Using concepts from biology to improve problem-solving methods

Erik D. Goodman*[a], Edward J. Rothwell[b], Ronald C. Averill[c]
[a]BEACON Center for the Study of Evolution in Action, Michigan State Univ., E. Lansing, MI 48824, USA
[b]Dept. of Electrical & Computer Engineering, Michigan State Univ., E. Lansing, MI 48824, USA
[c]Dept. of Mechanical Engineering, Michigan State Univ., E. Lansing, MI 48824, USA

ABSTRACT

Observing nature has been a cornerstone of engineering design. Today, engineers look not only at finished products, but imitate the evolutionary process by which highly optimized artifacts have appeared in nature. Evolutionary computation began by capturing only the simplest ideas of evolution, but today, researchers study natural evolution and incorporate an increasing number of concepts in order to evolve solutions to complex engineering problems. At the new BEACON Center for the Study of Evolution in Action, studies in the lab and field and *in silico* are laying the groundwork for new tools for evolutionary engineering design. This paper, which accompanies a keynote address, describes various steps in development and application of evolutionary computation, particularly as regards sensor design, and sets the stage for future advances.

Keywords: bio-inspired design, evolutionary computation, genetic algorithms, evolutionary design, antenna design, self-structuring antennas, metamaterial design

1. INTRODUCTION

Today, engineering design makes use of at least two major forms of inspiration from biology – biomimetic designs and biologically inspired design processes, especially evolutionary design. In the former, engineers observe structures or mechanisms in nature and copy them in their designs. These observations have inspired many notable designs, including arguably the earliest airplanes, Velcro fasteners, and many more. But often, the environment in which an engineered artifact must operate, or the materials from which it is to be built, do not resemble very strongly something found in nature, and we decide, instead, to mimic a *process* by which biological organisms have evolved or a manner in which they behave. That can be done in various ways, but among the most common is to use some form of *evolutionary computation*, such as a genetic algorithm, and a computational simulation that allows us to judge the performance of an arbitrary design in environments representing those in which the design will need to perform in the world. It is this type of process that will be explored here. Other nature-inspired methods such as ant-colony optimization, particle swarm optimization, etc., will not be discussed here, although they are also very relevant.

1.1 Beginnings of evolutionary computation

The history of harnessing evolutionary approaches to design engineering artifacts now extends nearly fifty years. In the 1964, Ingo Rechenberg, at Technical University of Berlin, conducted a series of wind-tunnel experiments in which he made mutations in each design parameter prior to each test, then used survival of the fittest between the "parent" and "offspring" to determine which design to use as the basis of the next one[1]. He found that, on strongly multimodal problems such as encountered in fluid dynamics, this approach outperformed a classical steepest-descent gradient approach. While this work was done without computers, he and Hans-Paul Schwefel later formalized the *evolutionsstrategie* algorithms ("evolution strategies"), as presented in 1973[2]. During the 1960's, John Holland, at the University of Michigan, was formulating and proving theorems about what have come to be known as *genetic algorithms.* In contrast to the work of Rechenberg and Schwefel, which started with physical experiments on engineering artifacts, Holland's work began with theory and computation, dealing with evolution of solutions to various

* goodman@egr.msu.edu; phone 1 517 355-6453; fax 1 517 355-7248; http://beacon-center.org

Evolutionary and Bio-Inspired Computation: Theory and Applications V, edited by Misty Blowers, Teresa H. O'Donnell, Olga Lisvet Mendoza-Schrock, Proc. of SPIE Vol. 8059, 805902 · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.889070

mathematically-expressed “toy problems.” Although Holland’s classic *Adaptation in Natural and Artificial Systems*[3] book did not appear until 1975, by the late 1960’s, he was teaching graduate courses about these topics, and one of the authors (Goodman) had the good fortune to take those classes and to join Holland’s research group, the Logic of Computers Group. Goodman’s Ph.D. research involved developing a model of bacterial growth in a variety of media, and to parameterize the microbial growth model, he formulated and ran a genetic algorithm that took nearly a year of computer time on a state-of-the-art minicomputer (IBM 1800) to generate a single solution, in the first use of a genetic algorithm to seek an unknown solution to a real problem. The long computation time required helps to clarify why evolutionary computation was not used heavily to solve real-world applications until the 1980’s, although the theoretical foundation had long been established: (1) one must be able to simulate the performance of hundreds or (typically) thousands of different designs in order for a genetic algorithm to find one worth actually building, and (2) the computational model must represent the true design faithfully enough that the performance of the design built from the optimal model has performance close to that predicted by the simulation. These two factors began to make some real-world problems addressable by evolutionary computation in the mid-1980’s, and today, problems in structural design, fluid dynamics, scheduling, layout, routing and logistics, and electromagnetics are routinely solved using evolutionary methods.

“Simple” genetic algorithms, as described in Goldberg’s classical 1989 book *Genetic Algorithms in Search, Optimization and Machine Learning*[4], maintain at any time a population of solutions being used to generate future offspring—new possible solutions to explore. They were inspired by biological evolution, although they dramatically simplify the mechanisms. Using the classical genetic algorithm operators—mutation with fixed probability at each locus, crossover, fitness-proportional selection and children replacing parents—the populations have a tendency to converge rapidly, so that all members of the population resemble each other, eventually approaching *popsize* copies of the best solution discovered to date. But even this “simple GA” mechanism is powerful enough to solve many practical real-world problems in the domain of electromagnetics and sensors. Before going on, let’s mention a few such applications.

2. EXAMPLES OF GA APPLICATION IN ELECTROMAGNETICS AND ANTENNA DESIGN

2.1 Early work

The Electromagnetics Research Group at MSU has been using GAs to solve a variety of optimization problems since the mid 1990’s. Originally GAs were employed in signal analysis for radar target identification. Applications include radar target natural resonance extraction from measured data sets[5,6], splicing data measured in different frequency bands[7,8], and radar target scattering center recovery[9].

2.2 Antenna design

Based on his work on radar target signature analysis, John Ross, a PhD graduate and post-doctoral researcher at MSU, began to apply GAs to antenna design and optimization. He continued this work at General Motors Research and Development Center and later as a consultant for Delphi Research Laboratories, and culminated it in the development of the software suite “GA-NEC.” At Delphi, he used a GA for design and refinement of in-glass and backlight automotive antennas[10]. Eventually, he formed the company *Viamorph* (http://www.viamorph.com) which applies GA optimization to a variety of antenna problems of both commercial and military emphasis. Recently, Viamorph has been working with the company Antennas Direct (http://www.antennasdirect.com), using GAs to design their Clearstream® line of antennas (Figure 1) for use in consumer digital television[11,12].

2.3 Self-structuring antennas

The concept of an antenna that is able to organize itself into a variety of electrical configurations in response to changes in its environment was conceived by Ed Rothwell in 1998. He built and tested a prototype at MSU and named the technology a “self-structuring antenna” or SSA. The fundamental concept is to provide an antenna template of elements interconnected with N simple on/off switches such that the template may be arranged into 2^N different combinations. Even modest numbers of switches (such as the 32 used in early templates) provide extremely large numbers of possible configurations. Unlike typical antenna design approaches it is assumed that the performance of any individual antenna configuration is unknown at the onset of operation and thus useful configurations are found only after searching through many possible candidate arrangements. This can be done previous to operation and stored in memory, or done in real time in response to changes in operating conditions or physical environment. In either case, the success of the concept is dependent on having an effective means for rapidly searching through the large pool of candidate configurations. A GA

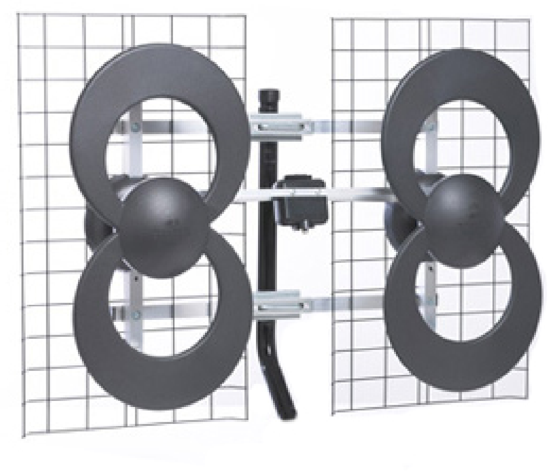


Figure 1. Antennas Direct Clearstream4 antenna designed using GA software.

was first used for this search due to the inherently binary nature of the switch configurations. After more than a decade of research into SSAs, a GA remains the algorithm of choice for finding appropriate switch states.

The SSA concept was patented by MSU in 2001 and first reported in 2000[13] and 2002[14]. Early results were obtained using the GA-NEC software developed by John Ross. An early prototype developed by students at MSU is shown in Figure 2. Collaboration between MSU and John Ross at Viamorph led to successful SBIR grants to study military applications of SSA technology (see Figure 3.) Military interest is particularly high due to the inherent survivability of the SSA. If a portion of the antenna template is destroyed, the remaining portion can reconfigure to compensate for the loss.

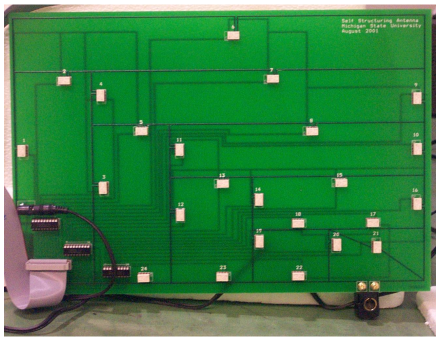


Figure 2. Early prototype SSA

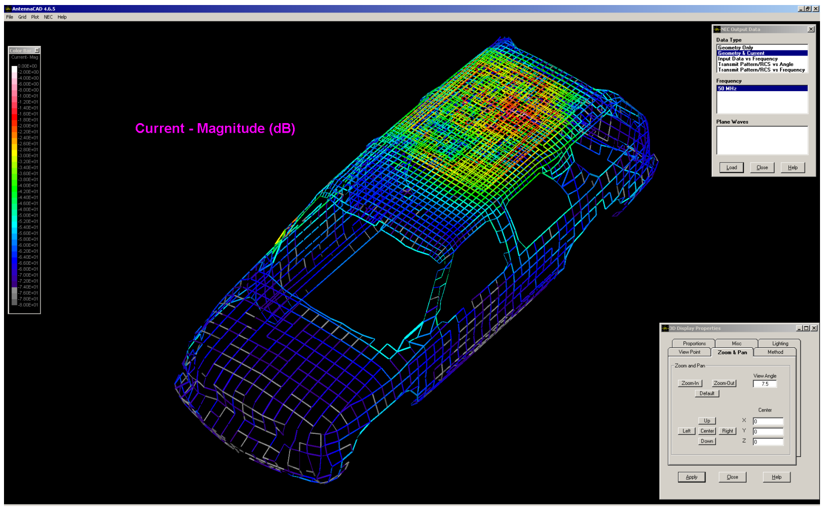


Figure 3. Low-profile vehicle communications antenna explored under US army SBIR

Delphi Research Laboratory became interested in SSA technology for use in automotive applications and supported SSA research at MSU for several years starting in 2002. This collaboration led to the production of many research papers and graduate theses, and Delphi obtained many patents related to using SSAs in the automobile industry. A typical application of an FM radio backlight antenna is shown in Figure 4. Delphi's interest stems from the possible use of a

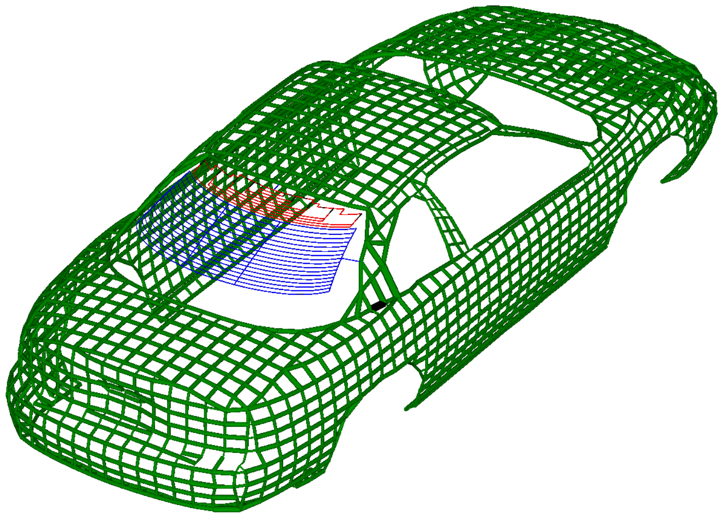


Figure 4. Backlight SSA designed for Delphi Research Laboratories

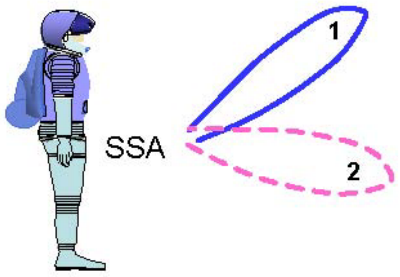


Figure 5. SSA for NASA return-to-moon program

single SSA to replace the large number of antennas currently deployed on a typical automobile (AM, FM, television, GPS, keyfob, satellite radio, cellular telephone, WIFI, WIMAX, etc.), and also from the ability of the SSA to respond to changes in the antenna environment. These changes can be due to car motion, vehicle alteration, or the addition of aftermarket products. Additionally, the SSA concept removes the need for the extensive engineering design currently needed for the creation of automobile antennas.

In 2007, the company *Monarch Antennas* (http://www.monarchantenna.com/index.html) was spun out from Delphi to commercialize SSA technology, and was granted a license of the MSU patent. Monarch has obtained several external grants and is currently developing the SSA for Wifi and Zigbee applications[15], for structural health monitoring[16], and for NASA astronauts to use in the return-to-moon program (Figure 5).

Work has continued at MSU on SSAs, and many new applications of the technology have been explored. These include a helmet-worn antenna (Figure 6), a body-worn antenna (Figure 7), a patch antenna (Figure 8) and a leaky-wave antenna (Figure 9). The patch antenna has potential applications for consumer communications, such as WIFI and cellphones, while the leaky-wave antenna has applications to commercial and military air vehicles. MSU has applied for patents on both the patch and leaky wave antennas.

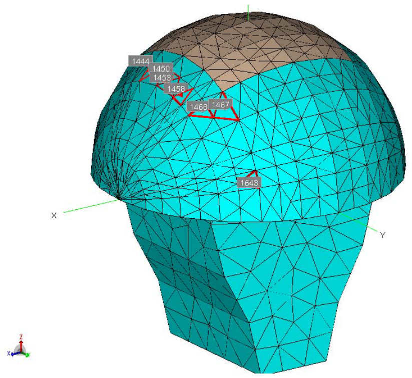


Figure 6. A helmet-worn SSA

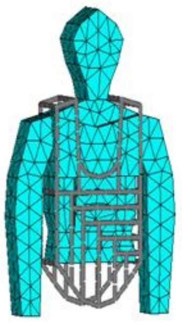
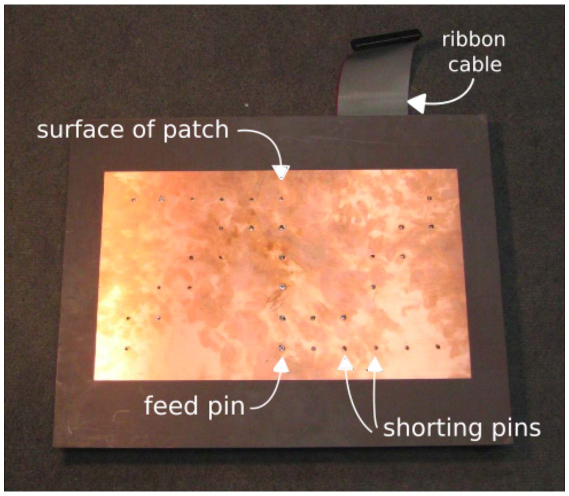


Figure 7. A body-worn SSA



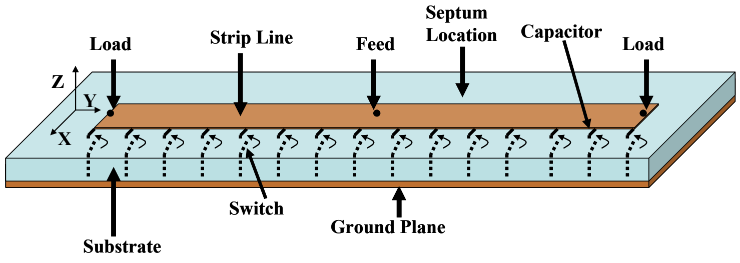


Figure 9. A leaky-wave SSA

Figure 8. A patch SSA

2.4 Applications related to SSAs

The SSA concept can be applied to numerous other areas of microwave and RF engineering. In these applications a structure is created with individual substructures interconnected by switches in order to perform some task. By using a GA to search among switch configurations, the performance of the structure may be optimized. Among the wide variety of applications conceived at MSU are a self-structuring electromagnetic cavity (Figure 10) that may be tuned to various frequencies using switches, an electromagnetic shutter that may be open or closed to electromagnetic waves by choosing appropriate switch configurations (Figure 11), a self-structuring 2-port network that may be configured as an antenna tuner, filter, or distribution network (Figure 12), and a self-adjusting scatterer in which the scattering properties of a surface are automatically adjusted (Figure 13).

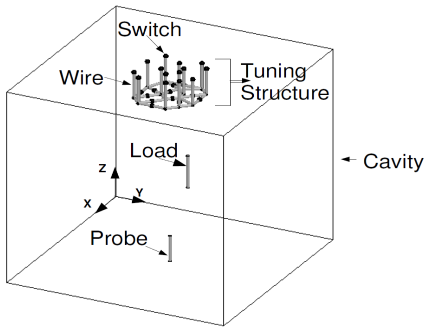


Figure 10. A self-structuring electromagnetic cavity

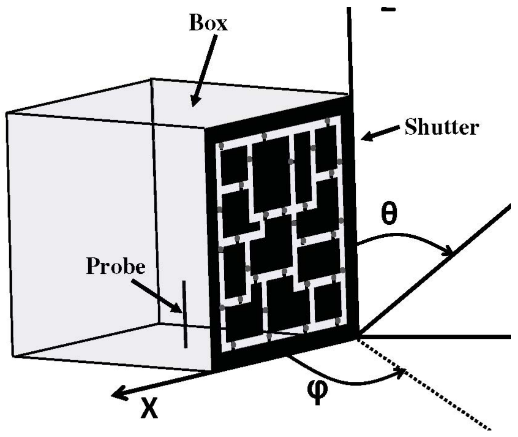


Figure 11. A self-tuning electromagnetic shutter

2.5 Limitations of the simple GA

As illustrated in these examples, simple genetic algorithms can solve many design problems very effectively, particularly if the search space is not too large or if there is a fairly high density of "good enough" solutions in the space. But there are also many more difficult problems in which the simple GA will prematurely converge to a solution that is not satisfactory. In that situation, it is important **not** to let the population reach a converged condition before the space has been adequately searched, because once it converges, its capacity for effective searching is essentially exhausted. Therefore, throughout the history of evolutionary computation, researchers have sought ways of maintaining search efficiency, capitalizing on what is being learned, while fending off "premature" convergence, without increasing the population size so much as to slow the overall solution process dramatically. In many cases, they have looked to nature for examples of factors or processes that assist evolution in nature to work as effectively as it has worked and continues to work, with the goal of mimicking those phenomena in an evolutionary computation algorithm. Let's look at some more ideas borrowed from nature for use in a GA.

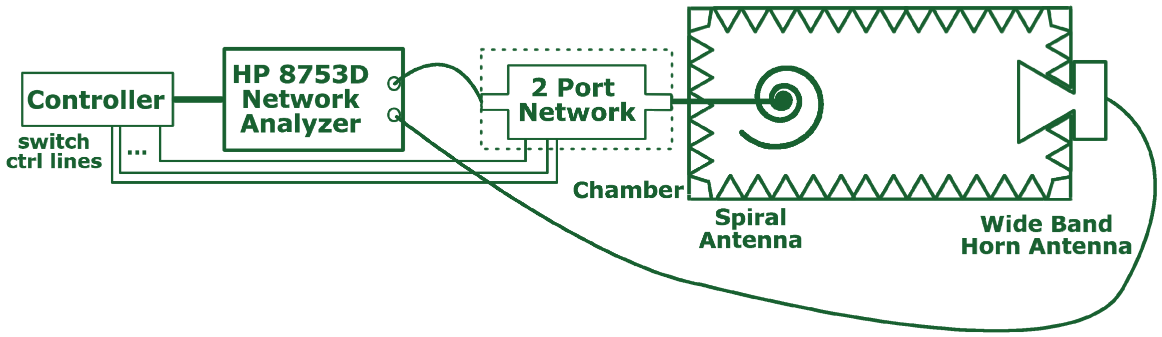


Figure 12. A self-structuring 2-port network

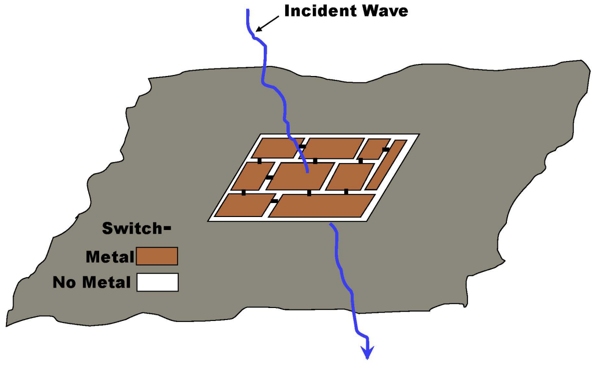


Figure 13. A self-adjusting scatterer

3. BORROWING MORE FROM NATURE

3.1 Selection for reproduction

In nature, all members of a species do not usually co-exist in a uniform environment, and generally are not all in direct competition with one another. Several innovations beyond the simple GA have derived from this observation. Fitness-proportional selection, as a means of determining probability that any individual in a population will generate offspring, is a poor representation of nature. Instead, tournament selection, in which two or more individuals compete only against each other, with the most-fit individual winning a chance to produce offspring, is much less prone to premature convergence, and does not require continual rescaling of the fitness domain as the population begins to converge[4]. Removing the population from acting like it is in a single, well-mixed reactor is also important. GAs have used two main ways to do this. One was to divide a population into several distinct ("island") subpopulations, with breeding only occurring within a subpopulation and with only occasional migration or exchange of individuals among the subpopulations in some user-specified pattern. This was also called a "parallel GA," but could be implemented within a single processor, of course, so is completely independent of the speedups available by distributing a GA's subpopulations or fitness evaluations among multiple processors (which can be implemented for any kind of GA architecture, including island parallel subpopulations)[17]. The second way was to assign each individual in a population a physical location on a tessellation, and to allow breeding and placement of offspring only in the local neighborhood of any individual[18]. Either of these approaches can slow the convergence of a population, preserving the overall diversity to allow productive search to continue longer than in a comparably-sized single well-mixed population.

Whatever the architecture for the population, note that the expansion of a GA to run on many processors simultaneously allows practical use of much larger population sizes, and, if combined with island parallel populations, resembles the proliferation of organisms to large numbers distributed across weakly interacting neighborhoods, as is often found in nature. Large, well-mixed populations also help to defer premature convergence, but at the cost of efficiency in exploiting the desirable features already learned about the fitness landscape.

3.2 Replacement policy

Replacement policies determine which individuals in a population are replaced by new offspring in order to maintain the classical fixed population size of a simple genetic algorithm. Classically, children replace parents: two parents are crossed over producing two offspring, the resultant offspring are mutated, and the children replace the parents in the population for the next generation. An alternative is for children to replace parents only if they are more fit. Neither of these reflects nature particularly well, especially given that *age* is often another strong factor—one organism outlives or, more importantly, out-reproduces another because it has matured and learned, but eventually dies of old age or from factors linked to aging. Of course, for organisms reproducing asexually by binary fission, there is no concept of age of a parent relative to that of a child, so it may be more natural to simply remove individuals from the population as new ones are introduced, either at random and a constant rate (as happens in continuous culture in a chemostat) or in a form more reminiscent of a space-filling population, in which more-fit individuals may displace less-fit ones. Computationally, this is sometimes modeled as *k*-elitism, in which the *k* most-fit individuals of the union of parent and offspring populations survive[19].

3.3 Sustainable Evolutionary Methods

In nature, there are many examples of competition being tempered so as to protect younger and weaker individuals from being systematically killed by stronger, more experienced individuals, allowing the newer individuals to mature before being subjected to the strongest competition. This observation was the inspiration for the Hierarchical Fair Competition[20] principle described next, which makes available a form of evolutionary computation that can continue to explore new regions of the solution space indefinitely, yet still allows rapid exploitation of the information gathered early in the search. The analogy is not complete—new, promising designs do not get older and more mature, but instead are refined by mutation and crossover with other designs. Nonetheless, it was the observations of processes in nature that inspired the creation of HFC.

In most evolutionary computation algorithms, populations tend to converge toward the best individuals found to date. As a result, they typically "fine-tune" the best individuals found fairly early in the search, rather than continuing to seek radically different new (and potentially better) individuals. In such scenarios, it is not practical to extend the search indefinitely simply by continually introducing new, random individuals… adding new random designs maintains the *statistical* diversity and the *possibility* of generating very different and superior individuals, but the probability of doing so generally remains vanishingly low. That is because the fitnesses of the new, random individuals and offspring resulting from combining them with other individuals in the population are generally much lower than those of the finely tuned remainder of the population against which they must compete for survival and reproductive opportunities. To combat this, in 2002, Hu et al., in Goodman's lab, introduced the Hierarchical Fair Competition[20] (HFC) principle. HFC stratifies a population according to fitness, and allows competition for reproduction to occur only among individuals of similar fitnesses. Mating occurs between individuals within each fitness range, and offspring of these matings never replace individuals in lower fitness ranges, and typically replace individuals of their own fitness range. However, when an offspring with much higher fitness is generated, it replaces an individual in its (higher) fitness range, so that it does not kill off its parents and monopolize future matings in the fitness range of its parents. Thus, "good guys move up, but bad guys don't move down." At the lowest range, new random individuals are continually introduced, and because they are competing only against other similarly-low-fitness individuals, they can be mutated and recombined, occasionally resulting in better offspring that move up, with new genetic material, to the next-higher fitness level. Using HFC, productive evolution and introduction of novel genetic material can continue indefinitely, producing "sustainable evolution." An ordinary genetic algorithm or genetic programming run tends to preclude any effect of new randomly-generated individuals; the population's average fitness climbs, but "pulls the ladder up with it." In contrast, under HFC, a ladder extends down through all the fitness ranges all the way to the bottom level, where new randomly generated individuals are continually introduced and can climb the fitness ladder. Naturally, once the maximum fitness individual has been encountered, no other individuals will climb the ladder all the way to the top to displace it, but until that time, it is possible for new regions of a multimodal fitness landscape to be explored. The HFC principle was used extensively in a genetic programming system to evolve bond graphs (a type of representation for multi-domain dynamic systems) representing novel designs of microelectromechanical systems (MEMS)[21] and other GP applications. Figure 11 is an example of a MEMS filter produced by this system.

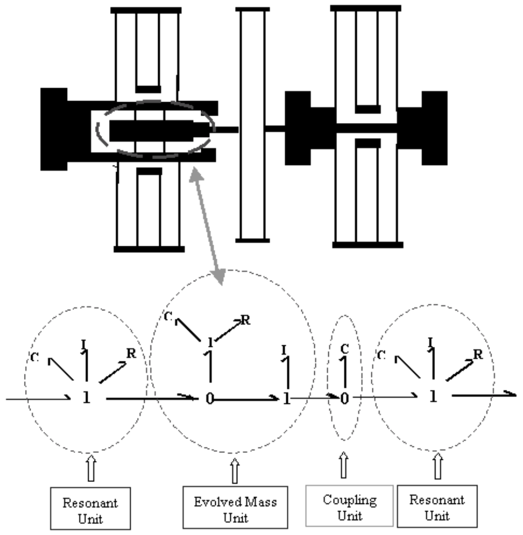


Fig. 11. A novel topology of MEM filter and its bond graph representation

All of the replacement policies described so far neglect aging of individuals—an individual in a GA does not usually have an age attribute. However, the recent Age-Layered Population Structure (ALPS)[22] method has used a surrogate for age—the average number of generations since introduction of the genetic material in an individual—and then restricted the competition for breeding such that "young" individuals need only compete against other young individuals for breeding rights, and no individual can persist indefinitely in the population. The effect is to allow exploration of the search space surrounding any innovation in an individual, but to make it increasingly likely that the individual will be replaced as the exploration continues.

3.3 Fitness Landscapes

In a classical evolutionary algorithm, including a GA, the fitness landscape is a static framework in which each genome is assigned a fixed fitness value. In some cases, the goal is a dynamic one, in which the population must attempt to continually optimize a changing value—a tracking problem of sorts. But let's consider only the case in which the objective is a single static measure with multiple constraints that must also be satisfied. This is an extremely common situation in engineering design. Even though the fitness measure is static, the fitness landscape—visualizing fitness as a height above a hypersurface depicting the values on the genome (design variables)—can be very complex. If a problem is strongly multimodal and the number of design variables and constraints is large, it is very easy for any search method beyond random search to get "stuck" in a local maximum from which it is unlikely to escape. One of the key ideas that we can steal from nature is the broad range of environments in which an organism of the same species can find itself. That means that the actual fitness of a given genotype may be very different, depending on where the organism "lands." Thus, nature offers an escape from convergence—a variety of fitness landscapes. A given mutation may be advantageous in one environment, although it is deleterious (or even fatal) in another environment. Thus, a given species in nature is evaluated in a variety of fitness landscapes, allowing natural selection to introduce diversity into the overall population by having selection operate in a variety of local environments. It is particularly easy for sexually reproducing populations to recombine parts of their genomes that conferred selective advantages in different environments to generate a new genome that can be tested in those or other environments, so long as individuals can migrate from one environment to another, at least occasionally. The example below will illustrate a design process that traverses a complex fitness landscape by making use of multiple populations having different fitness landscapes.

Another effect observed in nature can also be captured in evolutionary optimization: in nature, even individuals cloned from the same genotype display somewhat different phenotypes when subjected to different environments. Genotypes whose survival and reproduction are not as strongly affected by these phenotypic differences tend to outcompete those that are not as robust. To capture this effect, stochasticity in the design variables can be introduced, which helps to assure the robustness of the resulting design. In the example below, each design variable was subjected to a small amount of Gaussian-distributed noise, so that each time a design was evaluated—even if it was the SAME design—the values of the design variables were different by small amounts. It is easy to capture the idea of small changes in ontogeny of an individual (or implementation of a design) in this way. Some variations in the *environment* of an individual can be simulated as stochastic differences in the loading conditions in the model below. Overall, this stochasticity acts to make the resulting designs much more robust with regard to differences in loading conditions or manufacturing variation. Of course, it requires more computing resources to evaluate surviving designs more than once, but the gains in robustness of the resulting designs were well worth the additional effort.

The next example illustrates how differing fitness landscapes and phenotypic and environmental variation can be captured using evolutionary computation, to yield good, robust designs more quickly. A single-objective, multiple constraint engineering design problem often has a fitness function that describes performance desired of the design under several different loading conditions—that is, performance in several different environments. Imagine, for example, that we want to design a lower compartment rail for a new automotive chassis—the rail supports the front bumper, front suspension and motor mounts and joins with the chassis surrounding the passenger compartment. The requirement is that the rail perform well in both frontal and offset crashes. While this might be looked at as a problem with multiple objectives (for example, maximize energy absorption while minimizing weight and peak force on passengers), we can also make it a single-objective problem by setting a target weight (not to exceed) and a maximum allowable peak force, treating those as constraints. Figure 12 shows the architecture used in developing an "industrial-strength" solution to this problem[23]. From time to time, migration of designs was performed along the arrows shown, making any required adjustments in the resolution of the representation with changes in vertical levels. In some populations (left), only the frontal impact loading condition was used, and in others (right), only the offset crash conditions. Other populations (center) evaluated designs subject to both loading conditions. Vertically, near the top, only the beginning portion of the crash was simulated, and at the bottom, the entire duration was simulated. This allowed more rapid evaluation of solutions at the top, so that solutions that performed poorly at the beginning of the crash were fed to the lower populations to be simulated for longer times. Also, the evaluations at the top were done with simpler (and faster to compute with) models and less refined search spaces, to speed the "screening" of many candidate designs for their initial performances. As in nature, designs (at the top) that did not perform well were quickly killed off, whereas those that survived could migrate to the lower levels for evaluation under more stringent conditions and with more refined and

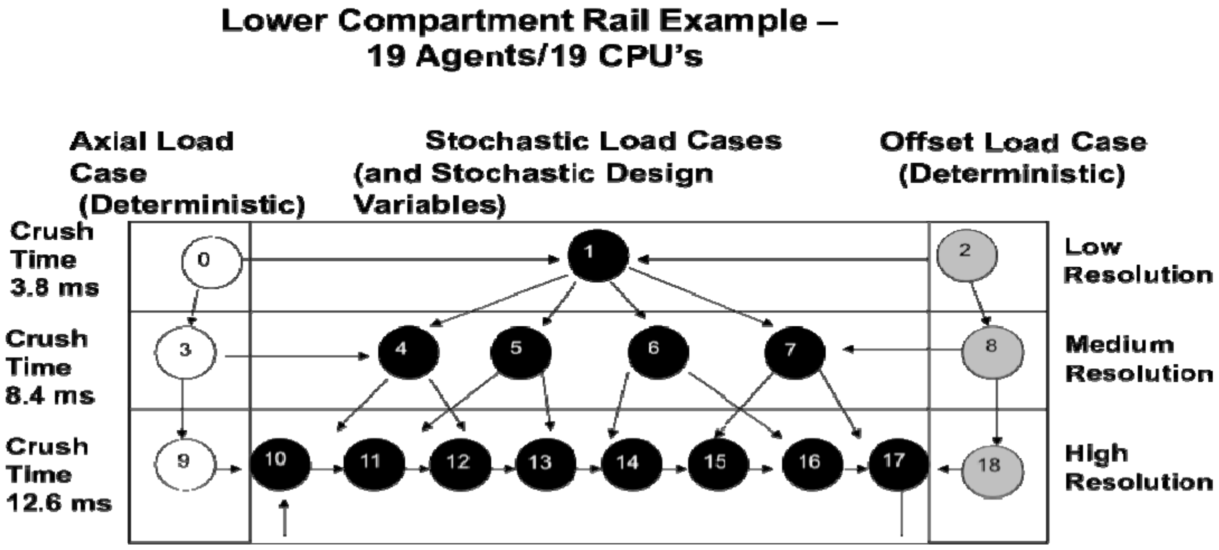

Lower Compartment Rail Example – 19 Agents/19 CPU's
Axial Load Case (Deterministic)
Stochastic Load Cases (and Stochastic Design Variables)
Offset Load Case (Deterministic)
Crush Time 3.8 ms
Crush Time 8.4 ms
Crush Time 12.6 ms
Low Resolution
Medium Resolution
High Resolution
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Figure 12. Architecture for using different fitness landscapes in different subpopulations

accurate models. The net effect was that even in 1998, it was possible to generate very innovative and high-performance solutions for a new hydroformed lower compartment rail in runs of two weeks on a 19-PC cluster. An example of a (prototype) rail design is shown in Figure 13, and Figure 14 shows the results of (a) the finite-element simulation of an

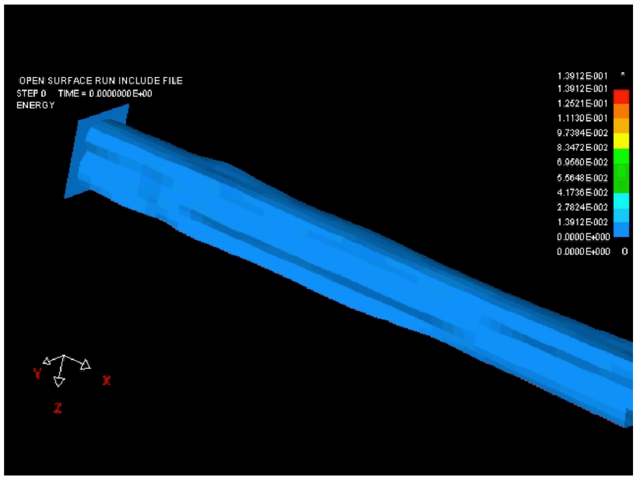
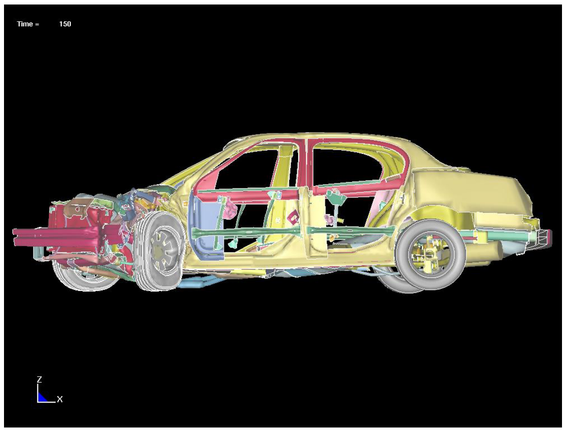
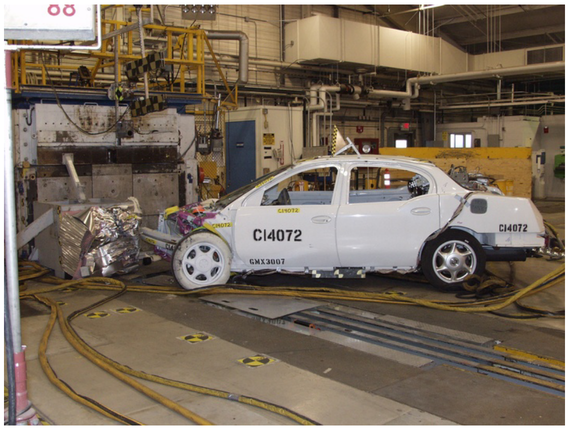


Figure 13. High-performance automotive lower compartment rail designed via evolutionary algorithm using multiple populations and different fitness landscapes, as shown in previous figure.





(a) (b)

Figure 14. (a) Finite element model of vehicle with evolved design for high-performance lower compartment rail. (b) crash-test of vehicle built with that rail design.

offset crash and (b) the vehicle after the actual offset crash test. Compared to the rail designed by the company's engineers, the GA-designed rail weighed 20% less, absorbed 100% more energy, and reduced peak force by 30%.

3.4 Digital evolution

In the previous example, multiple fitness landscapes were chosen by the researcher to be used simultaneously, with occasional migration among them. In other current work, two of BEACON's investigators, Nahum and Kerr, recently reported in a BEACON-wide meeting[24] that evolutionary exposure to different environments (exogenously or endogenously changing) can result in improved adaptation compared with only evolving in a single, unchanging environment. Indeed, it can sometimes be beneficial to switch away from a "target" environment to an alternative environment and back, in order to get improved adaptation in the target environment. Their work uses the Avida platform made available through the BEACON Center. The population is mapped onto a toroidal "checkerboard" and breeding is with local neighbors, with offspring also replacing a neighbor. However, the rewards available for solving subproblems must be shared among individuals in a neighborhood, so the fitness landscape seen by any individual is influenced by its neighbors, tending to support the evolution of diversity in the subproblems solved by the population—essentially, cooperation among the individuals in exploitation of the resource. As in nature, fitness landscapes can be different among "digital organisms," and may be changing, as induced either externally or through action of the members of the population. Any of these features may help individuals to escape from local optima in a fitness landscape to make progress toward a global optimum.

The Avida framework for digital evolution allows self-reproducing computer programs to evolve (primarily through mutation) while earning rewards for completing a variety of tasks. An individual's rewards give it more computing cycles, allowing it to replicate faster and out-compete its fellow Avidians. Mutations arise through induced errors in the process by which each Avidian makes a copy of its own genome.

Because evolution in Avida strongly resembles microbial evolution, it has been used to study many phenomena observed in microbes—most recently, in an article in *Science*[25], March, 2011. Other BEACON investigators have used the Avida platform to evolve robust mechanisms for robotics applications, including leader election and robot coordination[26,28]. However, because Avida often requires large population sizes and many generations to evolve desired results, it is not practical to substitute for genetic algorithms for many complex design problems.

3.5 Prospects for further contributions from nature

As can be seen from the examples above, when addressing very complex search tasks using evolutionary computation, it often appears helpful to structure an environment, fitness landscape, reward structure, representation, etc., such that it has spatial and/or temporal diversity, induced either exogenously or endogenously. The ideas of cooperative or competitive coevolution, seen in nature and in evolutionary computation, are mechanisms for providing an endogenously changing environment, and the concept of complexification[28] in that context provides exogenously changed representations for potential solutions. The injection island genetic algorithm[17] (iiGA) also provides a set of increasingly complex representations to enable parallel search using several different representations.

A realm of evolutionary computation of increasing interest is the so-called "evo-devo" domain, in which the solution representations on the genome are simple codes for complex expression of the phenotype in the context of the problem environment. That is, the solution "grows" much as an organism does, with a compact genetic representation generating individuals with a great many differentiated cells. Some systems—Stanley's HyperNEAT[29] and a number of other systems—are beginning to allow study in this domain, but the "killer" application of these methods remains to be discovered. It should be clear that by continuing to borrow and explore concepts from nature, evolutionary computation is developing increasingly sophisticated and powerful tools for attacking ever more complex engineering design problems.

4. A challenging application domain for evolutionary computation: metamaterials

Metamaterials are manufactured materials that display electromagnetic properties not found in nature. In particular, metamaterials may be designed to have either negative permittivity, negative permeability, or both simultaneously. These properties lead to unusual behavior of electronic systems, and have been used to explore such esoteric applications as "cloaking" and "perfect lensing." More practically, metamaterials have been used to create miniaturized antennas and novel microwave filters, and to improve the radiating properties of antennas.

Metamaterials are created by suspending small resonant structures, usually metallic, in a host dielectric medium. This can be done in a volumetric sense or by etching metal traces on a circuit board. Until recently, a few standard structures such as split rings and "S" shapes were used as the inclusions, since these have resonant properties that are easily predictable using simple theory. Unfortunately, these structures are not optimal in their response to electromagnetic fields. In particular, they have very narrow resonances, and extremely small frequency ranges over which both permittivity and permeability can be made negative.

In 2008, Diaz and Rothwell at MSU received a grant from NSF to explore ways to create novel metamaterial structures, and to employ these structures to improve the performance of electronic systems. They began by using GAs to optimize classical split-ring structures in order to miniaturize antennas. Figure 15 shows a loop antenna miniaturized by the presence of split rings. They then used both gradient search methods and GAs to create metamaterial structures with specific electromagnetic properties by optimizing on a pixellated grid. The resulting structures, such as shown in Figure 16, are quite different in appearance from classic metamaterial structures. These structures were then used to replace the split rings in the miniaturized loop antennas with the result being improved impedance performance.

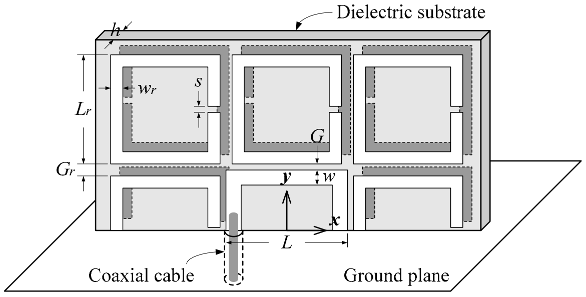


Figure 15. A loop antenna miniaturized by the presence of split rings.

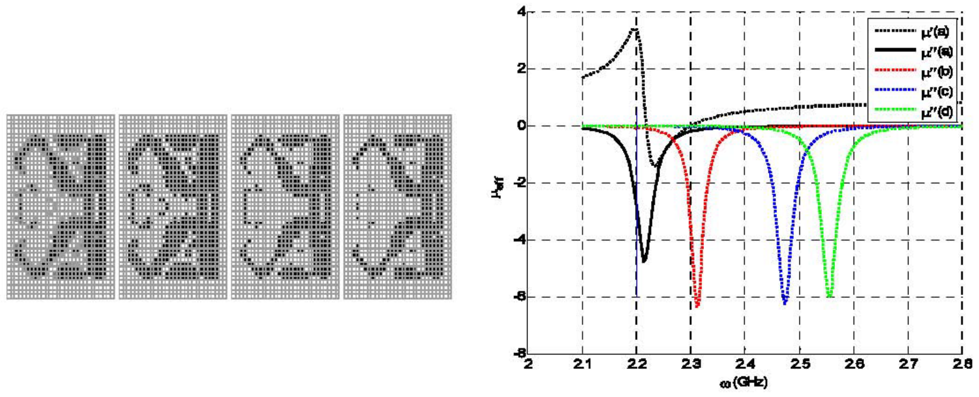


Figure 16. Metamaterial structures optimized to resonate at a specified frequency.

The next step was to optimize the performance of an antenna by altering the metamaterial structure *in situ*. The grid was placed near a loop antenna and the pixels turned on or off using a GA with the goal being to produce high return loss at a chosen frequency[30]. The resulting miniaturized antenna is shown in Figure 17, and the measured response of a prototype is shown in Figure 18. An improvement in miniaturization and performance can further be achieved by embedding the loop within the metamaterial structure as shown in Figure 19.

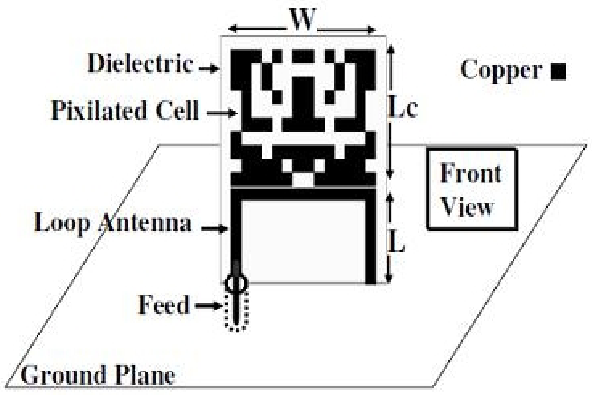


Figure 17. A metamaterial-inspired miniaturized loop antenna designed using a GA.

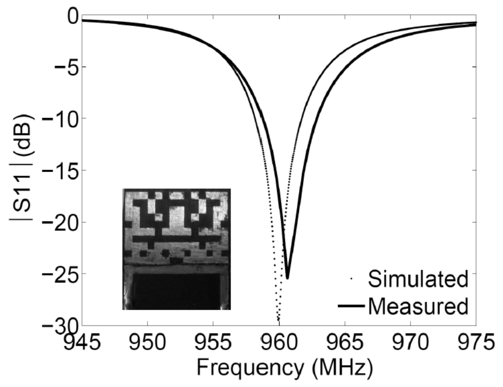


Figure 18. Performance of a metamaterial-inspired loop antenna.

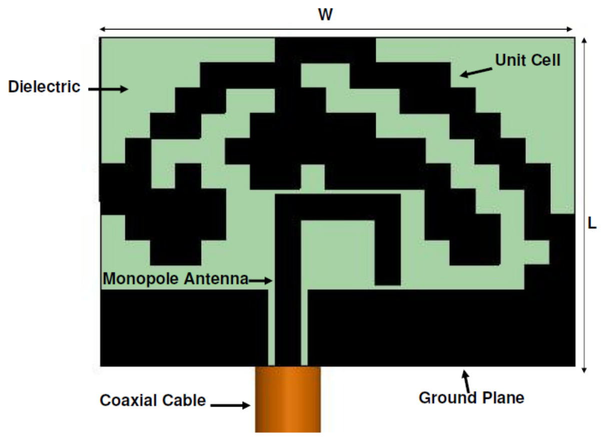


Figure 19. A loop antenna embedded within a metamaterial structure.

The metamaterial concept can be used to produce antennas with more complex radiating characteristics. Figure 20 shows a folded monopole with a sinuous structure surrounding it. The surrounding structure was designed *in situ* using a GA so as to produce dual-band radiation. This antenna can be used in the 900 MHz GSM cell phone band, and also the 1900 MHz GSM band and 2.4 GHz WIFI band. The performance of the antenna is shown in Figure 21. Perhaps the best

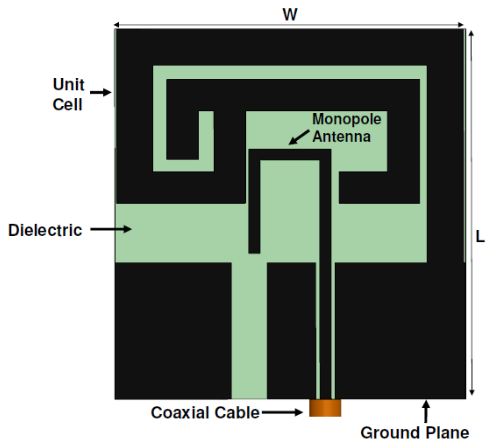


Figure 20. A dual-band folded monopole antenna based on metamaterial concepts designed using a GA.

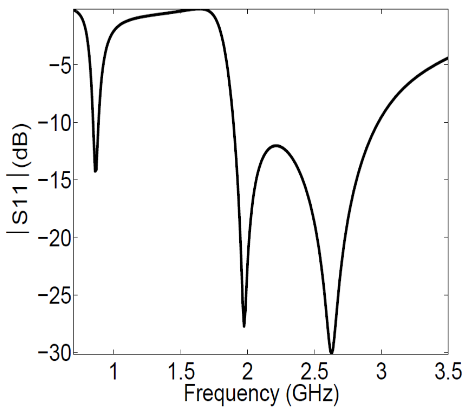


Figure 21. Performance of the dual-band folded monopole antenna

antenna performance has been achieved by placing complementary split rings beneath a circular patch antenna, as shown in Figure 22, and using a GA to optimize the rings. Figure 23 shows the performance of the antenna. Using this approach a size reduction of up to 16 in area has been achieved. Such miniaturized antennas are crucial for the future success of the wireless communications market since smaller and smaller consumer products demand wireless capabilities.

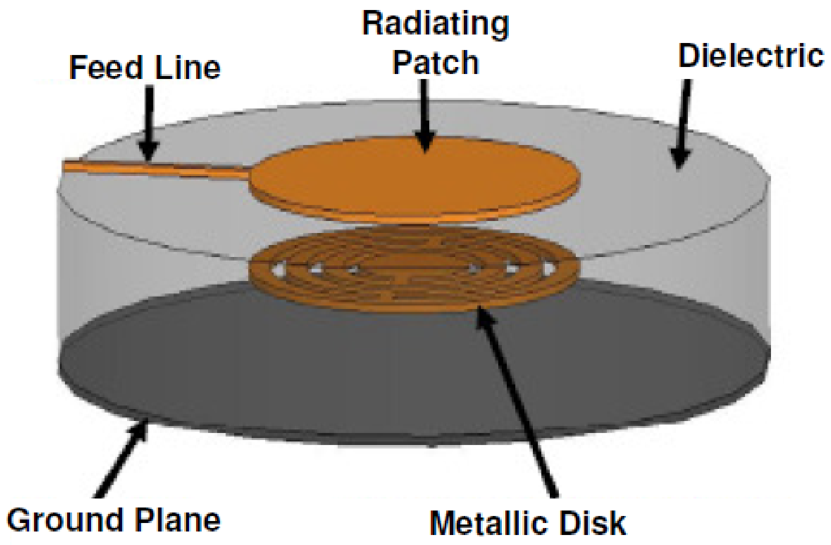


Figure 22. A miniaturized patch antenna containing complementary split-rings, optimized using a GA.

MSU researchers have applied the metamaterial concept to other RF systems. Figure 24 shows a waveguide metamaterial insert design to create an F-band waveguide bandstop filter. Figure 25 shows the simulated and measured filter responses. The metamaterial filter has high insertion loss within the wide design bandwidth, low insertion loss in adjacent bands, and sharp band edges.

4.1 Looking ahead

To date, these metamaterial optimizations have not required more sophisticated optimization methods than gradient-based search and genetic algorithms. However, future design work will seek to extend metamaterials design using the full sophistication of the nature-inspired search and optimization methods described above, and new methods yet to be developed. The mission of the BEACON Center for the Study of Evolution in Action, one of NSF's newest Science and

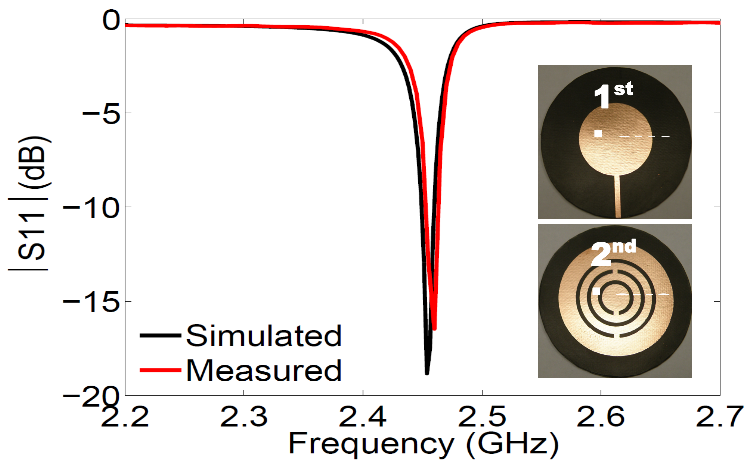


Figure 23. Performance of miniaturized patch antenna.

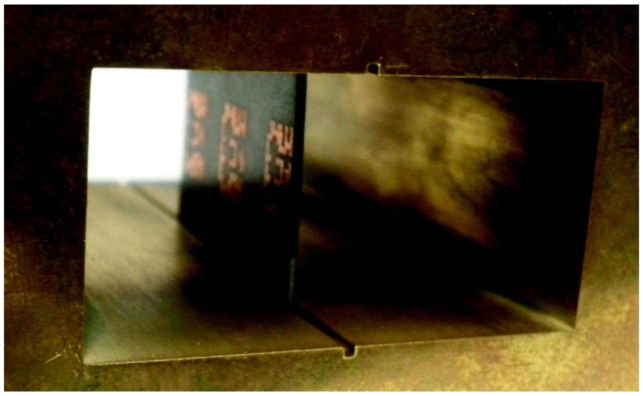


Figure 24. Waveguide with metamaterial inserts to act as a bandstop filter.

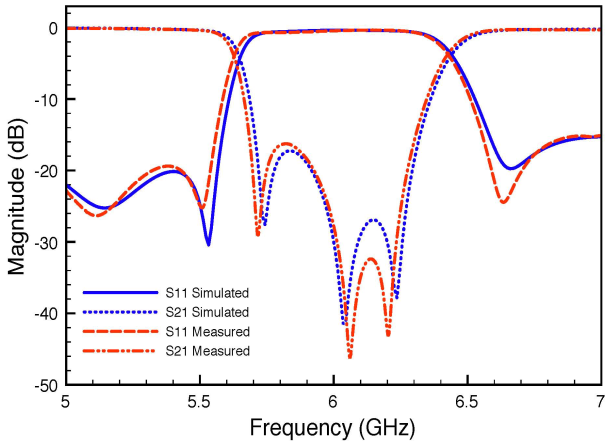


Figure 25. Performance of a metamaterial waveguide filter.

Technology Centers, includes promoting the exchange of lessons about evolution between the biological and digital domains, and applying those lessons to produce more powerful design tools.

Acknowledgment

The authors are grateful to BEACON Center for the Study of Evolution in Action, an NSF Science and Technology Center headquartered at Michigan State University. This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] Rechenberg, I., [*Cybernetic solution path of an experimental problem]*, Royal Aircraft Establishment, Library Translation 1122, Farnborough (1965).

[2] Rechenberg, I., [Evolutionsstrategie – *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution],* Fommann-Holzboog, Stuttgart (1973).

[3] Holland, J. H., [*Adaptation in Natural and Artificial Systems]*, MIT Press, Cambridge (1975).

[4] Goldberg, D., [*Genetic Algorithms in Search, Optimization and Machine Learning*], Addison-Wesley, Boston (1989).

[5] Ilavarasan, P., Rothwell, E. J., Bebermeyer, R., Chen, K.-M. and Nyquist, D. P., "Natural resonance extraction from multiple data sets using a genetic algorithm," IEEE AP-S International Symposium and URSI Radio Science Meeting, Seattle, WA (1994).

[6] Ilavarasan, P., Rothwell, E. J., Chen, K.-M., Nyquist, D. P. and Bebermeyer, R., "Natural resonance extraction from multiple data sets using a genetic algorithm," IEEE Transactions on Antennas and Propagation **43**(8), 900-904 (1995).

[7] Rothwell, E. J., Chen, K.-M., Nyquist, D. P., Ilavarasan, P., Ross, J. E., Bebermeyer, R. and Li, Q., "A general E-pulse scheme arising from the dual early-time/late-time behavior of radar scatters," IEEE Transactions on Antennas and Propagation **42**(9), 1336-1341 (1994).

[8] Rothwell, E., Chen, K.-M., Nyquist, D. P., Ross, J. and Bebermeyer, R., "Measurement and processing of scattered ultrawide-band/short-pulse signals," SPIE International Symposium on Optical Science, Engineering, and Instrumentation, San Diego, CA (1995).

[9] Li, Q., Rothwell, E. J., Chen, K.-M. and Nyquist, D. P., "Scattering center analysis of radar targets using fitting scheme and genetic algorithm," IEEE Transactions on Antennas and Propagation, vol. 44, no. 2, 198-207 (1996).

[10] Ross, J. E., Nagy, L. L. and Szostka, J., "CAD Tools for Vehicular Antennas," 1999 IX National Symposium of Radio Science, Poznan, Poland (1999).

[11] Ross, J., "Design of Receive Antennas for DTV", Society of Broadcast Engineers, 37th Annual SBE Chapter 22 Expo, Pittsburgh, PA (2008).

[12] Schneider, R. and Ross, J., "Antennas for the New Airwaves," IEEE Spectrum Magazine, 44-49 (2009).

[13] Coleman, C. M., Rothwell, E. J. and Ross, J. E., "Self-structuring antennas," IEEE AP-S International Symposium and URSI Radio Science Meeting, Salt Lake City, Utah (2000).

[14] Coleman, C. M., Rothwell, E. J., Ross, J. E. and Nagy, L. L., "Self-Structuring Antennas," IEEE Antennas and Propagation Magazine **44**(3), 11-22 (2002).

[15] Ozdemir, T., "A Polarization Diversity Multi-Beam Antenna for ZigBee Applications," WAMICON 2009, Clearwater, FL (2009).

[16] Özdemir, T., Goykhman, Y., Oberdier, L. and Lynch, J., "Smart antenna technology for structural health monitoring applications," Nondestructive Characterization for Composite Materials, Aerospace Engineering, Civil Infrastructure, and Homeland Security 2010. Edited by Shull, Peter J.; Diaz, Aaron A.; Wu, H. Felix. Proc. SPIE, 7649, 76490M-76490M-7 (2010).

[17] Lin, S.-C., Punch, W. and Goodman. E., "Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach," *IEEE Conf. on Parallel and Distrib. Processing,* (1994).

[18] Tanese, R. "Distributed Genetic Algorithms." *Proceedings 3rd Internat. Conf. on Genetic Algorithms*, Morgan Kaufmann, San Francisco, 434-439 (1989).

[19] Schwefel, H.-P., [*Numerical Optimization of Computer Models]*, John Wiley & Sons (1981).

[20] Hu, J., Goodman, E., Seo, K., Fan, Z. and Rosenberg, R., "The Hierarchical Fair Competition (HFC) Framework for Sustainable Evolutionary Algorithms", *Evolutionary Computation* **13**(2), (2005).

[21] Fan, Z. Seo, K., Hu, J., Rosenberg, R. and Goodman, E., "System-Level Synthesis of MEMS via Genetic Programming and Bond Graphs," Proc. 2003 Genetic and Evolutionary Computing Conference (GECCO 2003), Chicago, Springer, Lecture Notes in Computer Science, 2058-2071 (2003).

[22] Hornby, G. S. "ALPS: The Age-Layered Population Structure for Reducing the Problem of Premature Convergence," *Proc. Genetic and Evolutionary Computation Conference* (GECCO-2006), ACM (2006).

[23] Eby, D., Averill, R. and Goodman, E.. "How Well Can It Take a Hit?" *Mechanical Engineering Design,* American Society of Mechanical Engineers, New York, 26-28 (2001).

[24] Nahum, J. and Kerr, B. "*Pers. Comm. with E. Goodman,"* April 23 (2011).

[25] Woods, R. J., Barrick, J. E., Cooper, T. F., Shrestha, U., Kauth, M. R. and R. E. Lenski, R., "Second-order selection for evolvability in a large *Escherichia coli* population," *Science* **331**(6023), 1433-1436 (2011).

[26] Knoester, D. B. and McKinley, P. K., "Evolution of Synchronization and Desynchronization in Digital Organisms," *Artificial Life* **17**(1), 1-20 (2011)

[27] McKinley, P. K., Cheng, B. H. C., Ofria, C., Knoester, D., Beckmann, B. and Goldsby, H., "Harnessing Digital Evolution," *IEEE Computer* **41**(1), 54-63 (2008).

[28] Stanley, K. and Miikkulainen, R. "Competitive Coevolution through Evolutionary Complexification." *Journal of Artificial Intelligence Research*, **21**, 63-100 (2004).

[29] Gauci, J., Stanley, K. "Generating Large-Scale Neural Networks Through Discovering Geometric Regularities," *Proc. Genetic and Evolutionary Computation Conference (GECCO 2007),* ACM, New York (2007).

[30] Ouedraogo, R., Rothwell, E., Diaz, A., Chen, S.-Y., Temme, A. and Fuchi, K., "In Situ Optimization of Metamaterial-Inspired Loop Antennas," IEEE Antennas and Wireless Propagation Letters, **9**, 75-78 (2010).