 1995.

[39] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*, vol. 1, edited by G. J. E. Rawlins. Amsterdam, The Netherlands: Elsevier, 1991. https://www.sciencedirect.com/science/article/pii/B9780080506845500082

[40] M. W. Przewozniczek and M. M. Komarnicki, "Fitness caching - from a minor mechanism to major consequences in modern evolutionary computation," in *2021 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1785–1791, 2021.

[41] P. Dutta and A. Mukhopadhyay, "Improved compact genetic algorithms with efficient caching," *arXiv preprint arXiv:2504.02972*, 2025.

[42] E. E. Santos and E. Santos, "Reducing the computational load of energy evaluations for protein folding," in *Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering*, pp. 79–86, IEEE, 2004.

[43] D.-C. Dang and P. K. Lehre, "Self-adaptation of mutation rates in non-elitist populations," in *Parallel Problem Solving from Nature – PPSN XIV*, pp. 803–813, Springer International Publishing, 2016.

[44] B. Doerr, C. Witt, and J. Yang, "Runtime analysis for self-adaptive mutation rates," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 1475–1482, Association for Computing Machinery, 2018. https://doi.org/10.1145/3205455.3205569

[45] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Mathematical Problems in Engineering*, vol. 2015, Oct 2015. https://doi.org/10.1155/2015/931256

[46] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview," *Soft Computing*, vol. 22, pp. 387–408, 2018.

[47] G. Papazoglou and P. Biskas, "Review and comparison of genetic algorithm and particle swarm optimization in the optimal power flow problem," *Energies*, vol. 16(3), p. 1152, 2023.

[48] R. Thangaraj, M. Pant, A. Abraham, and P. Bouvry, "Particle swarm optimization: Hybridization perspectives and experimental illustrations," *Applied Mathematics and Computation*, vol. 217(12), pp. 5208–5226, 2011.

[49] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9(1), pp. 3–12, 2005.

[50] F. van den Bergh and A. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8(3), pp. 225–239, 2004.

APPENDICES

I. Individual Genes and Genetic Algorithm Parameters

Table 1. Parameters used in genetic algorithm

Category	Type	Parameters
General	Run	Fitness Function, Seed, Max. Workers
	ALPS Evolver	Layer Size, Num. Layers, Layer Order, Tournament Size, Age Gap, Refresh Interval, Elitism, Num. Evaluations, Age Progression
	XFdtd	Frequency Range, Frequency Step, Units, Gain Type
Array Individual	Node Settings	Base Antenna, Num. Nodes, Min. Distance Between Nodes, Max. Grid Rows, Max. Grid Cols
	Dist. 1	Min., Max., St. Dev., Mutation Probability
	Dist. 2	Min., Max., St. Dev., Mutation Probability
	Angle	St. Dev., Mutation Probability
	Azimuth	Enabled, Per-node, St. Dev., Mutation Probability
Fitness Target	Maximize	Zenith and Azimuth Angle Ranges
	Minimize	Zenith and Azimuth Angle Ranges

Table 2. Array Individual Genes

Gene	Values
Inter-Column Spacing, d_1	Integer value in [`dist1_min`, `dist1_max`]
Inter-Row Spacing, d_2	Integer value in [`dist2_min`, `dist2_max`]
Column Offset Angle, ψ	Integer value between 1° and 90°
Antenna Azimuth, α	Integer azimuth angle between 0° and 359°, shared across all antennas

II. Nebulous Hill Climber Results

A basic hill climber algorithm [29] was used to evolve antenna arrays under two different sets of conditions. For each condition, the hill climber was run five times (each with a unique random number generator seed). The results of the five runs were averaged and are presented below.

A. Gain and Phase Euclidean Distance from Target Minimization with Hill Climber

A hill climber was used to optimize an array of nine dipole antennas to a target radiation pattern. The target radiation pattern was generated by selecting the antenna coordinates of the target array and simulating it with XFdtd. An individual's fitness was calculated via the Euclidean distance fitness function as described in Section II.B.1. The same target radiation pattern was used for this experiment as in the evolution experiment.

Figure 8 shows the results of the hill climber when evaluated at a single frequency of 428 MHz. As in the evolutionary experiment, maximum rows and columns were each set to three, and number of antennas was fixed at nine. The average Euclidean distance achieved was 47.143, and the lowest Euclidean distance reached by any single run was 43.837. These high distance scores show the hill climber was unable to successfully match the target complex radiation pattern. This is also apparent in the 2D geometry plot (see Figure 9) and gain and phase plots (see Figure 10) comparing the best individual the hill climber runs produced to the target array.

B. Target Direction Gain Maximization of a Yagi-Uda Array with Hill Climber

A hill climber was used to optimize an array of nine Yagi-Uda elements toward a target steering direction. The target steering direction $\phi_{\text{target}} = 0°$ (broadside to the array x-axis) was evaluated, where ϕ denotes the azimuthal angle in spherical coordinates. The optimizer maximized gain within a 30° sector centered on ϕ_{target} in the horizontal plane ($\theta = 90°$), while suppressing gain outside that region. As in the dipole experiment, maximum rows and columns were each set to three, and the number of antennas was fixed at nine.

Five replicate hill climber runs were performed using different random seeds. Figure 11 shows the fitness convergence of each run. The highest fitness score reached by any single run was 11.529, while the average fitness score across all five seeds was 4.929, with individual runs ranging as low as 0.790, indicating substantial run-to-run variability in how well the hill climber was able to steer gain toward the target sector.

The gain pattern of the best individual produced across the five runs, shown in both the zenith and azimuth planes, is given in Figure 12. In the zenith plane, the gain pattern forms two broad lobes separated by deep nulls at exactly $\theta = 0°$ and $\theta = 180°$. The target sector at $\theta = 90°$ falls within the left-hand lobe, but the lobe is much wider

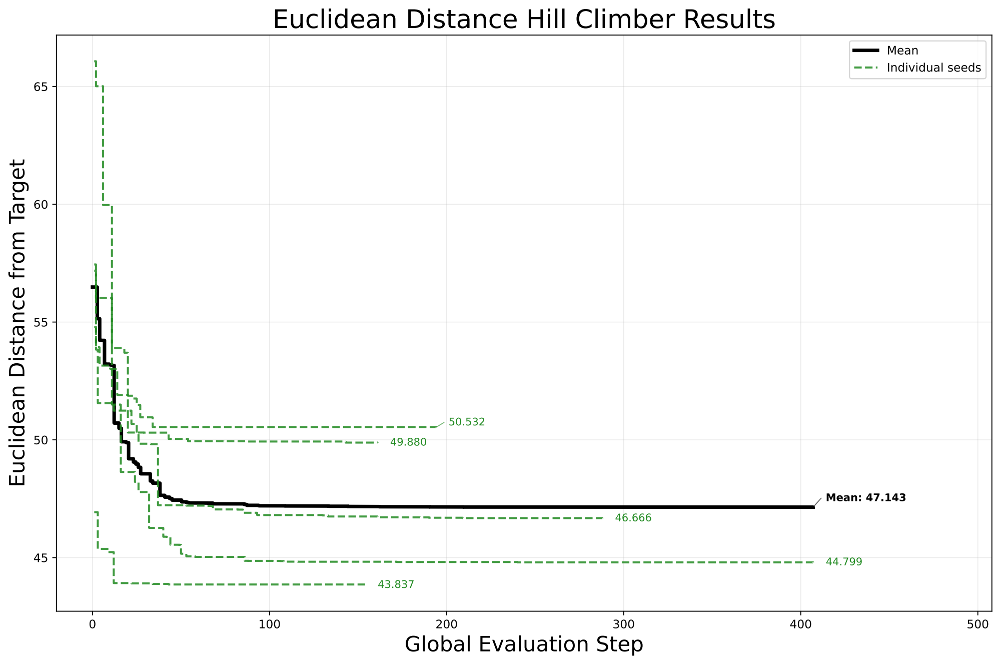


Figure 8. Euclidean distance minimization of five replicate hill climber runs optimizing a nine-dipole array to a target radiation pattern at 428 MHz. Each dashed green line is an individual run; the solid black line is the mean across seeds. Lower values indicate a closer match to the target.

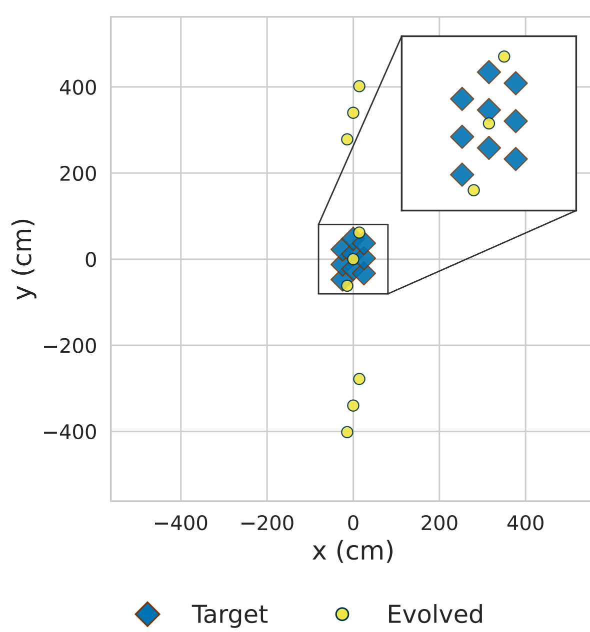


Figure 9. The 2D geometry of the most fit individual as of the final evaluation step of the single-frequency dipole array optimization toward a target radiation pattern using the hill climber algorithm, maximum three rows and columns, and nine nodes. Evolved individual node coordinates (yellow circles) are overlaid on the fixed target array geometry (blue diamonds). Inset: magnified view of the target node cluster (boxed region).

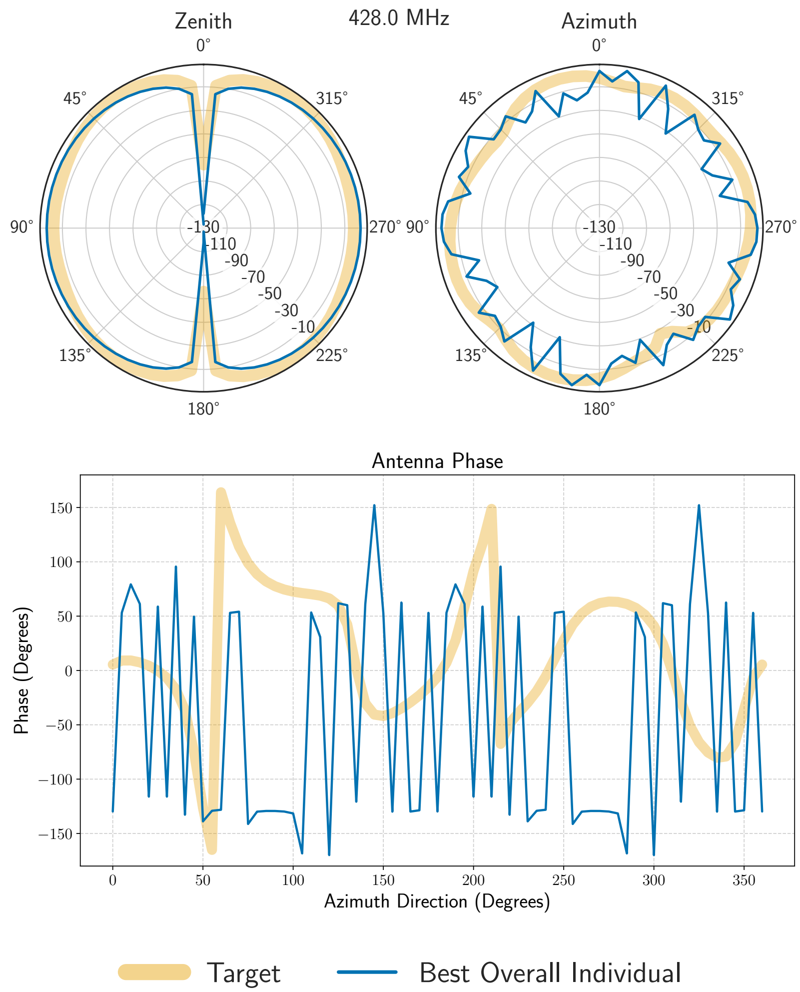


Figure 10. The gain (top) and phase (bottom) patterns of the most fit individual as of the final evaluation step of the single-frequency dipole array optimization toward a target radiation pattern using the hill climber algorithm, maximum three rows and columns, and nine nodes. Also overlaid on each is the corresponding pattern of the target. The target's line is enlarged for visibility.

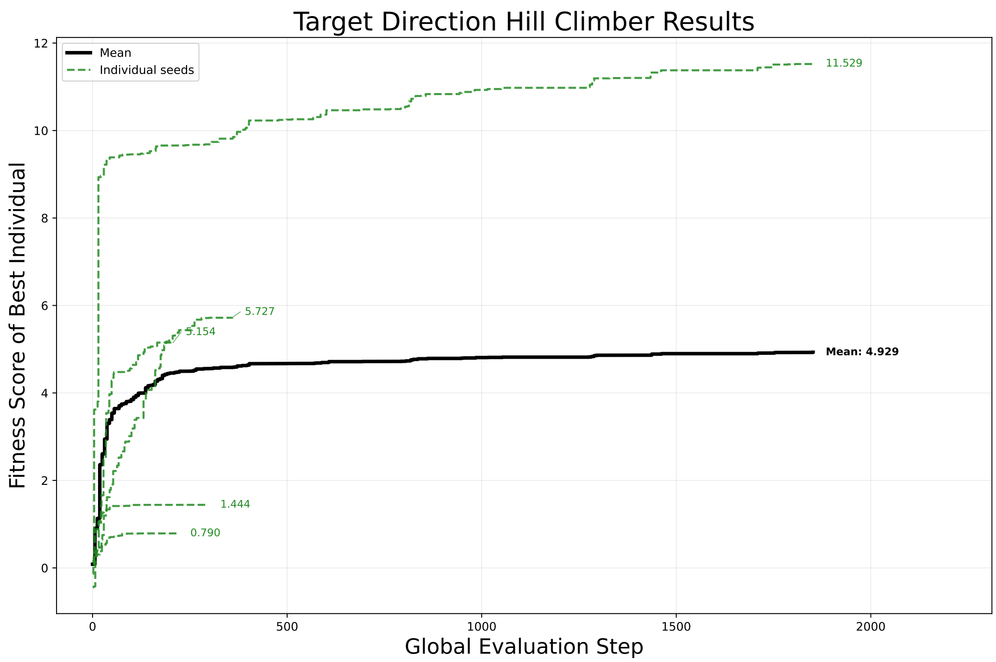


Figure 11. Fitness convergence of five replicate hill climber runs optimizing a Yagi-Uda antenna array toward higher gain in the target direction and lower gain in the off-target directions. Each green dashed line is an individual run; the solid black line is the mean best fitness score across replicate runs. Higher fitness score values indicate better alignment with the target direction and minimization in the off-target directions.

than the target wedge, showing inadequate minimization in the off-target directions adjacent to the target. In the azimuth plane, the pattern shows a clear dominant main lobe oriented toward $\phi_{\text{target}} = 0°$, reaching gains several dBi higher than the surrounding directions, alongside a smaller back lobe near $\phi = 180°$ and a handful of low-level sidelobes scattered elsewhere. Overall, the best individual achieves reasonably effective azimuthal steering toward the target, but the zenith pattern fails to concentrate gain in the target area, instead producing a large lobe from near $\theta \approx 45°$ to $\theta \approx 135°$.

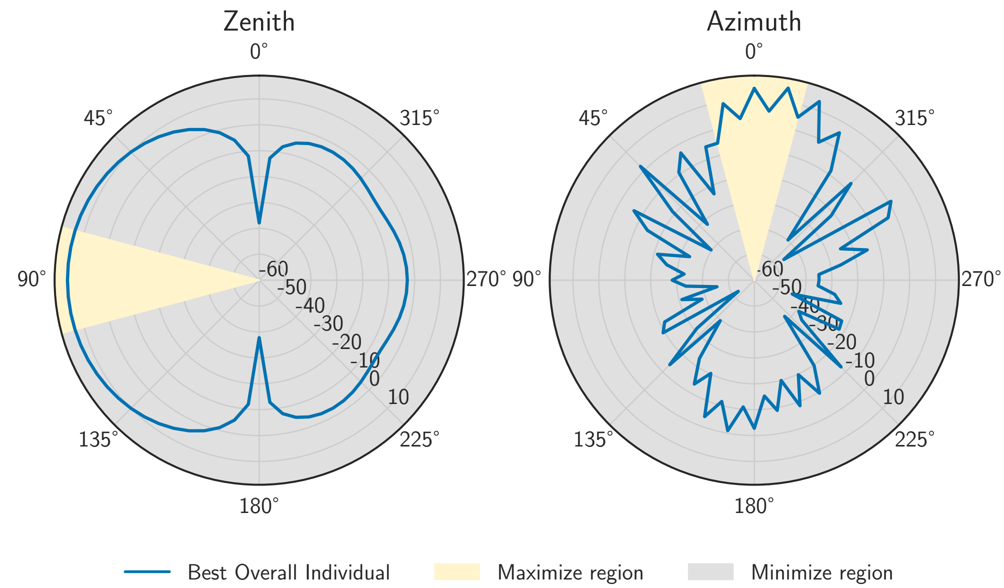


Figure 12. Zenith (left) and azimuth (right) gain patterns for the best-overall individual from the Yagi-Uda antenna array geometry optimization runs using the hill climber algorithm, maximum three rows and columns, and nine nodes. The radial axis shows gain in dBi (gridlines from -60 dBi at center to 20 dBi at the outer edge) as a function of angle. The blue trace is the simulated gain pattern of the best individual; the yellow wedge marks the angular region where gain was rewarded (maximize region), and the gray region marks where gain was penalized (minimize region).

III. Human-Designed Reference Antenna Arrays

To provide a meaningful baseline for the evolved directional array geometry, a hand-designed Yagi-Uda array was constructed with director/reflector spacings drawn from standard engineering design tables. This hand-designed array represents the gain achievable through established design heuristics alone, without the benefit of evolutionary search over the geometry space. Comparing the evolved array's performance against its hand-designed counterpart for the same target direction isolates the contribution of the evolutionary process itself, distinguishing gains attributable to the search algorithm from gains any competently engineered array would already achieve. This comparison also functions as a sanity check on the optimization pipeline as an evolved array that failed to outperform its hand-designed baseline would signal a problem with the fitness function or search configuration rather than a genuine limit on achievable gain.

A nine-element unidirectional Yagi-Uda array was constructed by spacing the elements by one-half wavelength (35 cm) in a straight line. The array was steered to a target azimuth of $\phi_{\text{target}} = 0°$ (broadside to the array's x-axis), with gain maximized within a 30° sector centered on ϕ_{target} in the horizontal plane ($\theta = 90°$) and suppressed elsewhere. The array's gain at a single frequency of 428 MHz was simulated using XFdtd (see Figure 13). The array achieved a fitness score of 11.482 when evaluated by the Nebulous software. Given the directionality of the Yagi-Uda design, the back lobe ($\phi = 180°$) is well minimized and the azimuth peak is well defined. Two areas leave room for improvement: the azimuth side lobes could be further reduced, and the zenith slice does not minimize gain well in the region immediately adjacent to the maximization sector.

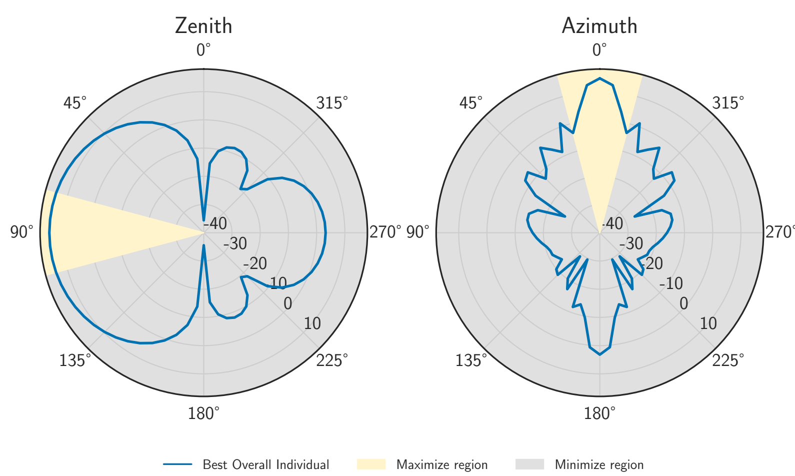


Figure 13. Zenith (left) and azimuth (right) gain patterns for the human-designed nine-element Yagi-Uda array. The radial axis shows gain in dBi (gridlines from -40 dBi at center to 20 dBi at the outer edge) as a function of angle. The blue trace is the simulated gain pattern of the array; the yellow wedge marks the angular region where gain was rewarded (maximize region), and the gray region marks where gain was penalized (minimize region).