Using Evolutionary Algorithms to Search for Control Parameters in a Nonlinear Partial Differential Equation

Corresponding author: Rogene M. Eichler West

R.M. Eichler West[1][2][3], E. De Schutter[1], G.L. Wilcox[3]
1. Born-Bunge Foundation, University of Antwerp - UIA, B2610 Antwerp, Belgium
2. Division of Biology, 216-76 , Caltech, Pasadena, CA, 91125 USA
3. Minnesota Supercomputer Institute, University of Minnesota, MPLS, MN, 55455 USA

Abstract

Many physical systems of interest to scientists and engineers can be modeled using a partial differential equation extended along the dimensions of time and space. These equations are typically nonlinear with real-valued parameters that control the classes of behaviors that the model is able to produce. Unfortunately, these control parameters are often difficult to measure in the physical system. Consequently, the first task in developing a model is usually to search for appropriate parameter values. In a high dimensional system, this task potentially requires a prohibitive number of evaluations and it may be impossible or inappropriate to select a unique solution.

We have applied evolutionary algorithms (EAs) to the problem of parameter selection in models of biologically realistic neurons. Our objective was not to find the "best" solution, but rather we sought to produce the manifold of high fitness solutions that best accounts for biological variability. The search space was high dimensional (> 100) and each function evaluation required from one minute to several hours of CPU time on high performance computers. Using this model and our goals as an example, we will: 1) review the problem from the neuroscience perspective; 2) discuss high performance computing aspects of the problem; 3) examine the suitability of EAs for the efficient optimization of this class of problems; and 4) describe and justify the specific EA implementation used to solve this problem.

1. The Problem in Neuroscience

Neurons are the fundamental units responsible for the transmittal and transduction of information in the brain. Information is transmitted between neurons via diffusable chemicals. The chemical signals from many convergent presynaptic neurons are spatiotemporally integrated and transduced into an electrical response by the postsynaptic neuron (Figure

1a). If the electrical response is sufficiently large, then the postsynaptic neuron will propagate a momentary event and release its own chemical transmitters, carrying the signal to the next set of neurons in the network. These suprathreshold electrical events are referred to as action potentials, bursts, or spikes.

How information is encoded in the rate and pattern of electrical signaling events in single neurons and populations of neurons is currently the subject of hot debate among neuroscientists (Ferster and Spruston 1995). Interpretations of the *neural code* have been proposed for many systems, for example: the representation of faces in monkeys (Abbott *et al* 1996); the force exerted in a voluntary motor behavior (Monster and Chan 1977); sensory information in the cricket cercal system (Theunissen *et al* 1996); the representation of odors (Laurent 1996); and cognitive encoding of direction in the motor cortex (Georgopoulos *et al* 1993). However, it is fair to say that at this time, *we don't really know how neurons encode information* and the answer to this question is at the heart of understanding brain function.

Spike production in neurons is governed by a class of membrane-spanning proteins known as voltage-gated and/or calcium-dependent (VGCD) channels (Figure 1b). These channels contribute in a nonlinear manner to the electrical response of the neuron by varying their ionic conductances in response to the voltage and/or calcium concentration. A large variety of VGCD channels exist, and they contribute substantially to the electrical properties of the neuron (Johnston *et al* 1996, Spruston *et al* 1994, Hille 1992). The contribution of VGCD channel subtypes to information encoding varies adaptively, however. Intracellular signaling compounds, whose concentrations are elevated in response to activity, functionally modulate the kinetics and maximum conductances of VGCD channels (Perezreyes and Schneider 1994, Kallen *et al* 1993, Li *et al* 1993, Toro and Stefani 1991, Numann *et al* 1991, Chetkovich *et al* 1991). A significant body of work in Aplysia suggests that some forms of animal learning can be attributed to the activity-dependent modulations of VGCD channels (Klein and Kandel 1978, Fitzgerald *et al* 1990, Edmonds *et al* 1990). Therefore, there are many combinations of VGCD channel dynamics that are congruent with the "normal" functioning of neurons. However, specific classes of combinations may be required for or achieved during information processing associated with certain cognitive or behavioral states.
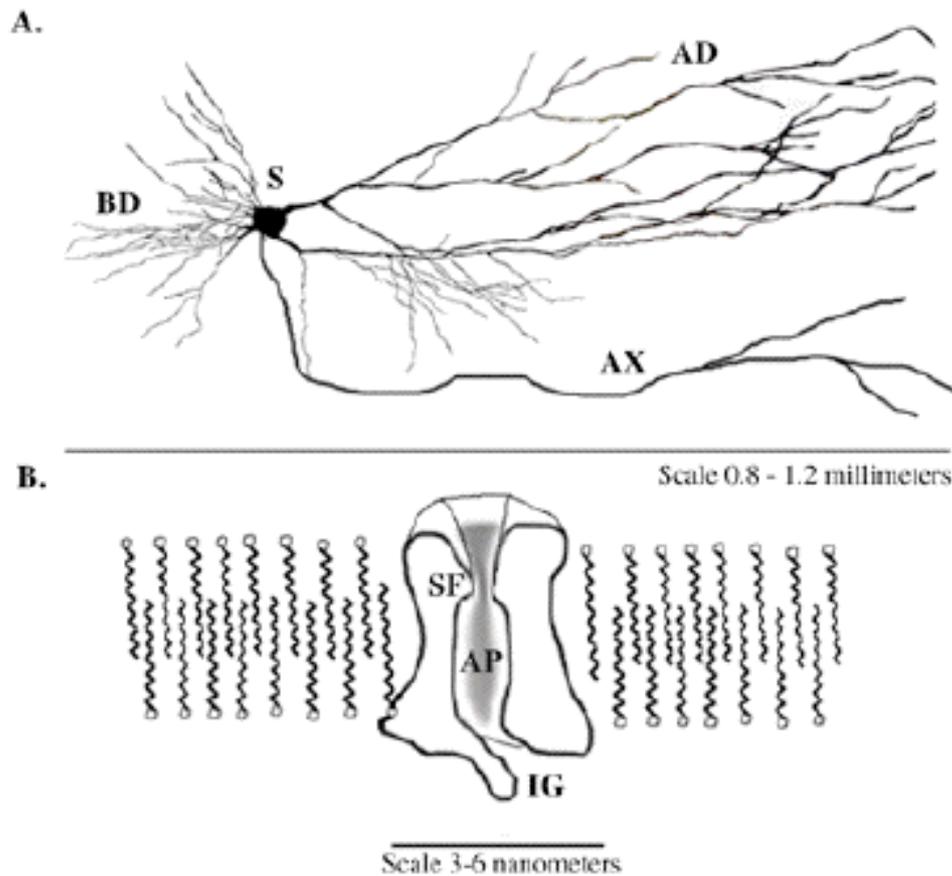
Figure 1

(**A**) CA3 hippocampal neuron morphology digitized from an experimental preparation. Presynaptic neurons transmit chemical signals onto receptor sites in the basal and apical dendrites. If the spatiotemporally propagated electrical signal becomes sufficiently depolarized in the region where the axon extends away from the soma (the axon hillock), action potentials are generated and transmitted via the axon to the next set of neurons in the network. *S, soma; BD, basal dendrites; AD, apical dendrites; AX, axon.* (**B**) VGCD channels are distributed throughout the lipid bilayer membrane of the neuron. Electrochemical conditions induce conformational changes in the three dimensional structure of the protein channel. Some conformations are more conductive to the flow of ions by, for example, releasing an inactivation gate blocking the aqueous pore. A region of charged animo acid residues at the mouth of the channel serves as a filter for ions of a particular charge or size. *AP, aqueous pore; SF, selectivity filter; IG, inactivation gate.*

To address the extent of VGCD channel influence on information processing, we need a substantial amount of information about the types and numbers of VGCD channels present in a given class of neurons. We need to know the kinetics of the channel conductances with respect to their voltage and calcium sensitivities. We need to know their distribution along the spatial extent of the neuron and how these distributions may change over time. Finally, we need tools to manipulate the channel distributions, conductances and/or kinetics so that we can observe how channel modulations affect the behavior of the entire neuron.

Substantial advances have been made over the past decade in the tools and

techniques available for the measurement and manipulation of VGCD channels and currents. These methods include single channel recording (Sakmann and Neher 1983, Wonderlin *et al* 1990), voltage-sensitive dyes (Ebner and Chen 1995), optical imaging of calcium ion concentrations (Denk *et al* 1990), and localization of channels via immunohistochemistry or autoradiography (Rhodes *et al* 1996, Maletic-Savatic *et al* 1995). However, we still don't have the ability to experimentally identify and control VGCD channels in a precise manner. Spatial limitations prevent the measurement of individual channels on the smaller dendritic processes. Channel localization methods are limited by the specificity of the marker compounds and the resolution of the optical signal (Kuhar and Unnerstall 1985). At this point in time, it is not possible to manipulate the spatial expression of channels nor is it possible to impose a specific localized change in the conductance of a single channel type.

As an alternative approach, biophysically realistic computational models of neurons (Segev *et al*. 1989, De Schutter and Bower 1994) allow observation of the state variables, such as voltage and calcium concentration, and control of the scaling parameters, such as the kinetics of channels and their spatial distributions, with a precision unattainable in experimental preparations. The past decade has seen the development of several public domain neural simulators (De Schutter 1992). The power of modeling and ease of model implementation are persuasive arguments for studying the influence that VGCD channels can have on the bioelectrical behaviors of neurons within a computational framework.

The outstanding problem in the development of these models has been to determine parameter sets that represent the spatial distributions of the VGCD channels. Models of neurons typically have from tens to thousands of unconstrained parameters. Because these models exist in high dimensional spaces, it is not possible to evaluate every combination of plausible parameters. For example, to enumeratively search through each combination of 100 binary parameters $\{0,1\}$, a total of 2^{100}, or 10^{30}, simulations must be performed. If each simulation requires one second to be run and evaluated, $4 * 10^{20}$ centuries would be required to complete this evaluation [1].

We would like to know ALL the parameter sets that produce behaviors appropriate to a given electrical signaling behavior. Such an encompassing manifold, or *boundary enclosing a volume of solutions*, captures the reality of biology-like diversity. We would like to examine the relative ranges of each parameter manifold as a measure of sensitivity of the model along each parameter axis. Finally, we would like an efficient method of producing these solutions.

2. High Performance Computing Aspects of the Problem

Because this IMA volume and associated workshop were intended to focus on the high performance computing (HPC) aspects of EA implementations, it seems reasonable to briefly review the computational structure of the model. This will additionally help the reader to better understand the source and the

significance of the parameters we would like to optimize. Our model is structurally similar to systems in other HPC fields such as fluid dynamics and geophysics. In many ways, the process and goals of model building are shared across HPC applications. We have a computational model of a physical process. We have many unconstrained parameters in our description. We want the behaviors, e.g. the time series, from our simulated model to correspond well with data collected from the physical system. We would like to interpret parameter sets that yield good correspondences to validate the appropriateness of the model or to make predictions about the physical system. Our approach should therefore be generalizable to a number of HPC endeavors.

The physics of electrical behaviors in biophysically realistic neurons is described by the cable equation (Rall 1989). The cable equation is a second order partial differential equation (*PDE*) of the parabolic type. It is analytically similar to the heat or diffusion equations. The equation describes voltage as a function of space and time.

$$\frac{\partial^2 V}{\partial X^2} - V - \frac{\partial V}{\partial T} = 0 \qquad (1)$$

The voltage term, *V*, represents the transmembrane voltage difference as a deviation from the resting value. *X* and *T* represent the dimensions of distance and time over a specified domain. The Method of Lines replaces the *PDE* problem by a mathematically equivalent system of ordinary differential equations (*ODE*) by discretizing the spatial domain. Within each local domain, or spatial compartment, kinetic equations such as those describing VGCD conductances can be incorporated. This resulting *ODE* system can be subsequently solved using numerical integration. The derivation follows.
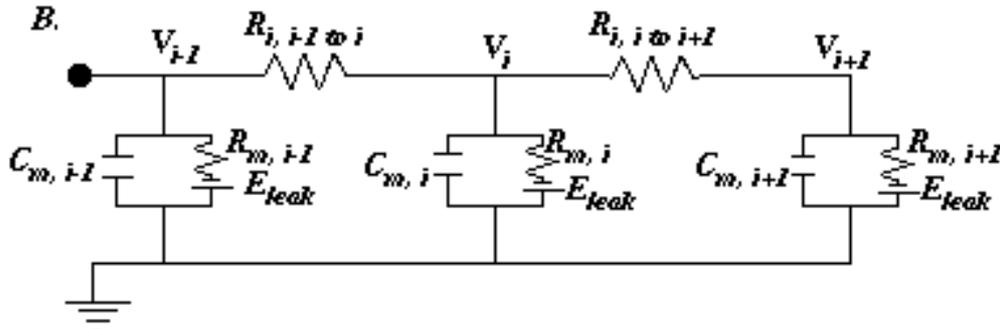
A.
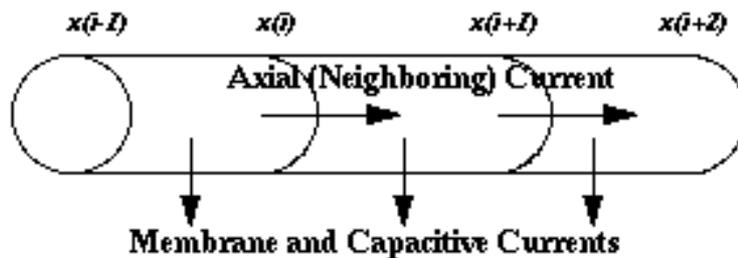
B.

Figure 2

(**A**) A length of a neuronal process is discretized into three cables that are spatially

adjacent to each other and coupled through a linear resistance term.

(**B**) The electric circuit diagram for three electrical compartments has an axial resistance term coupling the compartments. Each region of membrane in this figure has two sources of local current: membrane (passive leak) and capacitive.

The following substitutions

$$X = \frac{x}{\lambda} \qquad (2)$$

$$T = \frac{t}{\tau} \qquad (3)$$

produce a form of the cable equation that incorporates biological parameters.

$$\lambda^2 \frac{\partial^2 V}{\partial x^2} - V - \tau \frac{\partial V}{\partial t} = 0 \qquad (4)$$

x is the distance along the axis of a membrane cylinder (cm) and t is time (ms). λ (cm) and τ (ms) are scaling metrics for the electrical properties of the system. Known as the length constant and the time constant, respectively, they are metrics for the distance or time required for the voltage to attenuate to 1/e of the initial value when a deviation from the resting value is applied. They are defined in terms of the resistance and capacitance per unit length or area.

$$\tau = R_m C_m \qquad (5)$$

$$\lambda = \sqrt{\frac{r R_m}{2 R_i}} \qquad (6)$$

R_m is the specific membrane resistance (Ωcm^2), C_m is the specific membrane capacitance ($\mu F/cm^2$), R_1 is the axial resistance (Ωcm), and r is the radius of the cylinder (cm).

The *PDE* is replaced by a vector equation of N *ODEs* through the approximation of the second order spatial derivative with a central differences expression. The number of compartments[2], N, depends on the number of discretized elements required for the system to numerically converge. The number of adjacent elements is determined by the specific morphological representation of a given compartment [3]. Equation 7 describes the Laplacian for an element with two neighboring compartments.

$$\frac{\partial^2 V_i}{\partial x^2} \cong \frac{(\frac{\partial V_{i+1/2}}{\partial x}) - (\frac{\partial V_{i-1/2}}{\partial x})}{(\Delta x_{i+1/2,i-1/2})} \cong \frac{\frac{(V_{i+1} - V_i)}{\Delta x_{i+1,i}} - \frac{(V_i - V_{i-1})}{\Delta x_{i,i-1}}}{(\Delta x_{i+1/2,i-1/2})} \ ; \ i, i+1, i-1 \in N \qquad (7)$$

To clarify the notation, consider the simplifying assumption that all compartments have uniform radius and length. Rearrangement and substitution yields:

$$C_m \frac{\partial V_i}{\partial t} = \frac{r_i}{2 R_i} (\frac{(V_{i+1} + V_{i-1} - 2 V_i)}{(\Delta x^2)}) - \frac{V_i}{R_m} \qquad (8)$$

The axial (neighbor) and membrane currents are defined by application of Ohm's Law.

$$I_{neigh} = \frac{r_i}{2 R_i} (\frac{(V_{i+1} + V_{i-1} - 2 V_i)}{(\Delta x^2)}) \qquad (9)$$

$$I_{mem} = -\frac{V_i}{R_m} = -(G_{mem}V_i) \qquad (10)$$

Thus by substitution, the single compartment *ODE* can be described by three current sources; capacitive, neighboring (axial), and membrane.

$$\frac{\partial V_i}{\partial t} = -(\frac{I_{neigh} + I_{mem}}{C_m}) \qquad (11)$$

It is generally assumed that C_m and R_1 are constant for all compartments in the model, while R_m is often varied (Holmes and Rall 1992). The capacitance of the membrane is typically estimated to be a constant value of 1.0 µF/cm^2 (Hodgkin and Rushton 1946).

The voltages representing spatially adjacent compartments in the I_{neigh} term *couple* the system of N *ODEs*. This co-dependency requires that the entire system be solved simultaneously as a vector-matrix equation. The matrix is structurally symmetric, although not necessarily numerically symmetric when compartments of nonuniform physical dimensions are allowed. The matrix is sparse and constant (with respect to the location of zeros) for a given discretized morphology (Hines 1984, Mascagni 1989, Eichler West and Wilcox 1996). Numerical methods optimized for sparse matrices, such as the minimal ordering method, eliminate unnecessary operations on zero-valued matrix elements (Press *et al* 1992).

I_{mem} is composed of several current sources: injected, passive leak, and VGCD channel currents.

$$I_{mem} = I_{stim} + I_{leak} + I_{VGCD} \qquad (12)$$

Any carrier, pore, or mechanism for transferring ions across the membrane is considered a subset of this current type. Injected current, I_{stim}, is a driving term typically applied to the soma compartment. This term models microelectrode current injections during biological experiments. The membrane leak current, I_{leak}, stabilizes the voltage near the resting potential.

$$I_{leak} = G_{leak}(V - E_{leak}) \qquad (13)$$

G_{leak} is the conductance (µS) and E_{leak} is the reversal potential (mV) determined by the Nernst potential of the permeable ion species. G_{leak} and E_{leak} are typically constant-valued. I_{VGCD} is the sum of the currents produced by multipleVGCD channel types. Using the classic Hodgkin and Huxley model (1952) as an example, we would include currents representing the fast sodium (INa) and potassium delayed rectifier (IK(DR)) channels in the I_{VGCD} expression.

$$I_{VGCD} = I_{Na} + I_{K(DR)} \qquad (14)$$

The conductances of the VGCD channel subtypes are functions of voltage, calcium concentration, and/or time. Ion channels are assumed to exist in either an open (conductive) or closed state. Using the potassium delayed rectifier parameter as an example, n represents the normalized fraction of the population that is in an open state, α is the rate at which the channels open, and β is the rate at which the channels close. The rate constants α_n and β_n are functions of voltage and/or calcium concentration [4].

$$open \ (n) \ \underset{\beta_n}{\overset{\alpha_n}{\rightleftharpoons}} \ closed \ (1-n) \qquad (15)$$
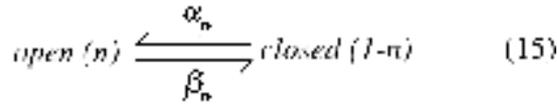
From this relationship, the first order kinetic equation for the open state is:

$$\frac{dn}{dt} = -\beta_n n + \alpha_n(1-n) = -(\alpha_n + \beta_n)n + \alpha_n \qquad (16)$$

To produce the current term for a given channel subtype, n is multiplied by a conductance scaling factor (G_n in μS) and the driving potential for the ionic species permeable to that channel ($V-E_n$ in mV). The driving potential, determined in part by the concentration gradient of the major ion species, favors inward currents through sodium and calcium permeable channels and outward currents through the potassium permeable channels.

$$I_n = G_n n(V - E_n) \qquad (17)$$

The conductance scaling factors, G_n, are typically the unknown parameters in compartmental models. *As described in the next sections, EA strategies can be used to determine appropriate values for these parameters.*

Many channel types have more complicated kinetics due to the existence of multiple subconductance states (*m*, eqn. 18; *n*, eqn. 19) and/or inactivation gates (*h*, eqn. 19). For example, the Hodgkin and Huxley (1952) sodium and potassium delayed rectifier currents are defined as:

$$I_{Na} = G_{Na} m^3 h(V - E_{Na}) \qquad (18)$$

$$I_{K(DR)} = G_{K(DR)} n^4(V - E_K) \qquad (19)$$

It should be noted that the classic biophysical kinetic description of Hodgkin-Huxley is only an approximation of the ensemble behavior of channels (Hille 1992, Keynes 1992). Further, this model may be inappropriate to describe the behavior of some channel types (Goldman 1943).

The state parameter *ODEs* (eqn. 16) must be solved concurrently with the voltage ODEs (eqn 11); however, the state parameter equations depend only on the local values of voltage and/or calcium. Inclusion of these current sources potentially increases the problem size substantially. For the Hodgkin-Huxley (1952) model with N compartments, 4N equations (m,h,n,V) must be solved simultaneously. The Traub (1991) CA3 hippocampal neuron model, which we have used in the modeling work presented in this paper, contains 11N equations. This model uses a simplified 19 compartment representation of a neuron for a total of 209 ODEs and 114 unknown conductance scaling factors. The De Schutter-Bower (1994) cerebellar Purkinje cell model contains many more channel types and an experimentally-derived morphology for a total of 32,000 ODEs and 19,200 unknown conductance scaling factors. The trend in neuroscientific modeling is to build larger models that include more experimental detail. Thus, the need to determine appropriate areas of parameter space will only continue to grow.

The Traub (1991) model offered many advantages for a proof-of-concept EA experiment; each simulation required minimal computer resources and the model expressed a range of characteristic hippocampal neuron behaviors. A

depolarizing afterpotential (DAP) was produced after a stimulus subthreshold for burst elicitation and suprathreshold bursts were followed by afterhyperpolarizations (AHPs). The model demonstrated a transition between single spiking to bursting regimes that corresponded well with experimentally-observed frequency-intensity responses. Finally, a network of these neurons was able to reproduce picrotoxin-induced synchronized multiple bursts and has been since refined as a model of gamma-frequency EEG activity (Traub *et al* 1996).

Traub's model reduced the morphology of a CA3 hippocampal neuron to a 19 compartment equivalent cylinder (Rall 1962). The passive membrane properties, $Rm = 10,000 \, \Omega cm^2$ and $Cm = 3.0 \, \mu F/cm^2$, yield a time constant [5] of 30 ms consistent with experimental observations (Turner and Schwartzkroin 1980, Turner 1984). The test stimulus was a 0.3 nA depolarizing current injection into the soma compartment because Traub described physiologically interesting behaviors produced by this stimulus. The model defined kinetic equations for six channel conductances within each compartment: a sodium channel, a hybrid high threshold calcium channel, a potassium delayed rectifier channel, a potassium A channel, a potassium AHP channel, and a calcium-dependent potassium channel.

The EA in this study searched the 114 dimensional space containing the six channel conductances in each of the 19 compartments. Traub selected his parameters by trial-and-error. One consequence of this approach was the predominant use of arbitrary zeros and the report of a single "best" solution of the search rather than a manifold. We wanted to demonstrate the ability of an EA to automatically select a volume of solutions that either included the parameter set of Traub or demonstrated an improved correspondence to experimental observations in comparison to Traub's parameter choice.

We defined two low dimensional (single value) metrics to characterize the parameter space for the purposes of post-analysis and initialization of the random seed parameters. The *total conductance* is the sum of all conductances over all compartments and the *excitatory/inhibitory conductance ratio* (EIR) is the sum of all excitatory (inward) conductances over all compartments divided by the sum of all inhibitory (outward) conductances over all compartments.

3. Why use Evolutionary Algorithms?

Evolutionary Algorithms (EAs) represent an efficient and robust method for parameter fitting in high dimensional spaces (Holland 1975, Forrest 1993). This algorithm has been successful when applied to difficult optimization problems: for example, in job shop scheduling (Whitley *et al* 1995), high-performance network design (Davis *et al* 1993), DNA fragment assembly (Parsons *et al* 1995), and models of the earth's mantle viscosity (Kido *et al* 1996). A statistical proof called the Schema Theorem guarantees improvement of the search over time (Holland 1975).

EAs have not previously been applied to the field of compartmental neuronal modeling, although reduced dimensional parameter searches using systematic (Bhalla and Bower 1993) or stochastic (Foster *et al* 1993) search

methods have been reported. Would these or methods other than an EA perform better? Descriptions of many optimization methods are available both conceptually in textbooks and algorithmically in computer executable codes obtainable from public domain sites on the Internet. These methods include hill climbing, simulated annealing, neural networks, genetic algorithms, local search, and heuristic search. With the large number of methods and their variations available, how does one choose the best for one's application?

Wolpert and Macready (1995) propose that there are "No Free Lunches" (NFL) for effective optimization. Using theoretical arguments, they claim that all search algorithms perform exactly the same when averaged over all cost functions. In other words, the set of functions for which algorithm A outperforms algorithm B is complementary to the set for which B outperforms A. Furthermore, observing how well an algorithm has done so far in a search will tell us little about how well it will continue to do. Consequently, it is a flawed strategy to choose a method by comparing initial algorithmic performance and then to search completely based on these preliminary results. Wolpert and Macready warn that if one does not incorporate any information about the function into the algorithm, then success must rely on a fortuitous matching between the features of function and the search path determined by the algorithm.

The NFL paper produced a great deal of discussion because the implication that choosing a successful method for a given application may be akin to the luck of a coin flip stands in contrast to empirical experience. Culberson (1996) extended the NFL theorem with a corollary proposing that *all algorithms perform the same as a random enumerative search when the method is blind*. These assertions suggest that we will not likely find a canned method to solve our optimization problem. A hand-tuned method is required, taking into account as much information about the function and solution space as we know.

The need to incorporate domain knowledge into a search is not newly revealed by the NFL-related reports. Based on empirical studies, Davis (1991) suggested that the difficulty of search is not a property of a given problem, but rather depends on the relationship between the problem and the algorithm. Using formal language theory, Hart and Belew (1991) also conclude that analyses of optimization methods (in particular, a GA), which do not specify the class of functions being analyzed, can make few claims regarding the efficiency for an arbitrary function. Specifically, they state: "These results suggest that future analyses of the Genetic Algorithm must pay close attention to the relationship between the algorithmic parameters of the GA and the function space from which the fitness function is selected. Since any fixed set of algorithmic parameters cannot enable the GA to efficiently optimize an arbitrary function in a broad class like the one we have considered, we must consider smaller classes or distributions of functions for which those parameters are most appropriate. Alternatively, if we have a function we wish to optimize, we must carefully select our algorithmic parameters if we wish to optimize the function efficiently with the Genetic

Algorithm."

This statement implies that there is a *specific* implementation of an algorithm that will be effective for each class [6] of functions. For an algorithm to be effective, the biases in how search space is explored must correlate well with salient features in the cost function. The problem is, of course, to know what is salient about the function when the whole point of the search is that little is known! (Otherwise, why would one be searching over it?)

To optimize the parameters for the compartmental neuronal model, one must first choose a method that sounds reasonable, gain some intuition into how the optimization algorithm works, and attempt to modify it with domain-based knowledge. Evolutionary Algorithms (EAs) are a popular and successful parameter optimization technique inspired by biological principles. Armed only with intuition and a predisposition towards this technique because of the familiarity of the biological metaphor, our initial optimization efforts focused on understanding the applicability of EAs for the compartmental neuron channel distribution problem.

4. Designing an EA Strategy

The specific mechanisms of selection and recombination applied to a particular problem space determine the success of an EA. Selection identifies and retains favorable parameter combinations. Crossover enhances the search mechanism through the breakup and recombination of parameter combinations, maximizing a good balance between these two extremes. Thus, the representation of the problem along the string and the type of crossover function applied are important factors in the rate of success of a search because probability favors the breakup of longer parameter combinations. In the following sections, we will discuss and justify our implementation with respect to: 1) representation; 2) recombination; 3) selection; 4) initialization; and 5) termination criteria.

Representation

The representation of a problem is the mapping between problem space and the data space operated on by the algorithm. Three aspects of representation must be considered for an EA application: 1) embedding information onto string values; 2) ordering values onto the linear array of the string; and 3) choosing the cardinality of the alphabet (binary vs. real-valued). These aspects should be examined with respect to the choice of recombination operators one intends to apply.

We considered two embedding spaces for our problem: 1) a string of values representing each of the individual parameters; and 2) a string of scaled functions which could be applied over the normalized distance from the soma. We use the soma compartment as a spatial reference point in our system because most experimental recordings are made in the soma. However, other PDE systems might consider the utility of applying scaled functions of the normalized length of the spatial domain along one or more dimensions.
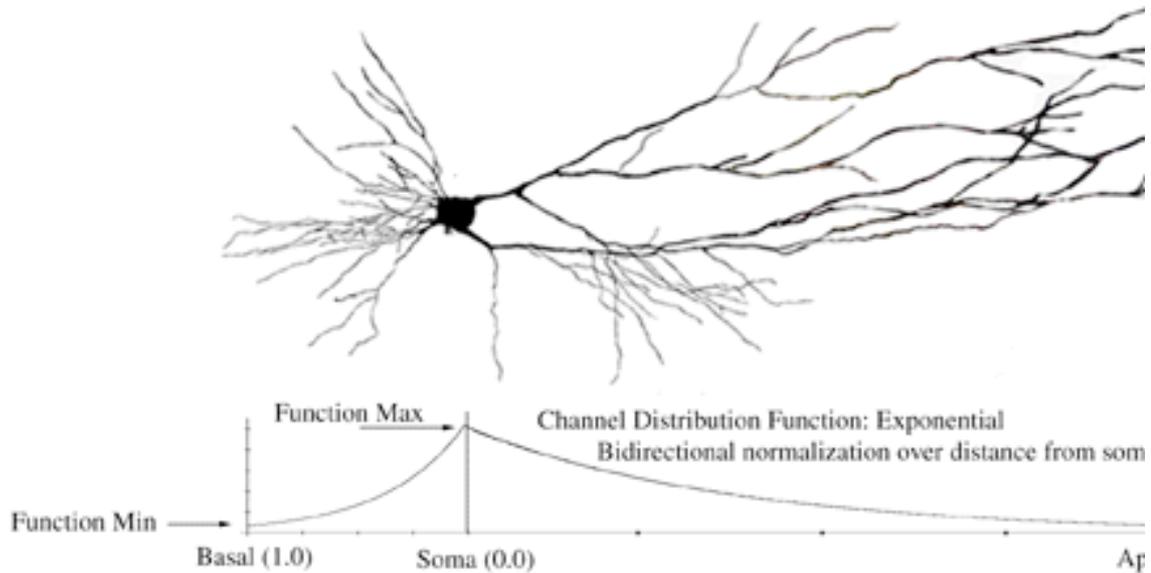
Figure 5

Only three parameters are needed to encode channel distributions as functions: function type (ex. linear, exponential, sinusoidal); a maximum function value; and a minimum function value. In this example, the function was applied over a bidirectional normalization of distance. However, this choice was arbitrary and other pertinent system metrics might be more appropriate.

For the string of individual values, two orderings appear to be natural for the problem: parameter ordering and compartment ordering. For parameter ordering, like-parameter types are ordered adjacently along the string. For example (using two channel conductance parameters):

$$Na_1 Na_2 ... Na_N ... K(DR)_1 K(DR)_2 ... K(DR)_N \quad (20)$$

where Na and K(DR) represent the sodium and potassium delayed rectifier conductance parameters, and the subscripts refer to compartment number (N total compartments). In compartment ordering, all parameters from the same compartment maintain adjacency along the string. For example:

$$Na_1 K(DR)_1 ... Na_2 K(DR)_2 ... Na_N K(DR)_N \quad (21)$$

Such an encoding would yield a string of length MN, where M is the number of parameter *types* and N is the number of compartments. Thus, this representation scales with the number of compartments. For the Traub model, we have a string length of 114. The De Schutter-Bower model would be searched rather inefficiently with this representation, as the string would contain a total of 19,200 parameters.

A scaled function embedding requires assumptions about the parameter distributions that we expect to see in nature. It is reasonable to assume that the actual distributions, with respect to the linear distance from the soma, may be described as noisy instances of a constant, linear, exponential, or sinusoidal function (Stuart and Sakmann 1994, Masukawa *et al* 1991). Additionally, structure-specific functions may be used to account for the observations that some channels may be present only at, for example, branch points (Catterall *et al* 1991). While this approach has the advantage

of incorporating domain-based observations, we are limiting the EA to exploring combinations of these plausible distributions. It would be somehow more satisfying to start from random conditions and have the EA converge on a set of parameters that could be fit by scaled functions.

In this embedding space (Figure 5), each parameter distribution is defined by three string values: a value encoding function type; a maximum function value; and a minimum function value. An additional term could be optionally added to include a degree of noise to the function distributions.

NaFunct NaMax NaMin [NaNoise] ... K(DR)Funct K(DR)Max K(DR)Min [K(DR) Noise] (22)

Such an encoding would yield a string of length 4M, where M is the number of parameter *types*. This representation is independent of any assumptions regarding the number of compartments appropriate for the model. The function would be scaled over the range determined by the maximum and minimum values. The parameter value for each compartment would be the evaluation of the function relative to the position of that compartment over the normalized length of the neuron.

The ordering of parameters along the string affects the defining length, e.g., the difference in the position along the string of those parameters that contribute most to achieving high fitness. Strings with longer defining lengths have a higher probability of being disrupted during crossover. Consequently, different orderings of the same problem will be differentially affected by the same crossover operator. On one hand, it is useful for crossover to be disruptive so that a thorough exploration of parameter combinations can take place. However, too much disruption may destroy any high order combinations of optimal parameters that the EA has discovered, rendering the overall search inefficient.

It is difficult to know *a priori* which of the three embedding descriptions would be most effective. If we chose a parameter ordering, the first parameter type and the last parameter type along the length of the string would be disrupted with a high probability (with respect to 1-pt crossover analysis). If we chose a compartment ordering, compartments numbered furthest apart would be disrupted most often. Which is more important to maintain - the longitudinal compartment distribution of parameters or the inner-compartment ratio of parameters types to one another? Without any empirical or theoretical basis for the decision, one approach is to develop more than one crossover operator so that both orderings will be operated upon. Estimating the disruptive effect of crossover on the scaled function embedding is not quite as intuitive because each *parameter value for the model* is embedded in three (four) dimensions instead of one in the *string parameter* space. Previous work suggests that EAs are most effective when each string parameter can be optimized independently (Davis 1991). Consequently, a function-based search may be difficult for the EA to optimize. Determining the pros and cons of each ordering will require empirical exploration.

Genetic algorithms have traditionally used binary representations, primarily because of the relative simplicity in analysis offered by low cardinality

alphabets. The parameters in neuron models are real-valued, however. Transformation of real values into a binary representation would have required a trade-off between the precision and efficiency of search. For example, if the resolution we desire for a given parameter is 0.1 arbitrary units, and if we assume a search range of 0.0-100.0, then each binary representation of a parameter would require 10 bits. The 114 parameters of the Traub model would be encoded in a string length of 1140. This is a relatively long string over which we would expect recombination to be inefficient. Increasing the number of compartments would exacerbate this problem. The EA would spend too much time searching the least significant digits of the string early in the search. Once the significant digits had converged, they would no longer require searching. However, genetic drift would have caused a certain degree of convergence on the least significant digits such that there would no longer be a random distribution in the population to sufficiently explore that area of parameter space. Real encodings offer many advantages over binary encodings, including the convenience and psychological comfort that comes with the direct correspondence between the mapping spaces. Technical benefits of real codes include avoidance of Hamming Cliffs, elimination of crossover disruption *within* a parameter, and a reduced susceptibility to deceptive path traversal (Goldberg 1991).

For reasons of convenience, we decided to implement a real-valued string. To enable a more direct comparison to Traub's parameters, we chose to represent the space as a string of 114 values. Since we lacked sufficient information to choose between parameter ordering or compartment ordering, we decided to develop crossover operators that transposed and exploited both orderings.

Recombination Operators

Historically, crossover has been associated with genetic algorithms and genetic programming, whereas mutation had been predominantly studied in evolutionary programming and evolution strategies. Exclusive use of either operator is becoming less prevalent, especially in applications. Theoretical and empirical studies have compared the conditions under which one operator is more successful than the other. The results do not provide us with much guidance, however. The main conclusion is that the success of a recombination strategy is specific to the problem being optimized and cannot be predicted *a priori*.

Commonly studied and applied forms of crossover include the 2-point and uniform operators. Early research suggested that the number of crossover points should be low to minimize disruption (Holland 1975, De Jong 1975). In contrast, many recent applications have demonstrated superior performance with more disruptive recombination operators (Syswerda 1989). It has been theoretically demonstrated that 2-pt crossover [7] is the least disruptive crossover operator. In contrast, uniform crossover [8] is the most disruptive but is without a positional bias (Spears and De Jong 1990). The question remains, which crossover operator should we apply to which ordering of our problem?

Mutation is both a mechanism for local search and a mechanism to prevent solution components lost during crossover from being eliminated for all successive generations. Mutation is applied in binary representations by flipping the value of the bit in the randomly selected position. In real-valued strings, more sophisticated scaling or replacement methods are required. Genetic algorithms typically apply mutation as a background operator at low rates, for example 0.1% (Goldberg 1989). In contrast, it has been shown that a high mutation rate can be beneficial in some applications (Bramlette 1991). In particular, a rate of 20% is recommended in some studies (Bagchi *et al* 1991).

Because we were left with questions about which rate and kind of real-valued mutation would be successful for our problem, we decided to let the EA tell us which strategy worked best. This was implemented as a self-adaptive strategy for choosing the rate of application of *all* the operators of potential interest to us. This approach has been used to select mutation rates in evolution strategies (Bäck *et al* 1991, Saravanan *et al* 1995) and crossover rates in genetic algorithms (Spears 1995). Further, Parsons *et al* (1995) report a synergistic effect gained by retaining many types of operator, suggesting additional benefits gained from this strategy.

We have custom-designed recombination operators [9] to explore the real-valued parameter space potentially suitable for the compartmental neuron problem. The set of recombination operators includes: *crossover 1 (2-point, compartment ordering), crossover 2 (2-point, parameter ordering), single parameter mutate, parameter-type scale, and scale all* (Figure 6). Two versions of the 2-point crossover operator were applied to exploit both *compartment ordering* and *parameter ordering*. This allowed us to reduce positional bias while maintaining low disruption. The *single parameter mutation* explored solutions adjacent in a single dimension to current solutions. We assumed that the ratios of the conductances of each of the channel types were important determinants of high fitness solutions and subsequently developed scaling operators, *scale all* and *parameter-type scale*, to explore this assumption.

All operators had the same initial probability of being chosen. Subsequent probabilities were based on the success of a given operator relative to that of the others. The minimum probability of operator selection was fixed at 1%. A success was awarded when a parameter set produced by a given operator scored a fitness value above the average fitness of the population.
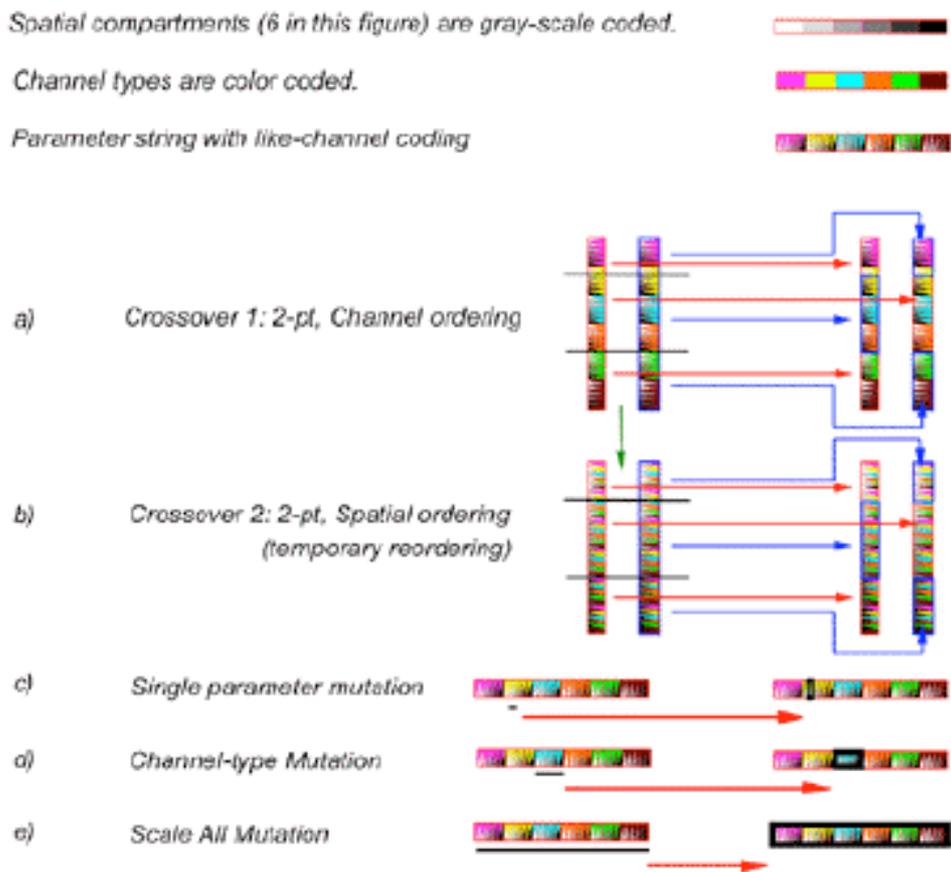
Figure 6

The crossover and mutation operators for the EA parameter search. Two parent parameter strings (represented with red and blue borders) were randomly selected from the mating pool for crossover operations. One parameter string was randomly selected from the mating pool for mutation operations. a, *Crossover 1 (2-pt, compartment ordering).* Two points along the string are randomly selected. The parent strings contribute alternate regions to the offspring string. b, *Crossover 2 (2-pt, parameter ordering).* The procedure is the same as in (a), except the parameters were first reordered such the same compartment parameters were adjacent. c, *Single Parameter Mutate.* Two random numbers were selected. The first number determined which of the parameters would be operated on. The second number was scaled to the range appropriate to that parameter and replaced the original value of the selected parameter. d, *Parameter-type Scale Mutation.* Two random numbers were selected. The first number determined which of the parameter types would be operated on. The second number was a scaling factor that ranged from 0.0 to 2.0. All conductances of the selected channel type were multiplied by the scaling factor. e, *Scale All Mutation.* Random number selection (a coin flip with probability 0.5) determined whether all values in a string were scaled by constant multiplicative factor 0.9 or 1.1.

Selection Mechanisms

Selection identifies and retains high fitness strings. Selection mechanisms are characterized based on strength. Strong selection concentrates quickly on exploiting the best individuals, favoring fast convergence at the expense of minimally sampling the search space. Such a strategy could potentially

lead to premature convergence on a less than optimal solution, but the answer is obtained quickly. Weak selection maintains a highly diverse population, but at the expense of increased search time. Very weak selection results in a random walk search. Since we want to evolve a *manifold* and not just a "best" single solution, we will avoid the stronger methods. This would potentially cost us an increased number of function evaluations, but we hoped that we would find better solutions.

We developed the following selection mechanism. An extinction threshold, a variable fitness criterion, was automatically adjusted to maintain a population size between 500 and 1000 population members. Population members from both current and previous generations were admitted to the mating pool if their individual fitness values exceeded the extinction threshold. This threshold-based "elite selection" mechanism allowed a highly fit individual to contribute to the mating pool for multiple generations if its fitness was sufficiently high, thus preventing the loss of good strings (Bäck and Hoffmeister 1991). Many EA implementations allow multiple copies of high fitness strings to contribute to the mating pool in proportion to the absolute or scaled fitness value of that string relative to the other strings (Forrest 1993, Goldberg 1989). Our implementation limited membership in the mating pool to a single copy of each suprathreshold string. This strategy reduced the possibility of premature convergence caused by the overexpression of proportionately higher fitness strings. The initial population size was 1000 parameter sets. Preliminary results suggested that this population size provided a sufficient compromise between diversity and efficiency [10]. The population size must be large enough to maintain a diverse gene pool. Otherwise, the EA must primarily rely on mutations to explore the space. In such a case, the rate of improvement is reduced to that of a random walk search. However, an extremely large population could increase the amount of computer time needed to solve the problem and might not improve the rate at which the problem is solved (Macready *et al* 1996).

Initialization

The initial parameters were randomly selected and scaled as follows: 1) 114 real-valued random numbers were drawn between 0 and 1; 2) a random number was selected between 0.5 and 2.0 to scale the *excitatory-to-inhibitory conductance ratio*; and 3) a random number between 0 and 2,500 mS/cm^2 was selected to scale the *total conductance*. Preliminary results (data not shown) suggested that parameter sets with values outside of these bounds did not yield fitnesses greater than 0.1. The recombination operators allowed subsequent generations to explore beyond these initial bounds.

Termination

Termination criteria in the EA literature are somewhat arbitrary because the statistical nature of the search prohibits knowing *when* the population will evolve to a given fitness value. Typically, EA searches are terminated after: 1) a predetermined number of generations or individual population members have been evaluated; 2) a goal fitness (such as 1.0) has been achieved; or, 3) the diversity of the string values in the population has been significantly

reduced. Our arbitrary termination criterion was satisfied after 200 CPU hours had elapsed on each of eight dedicated Silicon Graphics Power Challenge R8000 90 MHz processors (approximately 125 Tera floating point operations total). At most HPC research centers, researchers are allocated a certain number of CPU cycles on various machines during six month to one year grant periods. We find that a CPU-based termination criteria gives us a much more informative measure of what we can potentially solve with the CPU time we have available.

5. Designing a Fitness Measure

It was not clear how much information and what kind of information should be included in our fitness criteria. However, it would seem to be possible to manipulate the algorithm to produce whatever results we want. Therefore, any claims regarding the power of the EA parameter optimizing approach would be compromised if we *tweaked* the fitness measure until we obtained good performance. Therefore, we attempted to objectively design and adhere to a fitness measure philosophy before we began the EA search. While the specific quantitative aspects of our criteria were extracted from the biological literature, our approach to fitness measure design was intended to be a generally applicable to any time series.

We partitioned the information into three categories: 1) mathematical requirements; 2) etiological satisfaction; and 3) waveform decomposition. The decision to award 20% of the fitness points to the mathematical requirements and etiological satisfaction categories and 60% to the waveform decomposition was arbitrary. Given the variability in real physiological responses, we distributed the assignment of scores using a gradient rather than a step function. In other words, a *range* of values received the full score for satisfaction of each individual criterion and those values that were close to the optimal range received partial credit. By *coaxing* the evolution along through the assignment of partial credit, we provide the EA with more information to guide the search given a constant number of fitness criteria.

- *Mathematical requirements:* Time series exhibit damped, periodic, chaotic, or random characteristics. The stimulus we applied to the model should generate *approximately* periodic bursting. Therefore, the first goal was to partition the space of all solutions to eliminate parameter spaces that yield damped or steady state time series [11]. Credit was assigned for fulfilling the following successive criteria. Does the solution demonstrate spiking at all? Does it spike beyond the initial transient? Does it spike during the last 20% of the time series?

- *Etiological criteria:* Does the parameter set produce periodic solutions for the wrong reasons? The somatic time series should yield spiking in response to the depolarizing current injection. Dendritic compartments with sustained, elevated voltages could provide a source of depolarizing axial current that would inappropriately (in the absence of synaptic currents in this model) lead to somatic spiking. Thinking like an experimental biologist, we considered these solutions equivalent to neurons which had their dendritic membranes damaged during experimental preparation and rejected them.

- *Waveform decomposition:* The fitness filter is not a temporal point-by-point curve fitter. Instead, we decomposed the evaluation of each spatiotemporal dataset into a set of characteristics extended along multiple scales (figure 7). The largest scale feature was the somatic burst rate. We then *zoomed* into the characteristics of the individual bursts to evaluate such features as the peak-to-burst ratio, burst width, and the least squares fit to an experimental CA3 burst waveform. At the finest scale, we evaluated the postburst afterhyperpotential amplitude as well as quantitative aspects of individual spikes such as the width and height. Lastly, spatial behaviors such as the relative calcium influx and peak height were examined and scored.
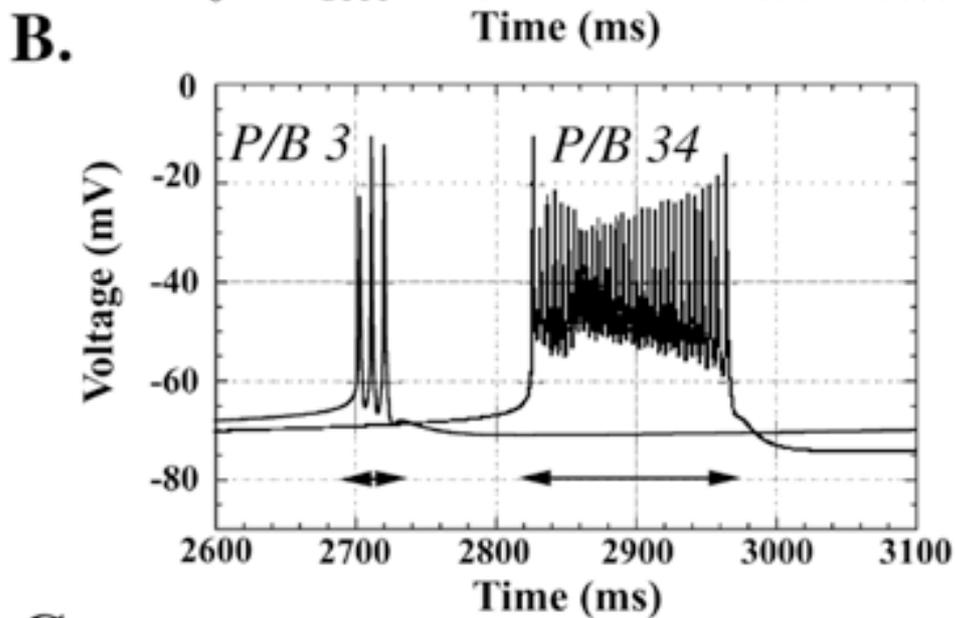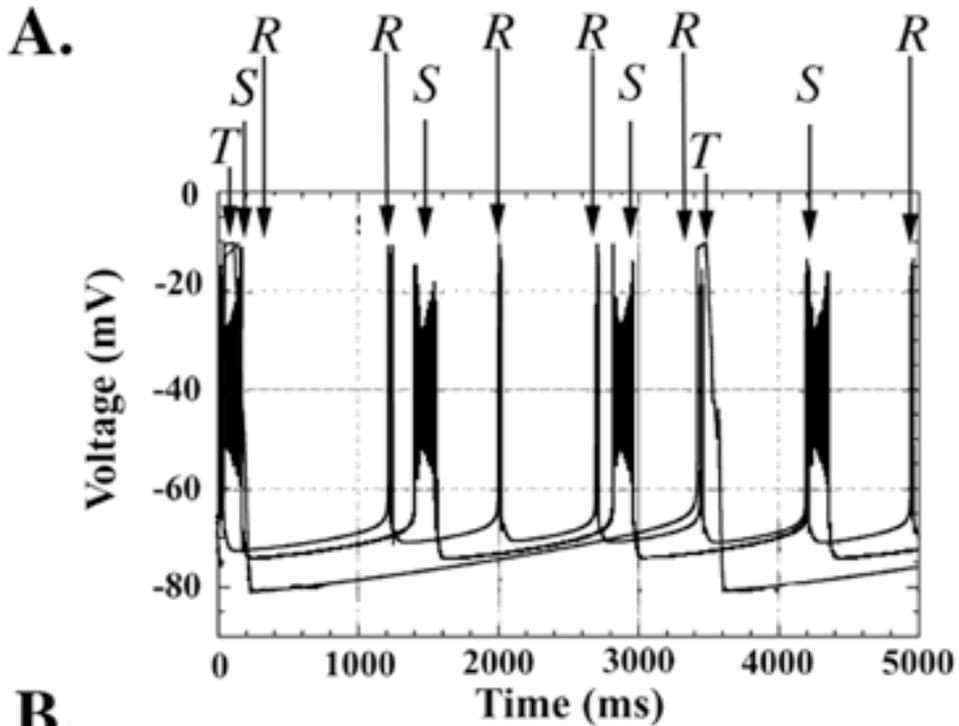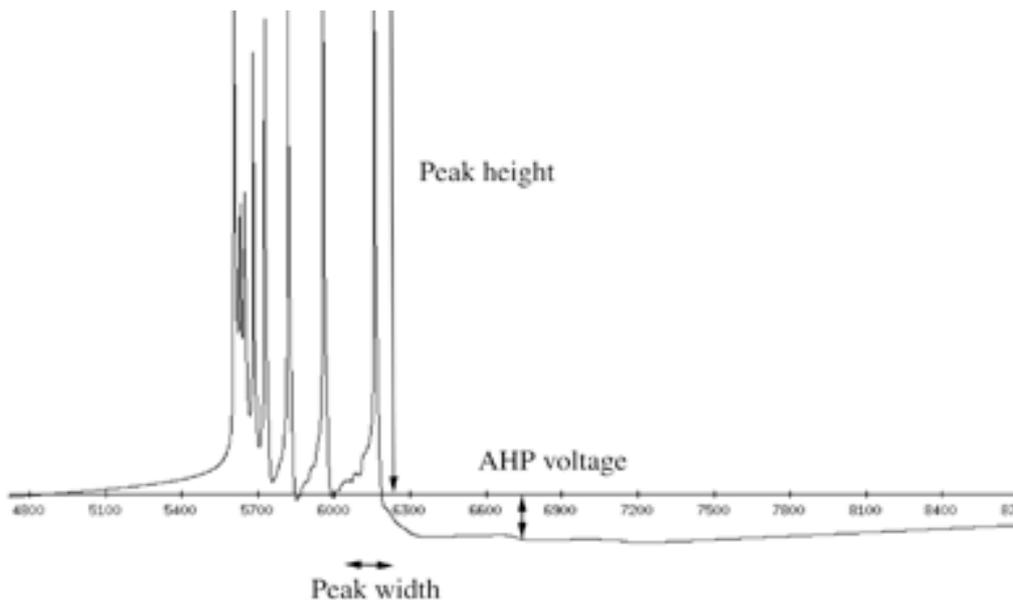
Figure 7

(cont): Waveform decomposition for the somatic time series. (**A**) Global behavior, such as the burst rate, was evaluated and scored. Time series R and S received full credit for satisfaction of this criteria because the burst rates, approximatley 0.8 and 1.2 Hz respectively, are within the experimentally observed range. (**B**) Qualities of the individual bursts, such as peak-to-burst ratio and burst width, were evaluated and scored. The left burst received maximal credit for a peak-to-burst ratio and burst width within the acceptable ranges. (**C**) The fine detail characteristics of the spikes within a burst, such as peak height and width and the afterhyperpotential (AHP) voltage, were the final behaviors to be evaluated.

Finally, we specifically included observations noted by Traub into our evaluation. It does not appear that Traub proposed a list of criteria and then optimized the parameters to meet these criteria. Rather, it is simply reported that the model is able to replicate experimental observations such as the following: 1) afterhyperpotentials fell 5 to 10 mV below resting potential; 2) somatic bursts produced at approximately 1 Hz in response to the application of 0.3 nA depolarizing current in the soma; 3) apical and basal dendrites yielded spikes; 4) maximum calcium influx observed in the midapical dendrites; and 5) spike width at half amplitude in soma should be approximately 0.8 ms.

6. HPC and Parallelism

The simulation executed for five seconds of neuron-time so that sufficient data were available for evaluation by the fitness filter. Each simulation required 49 seconds on a Silicon Graphics R8000 90 MHz processor. For a model such as the Purkinje cell (De Schutter and Bower 1994), each simulation could require hours. The evaluation of each parameter set required a significant amount of computer time. Because each parameter set can be evaluated independently, EA applications have a natural parallel decomposition. Our EA searches have been implemented in parallel in the following four ways: 1) fork/exec functions; 2) rcp/rsh scripts; 3) Loadleveler/

DQS application management; and 4) MPI/PVM libraries. We used combinations of these approaches to maximize our access to heterogeneous computing environments. This environment included a 9 processor Cray C916/10512, a 12 processor Silicon Graphics Power Challenge (R8000 75 MHz), a 8 processor Silicon Graphics Power Onyx (R8000 90 MHz), 8 IBM RS/6000 Model 590 workstations, and a 14 processor IBM SP Supercomputer.

In the simplest case, parallelism can be implemented at the process level using standard ANSI C system calls to *fork*/*exec* functions. No special software or libraries are required. Load balancing is handled by the operating system. However, the implementation is limited to shared memory architectures such as machines offered by Cray and Silicon Graphics. Our EA program executed each simulation independently as a child process. Each simulation was assigned an identification number using the *putenv* function. The simulator used the identification number to open the correct parameter file and to name the output file containing the fitness measure. The *wait* function monitored the completion status of the child processes. We found this method sufficient for most of our small to medium sized applications.

To implement parallelism on distributed memory systems and/or heterogeneous environments, we wrote shell scripts which used *rcp*/*rsh* to distribute the parameter files, execute the simulations on remote machines, and collect the results. This method required that an executable copy of the simulation program was installed on every machine. In the interest of security, many system administrators disable remote commands to prevent password-free access to their machines. While we were able to access many departmental workstations, we could not access the HPC machines in this fashion. To maximize efficiency, we wrote load balancing routines to determine the CPU load on each machine before submitting jobs. Unfortunatley, if other users submitted jobs after we started a simulation, we had to wait for simulations on the loaded machines to complete. If the wait became too long, we started another version of the delayed simulation on the fastest machine that was currently idle. We found this method of obtaining parallelism the least effective with respect to the machines to which we had access. However, this is a great way for researchers without access to machine time at a HPC center to steal extra CPU cycles from idle workstations overnight!

LoadLeveler is a sofware product developed by IBM for managing parallel applications. Versions of this product are available for IBM RS/6000 architectures, Sun, HP and Silicon Graphics workstations. This is reliable and easy to use software featuring machine specific configurations, load balancing, checkpointing, and a graphical interface for developing applications and monitoring the status of jobs currently executing. It schedules either serial or parallel (including MPI or PVM) jobs written as LoadLeveler or NQS scripts. The good news and bad news is that this is a commercial product; while it is technically well-supported by IBM, it is not free. As a low budg*et al*ternative, there are several shareware products

available via anonymous ftp that claim similar functionality and features. We did not evaluate the claims of the shareware products, but we found LoadLeveler to be a useful and stable tool.

MPI (Message Passing Interface) and PVM (Parallel Virtual Machine) are libraries of routines for passing messages between processors that potentially represent a range of architectures. These libraries have quickly become the standard for the development of parallel codes. The libraries are available via anonymous ftp, but must be installed on every machine participating in the virtual network. They offer a wide variety of communication classes including *gather*, *scatter*, and *all-to-all* using communication modes such as *standard*, *synchronous*, *ready*, or *buffered* in *blocking* and *non-blocking* forms (Snir *et al* 1995, Gest *et al* 1994). These libraries have limited error handling routines, do not support dynamic task spawning, and require the user to incorporate load balancing algorithms. However, the widespread interest in these libraries by developers and the adoption of these libraries by multiple vendors have resulted in a number of third party applications now available to manage CPU resources when the libraries themselves are insufficient.

For many applications, parallelism implemented with file transfers is inefficient compared with the message passing model. This is because the communication time required for the data transfer is large relative to the computation time performed by each distributed CPU (Kumar *et al* 1994). Most parallel architectures, (ex.. Thinking Machines Corporation's CM-5, Cray's T3E) have dedicated hardware and parallel versions of standard programming languages (ex. Fortran 90, C*) to reduce communication overhead, in addition to their support of MPI/PVM. Our programs yielded approximately linear speedup [12] with both the message passing and file transfer implementations because of the relatively large amount of CPU time required per simulation. However, it has been our observation from the EA literature that most fitness functions do not require as much CPU time as we needed. For maximum portability, longevity, and efficiency, we recommend parallelization using a message passing library such as MPI/PVM in conjunction with third party CPU management software. However, depending on the application and the available architectures, process/file-based parallelism is easy to implement and may be just as efficient as other methods.

7. A Success Story

We judged our application to be a success by two criteria. First, the average fitness of the population quickly exceeded the fitness score achieved by Traub's parameter set. Second, the high fitness parameter sets produced time series that resembled those observed experimentally. Similar results were obtained in three separate experiments, each using different random initial conditions. As a bonus, we identified manifolds for our low dimensional descriptors (the total conductance and the excitatory/inhibitory conductance ratio) of the evolved high dimensional parameter space which will allow us to reduce the parameter space in future refining searches. (See Eichler West 1996 for additional results.)

The EA evolved a population of high fitness parameter strings from random initial conditions. The initial generation yielded an average fitness of 0.04. This was consistent with our observations that parameter sets composed of random numbers showed a 96% probability of scoring a fitness less than 0.20 (n = 11,600). The EA improved the overall fitness of the population, rapidly at first but more slowly at higher fitness. The average fitness per generation exceeded the fitness of Traub's parameter set (0.85) by generation 62. The best fitness overall (0.92) was obtained in generation 101.
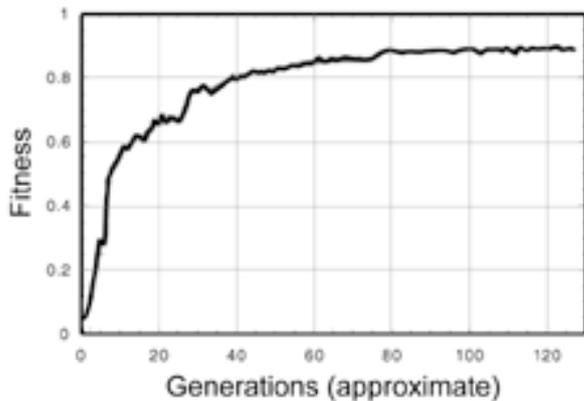

Figure 8

The average fitness per generation demonstrates rapid improvement. Because the population size is dynamic, the x-axis is only approximate.

The solution with the highest fitness demonstrates fast sodium-dependent bursting in the soma, low amplitude spiking in the basal dendrites, and slow calcium-dependent responses in the apical dendrites. Neither the EA-produced parameter sets nor Traub's parameter set satisfied all the fitness criteria, suggesting that additional channel kinetic equations may be necessary to achieve a higher correspondence to experimental results. A quantitative description is available elsewhere (Eichler West 1996).
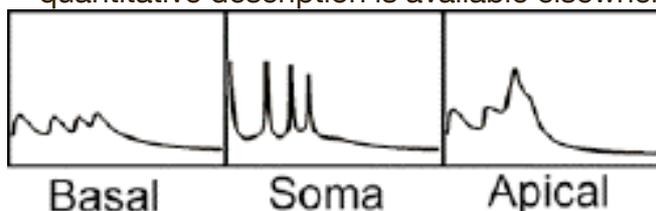

Basal   Soma   Apical

Figure 9

The solution with highest fitness yield spatiotemporal behaviors consistent with those observed experimentally. The y-axis range is -85 mV to +40 mV. The length of the x-axis is approximately 80 ms.

We examined the manifolds for the total conductance and the excitatory/inhibitory conductance ratio. The total conductances of high fitness solutions (> 0.9) lay between 442 and 652 mS/cm^2 ($\bar{x}$ = 513, s.d. = 34, n = 47,484), consistent with the total conductance of 527 mS/cm^2 proposed by Traub. The excitatory/inhibitory conductance ratio (EIR) for high fitness solutions (> 0.9) occurred in the range 0.78 to 1.31 ($\bar{x}$ = 0.96, s.d. = 0.12, n = 47,484), consistent with the EIR of 0.97 proposed by Traub. Many low

fitness solutions also fell within the ranges of these solutions. Thus, while selecting a parameter set within these ranges does not guarantee a high fitness solution, selection of any parameter set outside of this range yielded low fitness solutions.
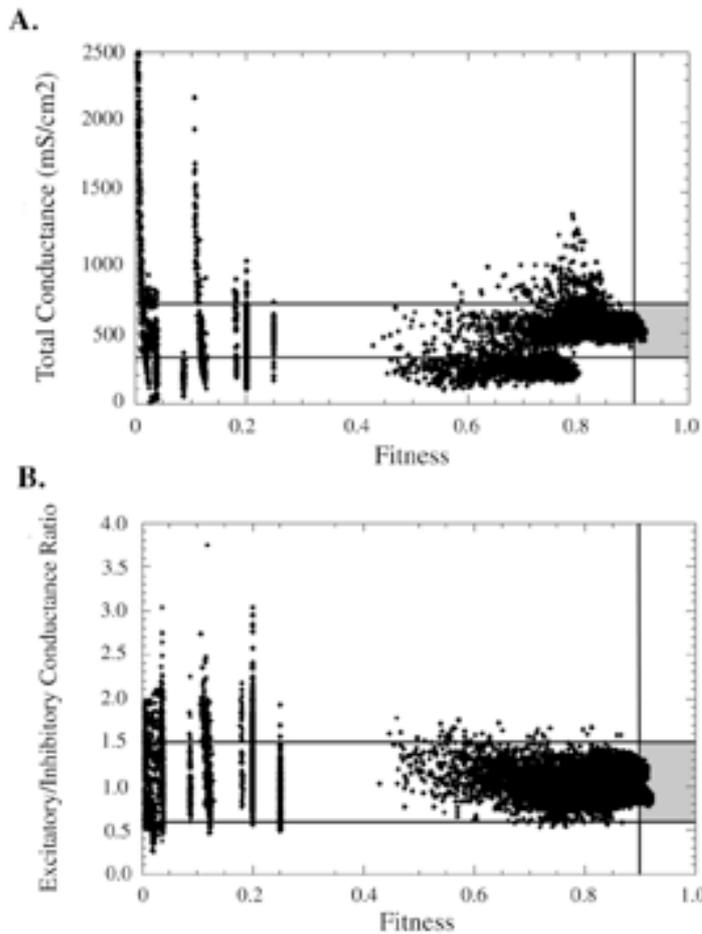

Figure 10

Solution manifolds for the low dimensional metrics as a function of fitness. a, Total conductance manifold as a function of fitness score for the search space explored by the EA. b, The excitatory/inhibitory conductance ratio manifold as a function of fitness score. Note that the range with respect to the y-axis decreases with increasing fitness.

8. Conclusions

This study has applied a new method for parameter optimization that promises to dramatically improve the robustness of neuronal simulations. Further, the automation aspects of the search process promises to improve the efficiency of model development. By robustness, we mean that the manifold of high fitness solutions produced by the EA application demonstrates that the model produces appropriate behaviors over a range of parameter values, reflecting a variability analogous to that observed in nature. The results of our proof-of-concept experiment yielded improved correspondence between simulated and experimental behaviors. The method should find general applicability in a broad range of HPC simulation endeavors outside of neurophysiological modeling.

The manifold approach suggests methods of analysis that would not have

been possible with previous approaches to parameter fitting.

- 1. *The EA-generated manifold can be used to determine an area of parameter space to examine in future refining searches*. While we carefully designed our recombination operators to explore outside of the boundaries of our initial conditions, high fitness solutions confined themselves to a more restricted area of parameter space.

- 2. The interpretation and significance of the individual channel parameter manifolds require a much greater description of the biological system and are thus not within the scope of this chapter [13]. However, *all the channel distribution manifolds resembled noisy instances of simple functions*: linear (potassium A, calcium-dependent potassium); exponential (sodium, potassium delayed rectifier); or, sinusoidal (calcium, potassium AHP). This encouraged us to re-evaluate the potential of the function representation described in section 4.

- 3. *The manifolds were subsequently analyzed to reveal covariance relationships between parameters, thereby providing an indication of the sensitivity of parameters with respect to modulations in the other parameters*. The manifolds identified varying degrees of parameter sensitivity for the potassium A and the calcium-dependent potassium channels and a strong positive covariance between the sodium and potassium delayed rectifier channel distributions. The complementary colocalization of the calcium and potassium AHP distributions is suggestive of an experimentally testable role for calcium-dependent conductances in the control of spatiotemporal plasticity (Eichler West 1996).

One future goal made evident by these preliminary studies is the need to reduce the number of function evaluations required to sufficiently search parameter space by applying better EA selection strategies. The results in figure 8 suggest that a reexamination of our termination criteria might be the first place to start. The search strategy we used required access to significant amounts of CPU time on supercomputer-class machines. We believe that our EA-based approach to neuronal modeling offers a powerful new set of methods and interpretive paradigms, but it will not be generally useful to all neural modelers until it is implementable by those who do not have access to high performance computers.

How do we know when we are including too much or too little information to the fitness measure? The concern is that additional criteria will increase the computational time required to evaluate the response of each simulation with little or no gain for parameter manifold refinement. One approach may be to divide criteria into "training and testing" sets. The set of high fitness solutions created from the "training" criteria can be evaluated for criteria for which there is no explicit selection (the "testing set"). Those testing criteria which fail must necessarily become members of the new training set.

We are first and foremost neuroscientists with collective intuition into our system gained by reviewing the experimental literature, performing experiments in the laboratory, and simulating models of neurons. In this first attempt at applying an EA, we tried to incorporate much domain-based intuition into the algorithm design. The application was quite successful.

However, with respect to the NFL theorem, we are still left with two nagging questions. Would another method perform better, given a similar attentiveness to the incorporation of domain-based information? Did we get lucky? Answers to these questions will require significantly more empirical research.

References

Abbott, L. F., Rolls, E. T., and Tovee, M. J. Representational capacity of face coding in monkeys. *Cerebral Cortex* **6**: 498-505, 1996.

Back, T., and Hoffmeister, F. "Extended selection mechanisms in genetic algorithms." In Fourth International Conference on Genetic Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan Kaufmann, 92-99, 1991.

Back, T., Hoffmeister, F., and Schwefel, H.-P. "A survey of evolution strategies." In Fourth International Conference on Genetic Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan Kaufmann, 2-9, 1991.

Bagchi, S.*et al*. "Exploring problem-specific recombination operators for job shop scheduling." In Fourth International Conference on Genetic Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan Kaufmann, 10-17, 1991.

Bhalla, U. S., and Bower, J. M. Exploring parameter space in detailed single neuron models: simulations of the mitral and granule cells of the olfactory bulb. *Journal of Neurophysiology* **69**: 1948-1965, 1993.

Bramlette, M. F. "Initialization, mutation and selection methods in genetic algorithms." In Fourth International Conference on Genetic Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan Kauffman, 100-107, 1991.

Chetkovich, D. M.*et al*. N-Methyl-D-Aspartate receptor activation increases cAMP levels and voltage-gated Ca2+ channel activity in area CA1 of hippocampus. *Proceedings of the National Academy of Sciences of the USA* **88**: 6467-6471, 1991.

Culberson, J. On the futility of blind search. *University of Alberta Technical Report TR96-18* 1996.

Davis, L. "Bit-climbing, representational bias, and test suite design." In Fourth International Conference on Genetic Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan

Kaufmann, 18-23, 1991.

Davis, L.*et al*. "A genetic algorithm for survivable network design." In Fifth International Conference on Genetic Algorithms in University of Illinois at Urbana-Champaign, edited by Forrest, S., Morgan Kaufmann, 408-415,1993.

De Jong, K. A. "An analysis of the behavior of a class of genetic adaptive algorithms." Doctoral Thesis, University of Michigan, Ann Arbor, 1975.

De Schutter, E. A consumer guide to neuronal modeling software. *Trends in Neurosciences* 15: 462-464, 1992.

De Schutter, E., and Bower, J. M. An active membrane model of the cerebellar purkinje cell. I. Simulation of current clamps in slice. *Journal of Neurophysiology* **71**: 375-400, 1994.

Denk, W., Strickler, J. H., and Webb, W. W. Photon laser scanning fluorescence microscopy. *Science* **248**: 73-76, 1990.

Ebner, T. J., and Chen, G. Use of voltage-sensitive dyes and optical recordings in the central-nervous-system. *Progress in Neurobiology* **46**: 463-506, 1995.

Edmonds, B.*et al*. Contributions of two types of calcium channels to synaptic transmission and plasticity. *Science* **250**: 1142-1146, 1990.

Eichler West, R. M. "On the development and interpretation of parameter manifolds for biophysically robust compartmental models of CA3 hippocampal neurons." Doctoral Thesis, University of Minnesota, 1996.

Eichler West, R. M., and Wilcox, G. L. A renumbering method to decrease matrix banding in equations describing branched neuron-like structures. *Journal of Neuroscience Methods* **68**: 15-19, 1996.

Ferster, D., and Spruston, N. Cracking the neural code. *Science* **270**: 756-757, 1995.

Fitzgerald, K.*et al*. Multiple forms of non-associative plasticity in Aplysia: a behavioral, cellular, and pharmacological analysis. *Philos Trans R Soc Lond* **329**: 171-178, 1990.

Forrest, S. Genetic algorithms: principles of natural selection applied to computation. *Science* **261**: 872-878, 1993.

Foster, W. R., Ungar, L. H., and Schwaber, J. S. Significance of conductances in Hodgkin-Huxley models. *Journal of Neurophysiology* **70**: 2502-2518, 1993.

Geist, A.*et al*. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing*. Cambridge, MA: MIT Press, 1994.

Georgopoulos, A. P., Taira, M., and Lukashin, A. Cognitive neurophysiology of the motor cortex. *Science* **260**: 47-52, 1993.

Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley, 1989.

Goldberg, D. E. Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems* **5**: 139-168, 1991.

Goldman, D. E. Potential, impedance, and rectification in membranes. *The Journal of General Physiology* **27**: 37-60, 1943.

Hart, W. E., and Belew, R. K. "Optimizing an arbitrary function is hard for Genetic Algorithms." In Fourth International Conference on Genetic

Algorithms in University of California, San Diego, edited by Belew, R. K., and Booker, L. B., Morgan Kaufmann, 190-195, 1991.

Hille, B. *Ionic Channels of Excitable Membranes*. second ed., Sunderland, MA: Sinauer Associates Inc., 1992.

Hines, M. Efficient computation of branched nerve equations. *International Journal of Biomedical Computing* **15**: 69-76, 1984.

Hodgkin, A. L., and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology* **117**: 500-544, 1952.

Hodgkin, A. L., and Rushton, W. A. H. The electrical constants of a crustacean nerve fibre. *Proceedings of the Royal Society of London Series B* **133**: 444-479, 1946.

Holland, J. H. *Adaptation in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press, 1975.

Holmes, R. W., and Rall, W. Estimating the electrotonic structure of neurons with compartmental models. *Journal of Neurophysiology* **68**: 1438-1452, 1992.

Jantsch, E. *The self-organizing universe: scientific and human implications of the emerging paradigm of evolution*. Elmsford, New York: Pergamon, 1980.

Johnston, D.*et al*. Active properties of neuronal dendrites. *Annual Review of Neuroscience* **19**: 165-186, 1996.

Kallen, R. G., Cohen, S. A., and Barchi, R. L. Structure, function, and expression of voltage-dependent sodium channels. *Molecular Neurobiology* **7**: 383-428, 1993.

Keynes, R. D. A new look at the mechanism of activation and inactivation of voltage-gated ion channels. *Proceedings of the Royal Society of London Series B* **249**: 107-112, 1992.

Kido, M.*et al*. Mantle viscosity derived by genetic algorithm using oceanic geoid and tomography for whole-mantle versus blocked-flow situations. *Phys. Earth Planet Int.* 1997.

Klein, M., and Kandel, E. R. Presynaptic modulation of voltage-dependent Ca2+ current: mechanism for behavioral sensitization in Aplysia californica. *Proceedings of the National Academy of Sciences of the USA* **75**: 3512-3516, 1978.

Kuhar, M. J., and Unnerstall, J. R. Quantitative receptor mapping by autoradiography: some current technical problems. *Trends in Neurosciences* 49-53, 1985.

Kumar, V.*et al*. *Introduction to Parallel Computing: Design and Analysis of Algorithms*. Benjamin-Cummings Addison-Wesley Publishing Company, 1994.

Laurent, G. Dynamical representation of odors by oscillating and evolving neural assemblies. *Trends in Neurosciences* **19**: 489-496, 1996.

Li, M.*et al*. Convergent regulation of sodium channels by protein kinase C and cAMP-dependent protein kinase. *Science* **261**: 1439-1442, 1993.

Macready, W. G., Siapas, A. G., and Kauffman, S. A. Criticality and parallelism in combinatorial optimization. *Science* **261**: 56-58, 1996.

Maletic-Savatic, M., Lenn, N. J., and Trimmer, J. S. Differential spatiotemporal expression of K+ channel polypeptides in rat hippocampal neurons developing in situ and in vitro. *Journal of Neuroscience* **15**: 3840-3851, 1995.

Mascagni, M. V.: Numerical methods for neuronal modeling. In *Methods in Neuronal Modeling*, edited by Koch, C., and Segev, I., Cambridge, Ma: MIT Press, 1989, p. 439-483.

Masukawa, L. M., Hansen, A. J., and Shepherd, G. Distribution of single-channel conductances in cultured rat hippocampal neurons. *Cellular and Molecular Neurobiology* **11**: 231-243, 1991.

Monster, A. W., and Chan, H. Isometric force production by motor units of extensor digitorum communis in man. *Journal of Neurophysiology* **40**: 1432-1443, 1977.

Numann, R., Caterall, W. A., and Scheuer, T. Functional modification of brain sodium channels by protein kinase C phosphorylation. *Science* **254**: 115-118, 1991.

Parsons, R. J., Forrest, S., and Burks, C. Genetic algorithms, operators, and DNA fragment assembly. *Machine Learning* **21**: 11-33, 1995.

Perezreyes, E., and Schneider, T. Calcium channels - structure, function, and classification. *Drug Development Research* **33**: 295-318, 1994.

Press, W. H.*et al*. *Numerical recipes. The art of scientific computing*. second ed., Cambridge: Cambridge University Press, 1992.

Rall, W. Theory of physiological properties of dendrites. *Annals New York Academy of Science* **96**: 1071-1092, 1962.

Rall, W.: Cable theory for dendritic neurons. In *Methods in Neuronal Modeling*, edited by Koch, C., and Segev, I., Cambridge, Mass: MIT Press, 1989, p. 9-62.

Rhodes, K. J.*et al*. Voltage-gated K+ channel beta-subunits - expression and distribution of KV-Beta-1 and KV-Beta-2 in adult rat brain. *Journal of Neuroscience* **16**: 4846-4860, 1996.

Sakmann, B., and Neher, E. *Single Channel Recording*. New York: Plenum, 1983.

Saravanan, N., Fogel, D. B., and Nelson, K. M. A Comparison of Methods for Self-Adaptation in Evolutionary Algorithms. *BioSystems* **36**: 157-166, 1995.

Segev, I., Fleshman, J. W., and Burke, R. E.: Compartmental models of complex neurons. In *Methods in Neuronal Modeling*, edited by Koch, C., and Segev, I., Cambridge, Mass: MIT Press, 1989, p. 63-96.

Snir, M.*et al*. *MPI: The Complete Reference*. Cambridge, MA: MIT Press, 1995.

Spears, W. M. "Adapting Crossover in Evolutionary Algorithms." In Proceedings of the Fourth Annual Conference on Evolutionary Programming in San Diego, CA, 1991.

Spears, W. M., and De Jong, K. A. "An analysis of multi-point crossover." In Proceedings of the Foundations of Genetic Algorithms Workshop in Bloomington, IN, 1990.

Spruston, N., Jaffe, D. B., and Johnston, D. Dendritic attenuation of synaptic potentials and currents - the role of passive membrane properties. *Trends in*

Neurosciences **17**: 161-166, 1994.

Spruston, N., and Johnston, D. Perforated patch-clamp analysis of the passive membrane properties of three classes of hippocampal neurons. *Journal of Neurophysiology* **67**: 508-529, 1992.

Stuart, G. J., and Sakmann, B. Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature* **367**: 69-72, 1994.

Syswerda, G. "Uniform crossover in genetic algorithms." In Third International Conference on Genetic Algorithms in edited by Shaffer, J. D., Morgan Kaufmann, 1989.

Theunissen, F. E.*et al*. Information theoretic analysis of dynamical encoding by four identified primary sensory interneurons in the cricket cercal system. *Journal of Neurophysiology* **75**: 1345-1364, 1996.

Toro, L., and Stefani, E. Calcium-activated K+ channels - metabolic regulation. *Journal of Bioengineering-B* **23**: 561-576, 1991.

Traub, R. D.*et al*. Analysis of gamma rhythms in the rat hippocampus in vitro and in vivo. *Journal of Physiology* **493**: 471-484, 1996.

Traub, R. D.*et al*. A model of a CA3 hippocampal pyramidal neuron incorporating voltage-clamp data on intrinsic conductances. *Journal of Neurophysiology* **66**: 635-650, 1991.

Turner, D. A. Segmental cable evaluation of somatic transients in hippocampal neurons (CA1, CA3, and Dentate). *Biophysical Journal* **46**: 73-84, 1984.

Turner, D. A., and Schwartzkroin, P. A. Steady-state electrotonic analysis of intracellularly stained hippocampal neurons. *Journal of Neurophysiology* **44**: 184-199, 1980.

Westenbroek, R. E., Ahlijanian, M. K., and Catterall, W. A. Clutsering of L-type calcium channels at the base of major dendrites in the hippocampal pyramidal neurons. *Nature* **347**: 281-284, 1990.

Whitley, D.*et al*. "Comparing Heuristic, Evolutionary and Local Search Approaches to Scheduling." In Third Artificial Intelligence Planning Systems Conference.1995.

Wolpert, D. H., and Macready, W. G. No free lunch theorems for search. Santa Fe Institute, 1995. 95-02-010.

Wonderlin, W. F., French, R. J., and Arispe, N. J.: Recording and analysis of currents from single ion channels. In *Neurophysiological Methods*, edited by Vanderwolf, C. H., Clifton, NJ: Humana Press, 1990, p. 35-142.