Evolutionary computing in neuronal modeling

Anil K Bheemaiah
Neuroscience Training Program
1300 University Avenue
Madison WI 53705

Abstract

The efficacy of genetic algorithms in the design of models that model specific and experimental aspects of action potentials in a wide variety of organisms is proven. A specific example of a plant action potential is used to illustrate the use of genetic algorithms in the search for parameters of models. The efficiency of the genetic algorithms as a search method is in the short generation span of the convergence of the algorithm.

Introduction

Neuronal modeling is the mathematical and computational modeling of single neurons and network of neurons in a phenomenological way. Experimental work has shed much light on the topic of ion channels in excitable membranes, but often the parameters of model construction are not derivable from simple experiments, this reduces the modeling to much an art of educated guess work. Specific instances include the kinetics of several of the processes and channels and the spatial density of channels. Much of such data is also very difficult to gather experimentally. To correctly model a neural system by this handwork is a laborious and time consuming process. Often on it is sure luck that models replicate the neurobiological data. Considering that the parameters interact with each other in highly nonlinear ways making iterative hand tuning of the parameters difficult. This is where parameter search algorithms fit in.

Further a similarity between the model and experimental data is the sole criteria for hand tweaked models, parameter search methods investigate the whole behavior of the parameter space.

They reduce the effort in building compartmental models. By automatically searching the search space for parameters which would lead to good models.

Thus several parallel efforts have been made by many groups(Baldi et al 1998, Balla and Bower 1993, Varnier and Bower 1996, Varnier and Bower 1999) in using automated search methods.

In the present compilation we use genetic algorithms, the most popular of evolutionary computing, to search the parameter space of a single compartmental model of an algae or a higher plant.

Since plant cells do not have a dendritic structure, a single compartment suffices to study plant action potentials.

Methods:

We use the genesis simulator to simulate a single compartmental model of a plant cell. The plant cell has a delayed rectification K channel, a Ca channel and a Ca dependent Cl channel. We have omitted the H⁺ ATPase pump from the model. The depolarization is by the release of Ca from stores and due to a Ca channel and due to the chloride channel that depends on Calcium. There is an IP3 mediated calcium release mechanism which we have simplified in a model of calcium concentration that decays exponentially with time. The genesis package comes inbuilt with search algorithms that includes the genetic algorithm, we use this to run a simulation to search for many parameters including the

gbar for the chloride and calcium channels and the time constant and middle point of the potassium channel parameters, N_{inf} and τ_N . The simulations have a time step of 20 micro seconds and are responses of the single compartmental model to current injections of .1 nanoampere to 1.0 nano ampere.

Genetic algorithms

Genetic algorithms is a biomimetic science, its origin comes from a computational study of natural evolution. Much of the terminology is borrowed from the field of genetics. We first introduce some of the terminology. All biological organisms consist of cells, these cells contain DNA or genetic information in one or more chromosomes. These chromosomes consist of genes. Each gene encodes a trait. The different settings for a trait are called 'allels'. Each genome is located at a particular locus or position in the gene. The complete collection of all the genes, that is the chromosomes of an organism is called the genome.

The term genotype refers to a set of genes in the genome. Two organisms with the same genes are said to have the same genotype. The genotype gives rise to the phenotype on development. Organisms whose chromosomes are in pairs are called diploid, a single set of chromosomes is haploid. During sexual reproduction, recombination or crossover of genes occurs to create the gamete. The gametes from the two parents combine to form a diploid offspring chromosome.

In haploid organisms, genes are exchanged from single strands of parent chromosome. The offspring genetic material is subject to random mutations owing to copying errors and other factors. The fitness of an individual organism is the probability the organism will live to reproduce (viability) or as a function of the number of offspring that organism has. (fertility)

In genetic algorithms, a chromosome is usually a bit string and denotes a candidate solution to a problem. An allele is usually a bit one or zero at each locus and in more complicated encodings, it can be more complicated. Cross over consists of exchange of genetic material between two haploid parents and mutation consists of inverting a single bit at some locus. Often there is no phenotype but problems like neural networks have both a phenotype and genotype.

A simple genetic algorithm

A genetic algorithm works as follows, given a clearly defined problem and a bit string representation for a candidate solution.

1. start with a randomly generated population of n l-bit chromosomes.
2. Calculate the fitness $f(x)$ of each chromosome x in the population.
3. Repeat the following steps until n offspring have been created.
 - a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of the fitness. Selection is done with "replacement" meaning that the same chromosome can be selected more than once to be a parent.
 - b. With crossover probability p_{cn} cross over the pair at a randomly chosen cross over point to form two offspring. If no cross over takes place form two offsprings that are exact replicas of the parent. There are also "multi-point crossover" versions of the genetic algorithms in which the cross over rate is the number of points the cross over takes place.
 - c. Mutate the two offsprings at each randomly chosen locus with probability p_m and place

the resulting
chromosomes in the new population.
4. replace the current population with the new population.
5. Goto step 2

A run is the entire set of generations, a generation being an iteration of the process.
A genetic algorithm(G.A)is typically iterated from 50 to 500 generations. At the end of the run there are several highly fit chromosomes.
G.A's tend to be more complicated than this scheme with different types of crossovers and mutations and also more complicated representation of the chromosomes.
There are however more details to fill like the size of the population and the probabilities of crossover and mutation. The success of the algorithm greatly depends on these parameters. Researchers often document the best fitness and the generation of that solution.

The simple operator definitions.

Selection: This operator selects chromosomes in the population for reproduction. The fitter the chromosome the more times it is likely to be selected to reproduce. A simple method of implementing fitness proportionate selection is the "roulette-wheel sampling" which is conceptually equivalent to giving each individual solution a slice of the circular roulette wheel equal in area to that individual's fitness.

The roulette wheel is spun, The ball comes to rest on one wedge shaped slice, the corresponding individual is selected.

Crossover: This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offsprings. For example the strings 10000100 and 11111111 could be crossed over after the third locus to give the offsprings 10011111 and 11100100.

Mutation: This operator randomly flips some of its bits in a chromosome. For example 00000100 can be mutated at the second position to give 01000100.

How do genetic algorithms work?

Although genetic algorithms are easy to code and perform very well, they are complicated. Many open questions exist about how they work and for what problems they are best suited. Much of the work done theoretically (Holland 1975) is on bit string representations of the chromosomes

The traditional theory assumes that GAs work by discovering, emphasizing and recombining good building blocks of solutions in a massively parallel way.

Holland introduced the notion of schemas or schemata to formalize the building blocks. A schema is a bit string of ones, zeros and asterisks, the asterisks in the schemas denoting don't care conditions or wild cards. For example the schema $H=1****1$ denotes all bit strings of $l=6$ That start and end with a one. The strings that fit this template are called instances, for example 100111 and 110011 are instances of H. The schema H has two defined bits and is called an order 2 schema.

Its defining length in a hyper plane of dimension l is 5.

Note that not every possible subset of the set of length l -bit strings can be defined as a

schema. There are 2^l possible bit strings and thus $2^{(2^l)}$ possible subsets of strings, but there are only 3^l schema. Any given bit string of length l is an instance of 2^l different schemas. Thus any given population of n strings contains instances of between 2^l and $n \cdot 2^l$ schemas.

This means that, at a given generation, while the GA is explicitly evaluating the fitness of the n strings of the population, it is actually implicitly estimating the average fitness of a larger population of schemas

Just as schemas are not explicitly represented or evaluated by the GA the estimates of the average fitness of the schema average fitness are not evaluated by the GA. However the GA's behavior in terms of increase or decrease of numbers of instances of given schemas in the population can be described as though it actually were actually computing and storing these averages.

Let H be a schema with at least one instance present in the population at time t , Let $m(H,t)$ be the number of instances of H at time t , and let $u(H,t)$ be the observed average fitness of H at time t .

We want to compute $E(m(H,t+1))$, the expected number of instances of H at time $t+1$, The expected number of offspring of a string x is $f(x)/f(t)$ where $f(x)$ is the fitness of x and $f(t)$ the average fitness of the population at time t . Then assuming that x is in the population at time t and x belongs to H , ignoring the effects of crossover and mutation we have.

$$E(m(H,t+1)) = \sum_{x \text{ belongs to } H} (f(x)/f(t)) = (u(H,t)/f(t)) m(H,t) \quad (1.1)$$

since $u(H,t) = \sum_{x \text{ belongs to } H} f(x) / m(H,t)$.

Cross over and mutation can increase or decrease the instances of H . To calculate a lower bound on $E(m(H,t+1))$ we consider only the destructive effects of mutation and crossover.

Let p_c be the probability of crossover of a single point. We can give a lower bound on the probability $S_c(H)$

that H will survive single point crossover.

$$S_c(H) \geq 1 - p_c(d(H)/(l-1))$$

where $d(H)$ is the defining length of H and l is the length of bit strings in the search space.

The disruptive effect of mutation can be quantified, let p_m be the probability of any bit being mutated. Then $S_m(H)$ is equal to $(1-p_m)^{o(H)}$ where $o(H)$ is the order of H , the number of defined bits in H . Thus the probability of surviving mutation is higher for lower order schemas. Amending this to equation 1.1 gives:

$$E(m(H,t+1)) \geq (u(H,t)/f(t)) m(H,t) \left(1 - p_c(d(H)/(l-1))\right) \left[(1 - p_m)^{o(H)}\right] \quad (1.2)$$

This is known as the Schema theorem. It describes the growth of a schema from one generation to the next. The schema theorem is often interpreted as implying that short low order schemas whose fitness remains above the mean will receive exponentially increasing number of samples over time.

The strength of a GA lies in the building block hypothesis that crossovers are the core of a GA's power. The lower bound evaluated in 2.2 is by no means a complete statement of

a G.A's power, for a constructive crossover definition see Holland 1975, Thierens and Golberg 1993 and Spears 1993.

Limitations of "static" schema analysis

Recent papers in genetic algorithms have criticized the schema formulation as a static building block hypothesis (SBBH)(Grefenstette 1993 and others). I will concentrate on Grefenstette's critique.

Grefenstette gives two possible reasons why the SBBH can fail

1. Collateral convergence: Once the population begins to converge at some loci, the samples of some schemes are no longer uniform.

For example suppose instances of $111*...*$ are highly fit and the population has more or less converged on this schema as a solution, then almost all samples of $***000*...*$ will actually be samples of $111000*...*$. This prevents the accurate estimate of $u(***000*...*)$

2. High fitness variance: If a schema's static average fitness has high variance, the GA may not be able to make an accurate estimate of this static average fitness.

for example the variance of $1*...*$ is high so the GA converges on the high fitness regions of this schema

This biases all subsequent samples of this scheme, preventing an accurate estimate of its static fitness.

There is nothing to indicate that the features listed above harm the search performance of GAs, they only demonstrate the danger of drawing conclusions about the expected behavior of GAs from the static average fitness of schemas. Instead a more dynamic approach is needed that takes into account the biases introduced by selection at each generation.

One of such a scheme that pertains to a dynamical analysis is the royal road function. one can refer to Mitchell 1996 for a detailed analysis.

Iterated hill climbing techniques

We now branch out to some other search techniques

a. Steepest ascent hill climbing(SAHC)

1. Choose a string at random, call this string current-hilltop

2. Going from left to right systematically flip each bit in the string, one at a time, recording the fitness of the resulting one bit mutants.

3. If any of the resultant one bit mutants has a higher fitness set current-hill top to that one bit mutant giving the highest rise to the fitness, Ties are chosen at random.

If there is no fitness increase then save current hill top and go to step one, otherwise go to step 2

5. When a set number of function evaluations has been performed, return the highest hilltop that was found.

b. Next-ascent hill climbing(NAHC)

1. Choose a string at random. Call this string current-hilltop.

2. For i from 1 to l , where l is the length of the string, flip bit i , if this results in an increase in fitness keep the new string otherwise flip bit i back.

As soon as an increase in fitness is found set current hill top to that string without any more bit flips of that string. Goto step 2 with the new current hilltop but continue mutating the new string starting immediately after the bit position at which the previous

fitness increase was found.

3. If no increase in fitness were found save current hill top and goto step 1
4. When a set number of function evaluations has been performed return the highest hill top found.

c. Random-mutation hill climbing(RMHC)

1. Choose a string at random, call this best-evaluated.
2. Choose a locus at random to flip, if the flip leads to equal or higher fitness, then set best-evaluated to the resulting string.
3. goto step 2 until an optimum string has been found or until a maximum number of evaluations have been performed.
4. Return the current value of best-evaluated.

For a detailed theoretical analysis of the above methods and the GA on simple problems, refer to Mitchell 1996.

Genetic algorithms: Implementation issues.

There are many a successful genetic algorithm application, however there are many problems where GAs do not fare well. Given a particular problem how does one know whether a GA is a good method to use?

There is no rigorous answer, but if a problem is complex enough, the search space is large, and if the search space is not perfectly smooth and unimodal. If the fitness function is noisy or if any solution though not the globally optimized solution is needed.

Encoding the problem for a genetic algorithm.

The nature of encoding the problem along with the genetic algorithm parameters determines the success of a GA. In this section we explore the nature of encodings either as simple bit strings or more complicated encoding including real numbers.

Binary encodings

This is the most common way of encoding the problems solution. Much of the existing GA theory is based on fixed length binary encodings. Heuristics for parameters such as the population size and the crossover and mutation probabilities have been developed for bit string encoding.

Binary encodings are unnatural and unwieldy for a number of problems like the weights of a neural network. Many character and real-valued encodings.

For many applications it is most natural to use an alphabet of many characters or real valued representations. Examples include Montana and Davis's real valued representation for neural network weights, Schultz-Kremer's real valued representations for torsion angles in proteins. Holland's schema counting argument seems to imply that GAs should exhibit worse performance on multiple character encodings, however this has been refuted by many(eg see Antonisse 1989)Presently there are no guidelines for predicting which encodings work best.

Tree encodings

Tree encoding schemes like John Koza's scheme for representing computer programs, have several advantages including the fact that they allow the search space to be open ended. However this leads to some pitfalls

The trees can grow large in uncontrollable ways, preventing the formation of more structured hierarchical structures.(koza's 1992, 1994, "automatic definition of functions"

is one way in which a GA can be encouraged to design hieratically structured programs. There are as yet very nascent attempts at applying tree encodings to GAs.

How is one to decide which encoding to use for ones target problem? Lawrence Davis a researcher with much experience in GAs suggests that one uses what is natural to ones problem and then designing ones GA for that problem.. Most research is presently done by guessing the encoding and then adapting the GA to it. This is nor very different from machine learning approaches like for example encoding a neural network is done by trail and error. One appealing idea is however to have the encoding adapt itself to that the GA can make a better use of it.

Adapting the encoding

The creation of the ideal encoding for a problem is paramount to solving that problem itself. Usually genetic algorithms are used for hard problems without an easy solution. Apart from guessing the encoding allowing it to adapt is one solution.

The ordering of bits in the chromosome is another issue. If these bits that were co adapted were close together, then they would stay together in a crossover leading to fitter chromosomes. But we have no idea on how to best order the bits ahead of time for the problem. This is called the linkage problem in GA literature. A second reason for adapting the encoding is that a fixed length chromosome limits the complexity of the candidate solutions. It would thus be interesting to know if a chromosome were of variable length, what strategies would evolve. Such an experiment was done by Lindgren 1992, in which "gene doubling" and "deletion" operators allowed the chromosome length to increase or decrease over time. More examples of such work are described in Mitchell 1996.

Selection methods

Once the chromosomes are encoded. comes the next problem of selection, that is the criteria to select some n parents from the current population for crossovers and mutations. We encounter the classical exploitation/exploration dilemma. In the next few sections , is described some of the selection methods. For a detailed review see Mitchell 1996.

Fitness-Proportionate selection with "Roulette Wheel" and "Stochastic Universal" sampling.

The most common way of implementing selection is the roulette wheel where each individual is assigned a wedge in a circle proportional to their fitness

The wheel is spun N times where there are N parents. This method can be implemented as follows

- 1.Sum the total expected value of individuals in the population, Call this sum T.

- 2.Repeat N times: Choose a random integer between 0 and T

Loop through the individuals in the population, summing the expected values, until the sum is greater than or equal to r. The individual whose expected value puts the sum over the limit is the one selected. James Baker 1987, proposed a different sampling method. Rather than spin the roulette wheel N times to select N parents, SUS spins the wheel once, but with N equally spaced pointers, which are used to select the N parents. Baker (1987) gives the following code fragment for SUS

```
ptr = Rand();/*returns random number uniformly distributed in [0 1]*/
for (sum = i=0; i<N; i++)
for(sum += ExpVal(i,t); sum>ptr; ptr++)
```

Select (i);

Where i is an index over population members and where ExpVal(i,t) gives the expected value of individual i at time t.

SUS also suffers from premature convergence where the fitness variance of the population decreases rapidly leading to little or no exploration.

Sigma scaling:

To address the above problems, GA researchers have experimented with scaling methods, methods to map raw fitness values to expected values. One example is sigma scaling.

Under sigma scaling an individual's expected value is a function of its fitness, the population's mean and the population's standard deviation.

$ExpVal(i,t) = 1 + (f(i) - f(t)) / 2\sigma(t)$ if $\sigma(t)$ not equal to zero and
 $ExpVal(i,t) = 1.0$ if $\sigma(t) = 0$

here ExpVal(i,t) is the expected value of individual i at time t, f(i) is the fitness of i, $f_{-}(t)$ is the average fitness of the population at time t and $\sigma(t)$ is the standard deviation of the population fitness at time t.

For more details on sigma scaling refer to Mitchell 1996.

Elitism

Elitism first introduced by Kenneth De Jong 1975, many researchers have found this to improve performance.

It involves retaining verbatim some of the members of the population which can be destroyed by mutation and crossover.

Boltzman selection.

It may be advantageous to have a varied program of fitness selection over time with a liberal schedule early on so that the variance of fitness is large and in later stages to have the selection pressure up in selecting only the very fit individuals.

Boltzman selection like simulated annealing is such a scheme. It has a variable called temperature that can be increased over time, so the ExpVal takes on the form

$ExpVal(i,t) = e^{(f(i)/T)} / \langle e^{(f(i)/T)} \rangle$

where T is the temperature and $\langle \rangle$ is the average over the population at time t.

Rank selection

Rank selection prevents premature convergence, here the population at time t is ranked 1 to N rather than use the fitness values directly. Ranking avoids giving the far largest share of offspring to a few highly fit individuals thus reducing the selection pressure when the variance of fitness is high.

The linear ranking method proposed by Baker is as follows :

Each individual in the population is ranked in increasing order of fitness. The user chooses the expected value Max of the individual with rank n, and $max \geq 0$. The expected value of each individual at time t, is given by

$ExpVal(i,t) = Min + (Max - Min)((rank(i,t) - 1) / (N - 1))$

For a more complete analysis of ranking see Mitchell 1996

Tournament Selection

The above mentioned selection requires two passes over the population at each generation. One pass to compute the mean fitness and one pass to compute the expected value for each individual

Rank selection requires sorting the population by rank a time consuming procedure, Tournament selection is computationally more efficient and more amenable to parallel implementation. Two individuals are chosen at random from the population. A random number r is then chosen between 0 and 1. if $r < k$ where k is a parameter, the fitter of the two individuals is selected to be the parent otherwise the less fit individual is selected. The two are returned to the original population and can be selected again. An analysis of this method was presented by Goldberg and Deb(1991).

Steady-State selection

In steady state selection only few individuals are replaced in each generation usually a small number of highly unfit individuals.

For more details see Mitchell 1996.

Genetic Operators

The third decision in a GA is the nature of genetic operators to use on the parent population.

Apart from crossover and mutation a number of other operators are also mentioned.

This includes inversion and gene doubling. De Jong 1975, experimented with a crowding operator in which newly formed offspring replaced the existing individual most similar to itself, this prevented crowding.

Goldberg and Richardson 1987, accomplished a similar result by an explicit fitness sharing. In the case of fitness sharing the sharing was decreased with a factor that was directly proportional to the similarity of the between two individuals. They showed that in some cases this could induce speciation, allowing the population members to converge on several peaks in the fitness landscape rather than the population converging on one peak.

Another way to preserve diversity is by controlling the mating, several methods have been suggested, using mating tags, a kind of sexual selection where only individuals with similar tags are allowed to mate, preventing incest where similar individuals cannot mate and specism where only like individuals mate.

Finally there have been spatially restricted mating, the individuals are on a special lattice and mating can occur only with the neighbors in the lattice.

Parameters for genetic algorithms.

The forth and last step is to determine the parameters such as the population size, probability of crossover and mutation.

Much experimental work has been done on this issue a small part of which is described in this section

There is no theory to compute the parameters and it still remains an experimental task. De Jong performed much a investigation experimentally on a small test suite. His

experiments indicate that the best population size is 50-100

The best single point crossover rate was ~0.6 per parent and the best mutation rate was ~0.001 per bit.

Later, Grefenstette 1986, used a GA as a meta GA to optimize the GA parameters, his results were a population size of 30, crossover rate of 0.95 and a mutation rate of 0.01, the generation gap of one and used elitist selection. Schaffer, Caruana, Eshelman and Das spent over a year of CPU time investigating the issue, their results were population size 20-30, crossover rate 0.75-0.95, mutation rate 0.005-0.001.

Many people believe in a adaptable GA, where the parameters are dynamic and change over time. We describe Lawrence Davis's approach to self adapting operator rates (Davis 1989 1991). Davis assigns to each operator a fitness value based on how many extremely fit individuals that operator has produced. Operators get high fitness both for creating parents of highly fit individuals and for creating highly fit individuals.

The fitness of each operator over a given time interval is a sum of its previous fitness and the credits received from all the fitter than average individuals it has created.

A big question then for any adaptive approach to setting parameters is this: How well does the rate of adaptation of parameter settings match the rate of adaptation in the GA population?

The feedback for setting parameters comes from the populations success or failure with the fitness function. Very little work has been done on measuring these different rates of adaptation and how well they match in different parameter adoption experiments.

Ionic channels

Ionic channels are pores in cell membranes. They are the fundamental excitable members of excitable membranes. They are usually called transport mechanisms since they transport water and ions through them creating potential differences between the outer medium and inner medium of cells.

Transport as known to physiologists are by carriers or pores, ionic channels apart from gap junctions are the pores.

Excitation and electrical signaling in the nervous system involve the movement of ions in channels. The Na^+ , K^+ , Ca^{++} and Cl^- ions seem to be responsible for all the action.

The channel may be regarded as an excitable molecule specifically sensitive to a membrane potential or to a particular molecule like a neurotransmitter.

The channels response called gating is a simple closing or opening of the pore. The open pore has selective permeability allowing a small class of ions to flow (at rates $>10^6$)

Ion movement across a pore causes an electric current to flow, this in turn changes the membrane potential, other channels sense this potential and open to generate and regenerate the pulse in the membrane potential, thus ion channels create solitons for the propagation of action potentials. Electricity is thus a means to carry a signal to the point where a non electrical response is generated. The membrane potential opens or closes a Ca^{++} channel. This increases the cells inner free concentration of Ca^{++}

This in turn triggers secretion of neurotransmitters, neurohormones, digestive enzymes etc. The study of Na^+ , K^+ and Ca^{++} and Cl^- ions leads to the creation of a current

$I_{\text{Na}}, I_{\text{K}}, I_{\text{Ca}}, I_{\text{Cl}}$. By applying Ohm's law,

we know that $I = g * E$ where g is the conductance measured in siemens.

Thus the total current $I = I_{\text{Na}} + I_{\text{K}} + I_{\text{Ca}} + I_{\text{L}}$, where I_{L} is the leak current.

Also by Ohm's law, $E = IR$ where R is the resistivity measured in ohms and we have

$$R = (\rho * l) / A.$$

In addition to the membrane current, the membrane acts as an insulator between the extra cellular fluid and the internal

fluid of a cell, hence defining a capacitance.

$C = Q/E$ where Q is the charge transferred and E the potential.

Thus

$$C = \epsilon * \epsilon_0 * A / d$$

by ohms law ,

the solution is $E = E_0 * \exp(-t / (RC)) = E_0 * \exp(-t / \tau)$

Equilibrium potentials.

Diffusion creates a charge transfer or a directional current till equilibrium is reached, this leads to a resting potential across the cell.

There are separate resting potentials for different ions, E_K, E_{Na}, E_{Ca} are the potentials for potassium, sodium and calcium.

By boltzman relation

by applying this to a cell membrane we get the nerst equation

$$E_s = E_I - E_2 = (RT / (ZsF)) \ln([S_2] / [S_1])$$

thus

$$E_k = (RT / F) \ln([K_o] / [K_i]) \text{ and similarly for } E_{Na}, E_{Ca}, E_{cl}$$

and this potential is like a battery so, $I_K = g_K (E - E_K)$ however this linearity is empherical, some channels are known to have a nonlinear I,E relationship.

The Hodgkin-Huxley (HH) model

The HH model assumes a maximum conductance g_K for g_K and a parameter n that takes on a value between 0 and 1. Similarly for g_{Na} two particles are assumed to exist, m and h .

so for a simple axon with two channels K and Na channels,

$$I_k = n^4 * g^k * (E - E_k)$$

and $I_{Na} = m^3 * h * g^{Na} * (E - E_{Na})$

$$I_i = m^3 * h * g^{Na} * (E - E_{Na}) + n^4 * g^k * (E - E_k) + g^L * (E - E_l)$$

so where:

n, m and h have first order kinetics described by $dn/dt = (n_\infty - n) / \tau_n$ and similar definitions for m and h .

The Fitzhugh-Nagama model

A simplification of the Hodgkin Huxley model, This model seeks to explain the action potential as a positive feedback factor and a negative feedback recovery of the membrane gate potential.

$$dV/dt = V - V^3 - w + I_{ext}$$

$$\tau dw/dt = V - A - b * w$$

The parameters $a=-0.7$, $b=0.8$ $\tau=1/0.08$

We have an equation of the membrane voltage V , external current and a slower gate voltage w .

Morris-Lecar model

Morris and Lecar combined the Hodgkin-Huxley and the FitzHugh-Nagumo into a voltage gated calcium channel model with a delayed rectifier potassium channel represented by

$$C dV/dt = -I_{ion}(V, w) + I_{ext}$$

$$dw/dt = \phi(w_{\infty} - w) / \tau_w \text{ where}$$

$$I_{ion}(V, w) = g_{Ca} * m_{\infty} * (V - V_{Ca}) + g_K * w * (V - V_K) + g_L * (V - V_L)$$

Hindmarsh-Rose model

Building upon the FitzHugh-Nagumo model, Hindmarsh and Rose proposed a model of neuronal activity based on three first order differential equations.

$$dx/dt = y + 3 * x^2 - x^3 - z + I_{ext}$$

$$dy/dt = 1 - 5 * x^2 - y$$

$$dz/dt = r * (4 * (x + 8/5) - z)$$

with $r^2 = x^2 + y^2 + z^2$ and $r^2 \text{ approximates } 10$ $r \text{ Sup } \{ \text{size } 8 \{ - 2 \} \} \{ \}$ so that the variable z changes very slowly.

It is a simple model with a rich variety of dynamical behaviours for the membrane potential described by the variable x . This also includes chaotic behavior.

There is a vast diversity of potassium channels.

The most impressive diversity is in voltage dependant potassium channels. Most open only after the membrane is depolarized but some only after it is hyperpolarised. Some open rapidly and some slowly. Some are also modulated by other influences like neurotransmitters or intracellular messengers. Although K^+ ions are the major current carrier, the pharmacology is so different in different channels that it requires that fundamentally different channels are involved. Each membrane uses a mix of these channels to fulfill its needs. K channels can regulate pacemaker potentials, generate spontaneous trains and bursts of action potentials or make long plateaus on action potentials.

K channels stabilize the resting potential, they draw the potential more towards the potassium equilibrium position and farther away from the threshold of firing.

The roles of all K channels is related to this stabilisation. Potassium channels set the resting potential, keep fast action potentials short, terminate periods of intense activity slow the rate of repetitive firing and generally lower the excitability of the cell when they are open. The roles given here are only examples, many roles for potassium channels yet remain to be discovered.

We consider seven main K channels: The delayed rectifier, the calcium dependant K(Ca)channel, the A channel, inward rectifier channel, outward rectifying channel, Slowly activating channel and the silencer channel.

Delayed rectifiers keep action potentials short(Hille B. 1984)

Brevity of action potentials in axons requires the rapid inactivation of Na channels and a high K permeability. This permeability of K is by the delayed rectifier channel. The delayed rectifier is not one channel but a group of similar functional channels.

$K_v\alpha 1.x$ - Shaker-related: $K_v1.1$ ([KCNA1](#)), $K_v1.2$ ([KCNA2](#)), $K_v1.3$ ([KCNA3](#)), $K_v1.5$ ([KCNA5](#)), $K_v1.6$ ([KCNA6](#)), $K_v1.7$ ([KCNA7](#)), $K_v1.8$ ([KCNA10](#))

$K_v\alpha 2.x$ - Shab-related: $K_v2.1$ ([KCNB1](#)), $K_v2.2$ ([KCNB2](#))

$K_v\alpha 3.x$ - Shaw-related: $K_v3.1$ ([KCNC1](#)), $K_v3.2$ ([KCNC2](#))

$K_v\alpha 7.x$: $K_v7.1$ ([KCNQ1](#)) - [KvLQT1](#), $K_v7.2$ ([KCNQ2](#)), $K_v7.3$ ([KCNQ3](#)), $K_v7.4$ ([KCNQ4](#)), $K_v7.5$ ([KCNQ5](#))

$K_v\alpha 10.x$: $K_v10.1$ ([KCNH1](#))

Transient outward currents space repetitive responses(Hille B,1984)

Axons blindly propagate action potentials given to them, the potentials originate from axon hillocks, cell bodies, dendrites or sensory receptor terminals.

They transform the sum of all graded intrinsic and extrinsic, excitatory and inhibitory influences into a code of action potential firing.

This leads us to believe that a frequency based code of action potentials is the method of functioning of the nervous system. Thus a frequency modulated code is derived for the neurons. However most axon membrane could not be used for graded rhythmic encoding because in the face of stimulus current they fire only once and then remain refractory or fire repetitively at a very high frequency that varies little with the stimulus current. Encoding membranes fire with a frequency modulated code and are slow enough(1 to 100 Hz) not to exhaust nerves and muscles. Encoding membranes have one another K channel, it activates in the sub threshold regions of membrane potentials (Hagiwara et al, 1961;Connor and Stevens, 1971 a, b, c;Neher, 1971).We will use the term A current and IA for this channel. The A channel can be activated when a cell is depolarized after a period of hyperpolarisation.

The kinetics of A channels, analogous to the HH model are described to be a^4bg_- -A(Connor and Stevens, 1971c;Smith 1978) or a^3bg_- -A(Neher 1971). The voltage dependence of inactivation is quite steep, with a midpoint near -70mV and falling almost to zero at -40mV.Activation is also steep and occurs at potentials more positive than -65mV.Thus in steady state this channel only conducts within a narrow window of negative potentials(-65 to -45 mV).At the typical resting potential in these cells (-45mV)most of the channels are inactivated.

At the end of the first action potential A channels are all inactivated but K channels are so activated that the cell hyperpolarizes, this in turn activates the

A channels which fire a current that more or less balances out the applied depolarization, however the A channels soon deactivate leading to the voltage reaching firing threshold resulting in an action potential. Thus A channels serve as a damper in the interspike

interval to space pulses more widely.

$K_v\alpha 1.x$ - Shaker-related: $K_v1.4$ ([KCNA4](#))

$K_v\alpha 3.x$ - Shaw-related: $K_v3.3$ ([KCNC3](#)), $K_v3.4$ ([KCNC4](#))

$K_v\alpha 4.x$ - Shal-related: $K_v4.1$ ([KCND1](#)), $K_v4.2$ ([KCND2](#)), $K_v4.3$ ([KCND3](#))

Ca-dependant K currents make long hyperpolarizing pauses(Hille. B.1984)

In this example we consider spontaneously active cells like the bursting pacemaker cell. The fire relatively complicated patterns of bursts followed by silent periods.

This oscillation is intrinsic even in the absence of excitatory and inhibitory input. Several authors have suggested that the slow bursting rhythm is from cyclical variations of intracellular free Ca^{++} . (Meech et al 1975;Eckert et al 1976;Smith 1978, Gorman et al 1978,1981).During a burst Ca^{++} ions enter with each action potential far faster than the cell can reabsorb it, this leads to the calcium dependant K channels to activate. They hyperpolarize the cell and shut off calcium entry. The calcium load is cleared out and this leads to a new cycle of bursts after the K channel inactivates. The details are complex as at least seven different channels are active.

K channels, A-channels, Ca dependant K channels, Na channels, Ca channels, leak channels and a very slowly gated channel conducting Na and Ca ions.

We call the $K(Ca)$ current as $IK(Ca)$. Except for the general statement that $IK(Ca)$ serves

to terminate periods of Ca^{++} entry, by hyperpolarizing the cell, its actual physiological function needs to be elucidated. In most cases, $\text{K}(\text{Ca})$ channels have an intrinsic voltage dependence apart from the voltage dependence of calcium ions.

At a fixed value of $[\text{Ca}^{++}]_i$, $\text{K}(\text{Ca})$ channels behave like delayed rectifier channels activating with depolarization's. (Gorman et al 1980 Pallota et al 1981; Latorre et al 1982) The gating is extremely complicated at the microscopic level (Hille B 1984). The physiological kinetics of $\text{K}(\text{Ca})$ channels is a blend of Ca^{++} dynamics, buffering, re uptake and storage, Ca binding and voltage dependent transitions. The effective kinetics should be faster in cells with a higher surface to volume ratio where variations of $[\text{Ca}^{++}]_i$ would be quickest. The response would always be a depolarization or a hyper polarization of the cell when the calcium level rises above a certain level.

Outward-rectifying

$\text{K}_v\alpha 10.x$: $\text{K}_v10.2$ ([KCNH5](#))

Inward rectifiers permit long depolarization's responses (Hille. B. 1984)

Three unusual properties distinguish inward rectifier potassium channels from other known channels.

1. They open with steep voltage dependence on hyperpolarisation.
2. The voltage dependence of their gating depends on extra cellular $[\text{K}^+]_o$ shifting on the voltage axis with the quality $RT \ln[\text{K}]_o$
3. Part of their steep rectification seems instantaneous, occurring in less than 1ms. an additional fraction develops exponentially with a time constant of many milliseconds to 0.5 seconds.

$\text{K}_v\alpha 11.x$ - [ether-a-go-go potassium channels](#): $\text{K}_v11.1$ ([KCNH2](#)) - [hERG](#), $\text{K}_v11.2$

Fig 1: Fitness as a function of no of generations or iterations of the genetic algorithm.

([KCNH6](#)), $\text{K}_v11.3$ ([KCNH7](#))

Slowly activating

$\text{K}_v\alpha 12.x$: $\text{K}_v12.1$ ([KCNH8](#)), $\text{K}_v12.2$ ([KCNH3](#)), $\text{K}_v12.3$ ([KCNH4](#))

Modifier/silencer

Unable to form functional channels as homotetramers but instead heterotetramerize with $\text{K}_v\alpha 2$ family members to form conductive channels.

$\text{K}_v\alpha 5.x$: $\text{K}_v5.1$ ([KCNF1](#))

$\text{K}_v\alpha 6.x$: $\text{K}_v6.1$ ([KCNG1](#)), $\text{K}_v6.2$ ([KCNG2](#)), $\text{K}_v6.3$ ([KCNG3](#)), $\text{K}_v6.4$ ([KCNG4](#))

Plant action potentials.

Plant action potential is the electric excitability of the membrane of plant cells. It consists of the rapid decrease of the negative membrane potential followed by a slower repolarization. Sufficient similarities exist between plant and animal action potentials proving that it is one and the same phenomenon (Beilby M J).

The Hodgkin Huxley model influenced the early thinking on plant action potentials. However there are marked differences between the action potentials in plant and animal cells. Plant action potentials are 10^3 times slower than animal ones. In plants there are two membranes, the outer plasmalemma and the inner tonoplast. Both these membranes are excitable. This may however be the primary reason why plant action potentials are highly variable. The plant potential remains mostly in the negative region, becoming positive only in unusual circumstances. The outflow of chloride ions rather than sodium marks the depolarizing phase of the action potential. Calcium ions play a very important role in plant action potentials.

Owing to the large size of their cells, a vast repository of literature exists on the charophyte algae. These results are also applicable to embryophytes and thus to land

Fig 2: results of the search algorithm over 100 generations for various current injections from 0.1 nA to 1.0 nA.

plants. Action potentials play an important role in signal transmission not only in animal muscles and nerves and the nervous system but also in higher plants like the touch sensitive mimosa pudica and the carnivorous plant dionea muscipula, which respond to mechanical stimuli, where action potentials are evoked that travel to motor tissues where turgor related movement is initiated along with cytoskeletal movement.

Models of plant neurobiology.

The equations describing the Na^+ transient in the nerve have been adopted to describe the Cl^- transient and the Ca^{2+} transient.

$I_{cl} = g_{cl} * (E_m - E_{cl})$, where I_{cl} is the chloride current, g_{cl} is the chlorine conductance E_m is the membrane potential and E_{cl} is the equilibrium potential for Cl^-

$$g_{Cl} = m^2 * h * g_{max}$$

here g_{max} is the maximum conductance m and h are activation and inactivation parameters respectively

$$m = m_{\infty} - (m_{\infty} - m_0) \exp\left[-(\delta_{cl} - t) / \tau_m\right]$$

$$h = h_{\infty} - (h_{\infty} - h_0) \exp \left[(\delta_{cl} - t) / \tau_h \right]$$

where t is the time and δ_{cl} is a delay before excitation begins, time constants

τ_m and τ_h The above equations describe how m and h evolve from their initial values m_0 and h_0 respectively. Similar set of equations was used for the second transient, as the membrane depolarises the outward rectifier is activated and K^+ flows out restoring the PD (together with Cl^- inactivation)

The AP can be reconstructed by

$$dV/dt = (1/C) * [g_{cl} * (E_m - E_{cl}) + g_{Ca} * (E_m - E_{Ca}) + I_{ss}]$$

where I_{ss} is the steady stream current which was measured at every PD level.

The resting PD restoring current activates more gradually and is fitted with a similar H.H equation.

$$I_k = g_k * n^4 * (V - V_k),$$

$$n = n_{\infty} - (n_{\infty} - n_0) \exp \left[(\delta_k - t) / \tau_n \right]$$

where g_k is the maximum K^+ conductance, V_k is the equilibrium PD and n the inactivating parameter. The delays in excitation indicate the existence of intermediate steps before excitation channels can be activated, these ideas are compatible with data on the cytoplasmic second messenger taking part in the plant AP and newer models to describe them.

Newer approaches: (Beilby M J)

IP3 (inositol 1,4,5 -triphosphate) and Ca^{2+} from internal stores.

The experiments with Mn^{2+} as a quencher for fura -2 fluorescence (Plieth et al 1998) confirmed the importance of internal stores of Ca^{++} . Further the elevation of IP3 in the cytoplasm was able to elicit APs (Thiel et al 1990).

The enzyme phospholipase C (PLC) is responsible for mobilising IP3 from its membrane bound precursor phosphatidyl inositol 4,5 biphosphate (PIP2), Inhibitors of PLC , neomycin and U73122 inhibited excitation in Chara (Biskup et al 1999).

The Ca^{2+} concentration in the cytoplasm was monitored while depolarizing pulses were applied. The concentration of Ca was all or none, with a sharp threshold for CA activity with respect to the duration and strength of the applied pulses. (Wacke and Thiel, 2001) Two consequent sub threshold pulses elicited a full response.

This experiment supports the hypothesis that second messenger activity such as IP3 is produced in response to a depolarizing current stimulus and the IP3 produced should reach a threshold for Ca activity to be induced. The double sub threshold response was valid only if the two pulses were separated within a time scale of 3 seconds, well within the IP3 characteristic time scales. The dynamics of IP3 production from the PIP2 pool and the decay to IP2. If the second pulse came too close to the first no response was obtained owing possibly to a depleted PIP2 inertia.

Wacke et al 2003, borrowed another animal model (Othmer 1997) to describe in a quantitative model of Chara AP mediated by IP3 activated Ca^{2+} channels and pumps on the endomembranes of internal stores such as the ER.

Othmer described the IP3 mediated Ca channels to have four states.

1 unbound

2 bound to IP3

3 bound to IP3 and activating Ca modeled

4 bound to IP3 and to a second inactivating Ca 2+

This channel is active only in step 3 and is inactivated by a further increase in Ca concentration.

Wacke used a sophisticated periodic stimulation technique to validate the fact that the inactivation was due to step four, a model constructed with parameter values from Othmers animal model , modeled the time period of inactivation to be 30 seconds.

Although this modeling looks encouraging, Tawana and Kikuyama failed to duplicate the experiments in Chara. The injection of IP3 into the cytoplasm failed to elicit Ca2+ release and the inhibitors of PLC, U73122 and neomycin did not affect the AP.

An inhibitor of the IP3 receptor 2APB did make cells in excitable, but the same effect was noted in tonoplast free cells where external CA does the signaling. The authors suggest that the inhibitors affect the external plasma membrane rather than the internal membrane, Clearly more studies are warranted to resolve this.

The fitness function.(Vainer and Bower 1999)

The fitness function was computed as the inverse square root of match, the floating point result of comparison between the experimental spike train and the computed spike trains of candidate solutions.

The spike timing match function is as follows:

$$match = 1/N * \sum \left(\sum \left[\theta(i,n) * Mod(t_{(i,n)} - o_{(i,n)}) / A_{(i,n)} + B_{(i,n)} \right] + C_n \right)$$

here $t_{(i,n)}$ is the time of the i th spike in trace n of the target data and $o_{(i,n)}$ is the i th spike in trace n of model data. N is the total number of spikes in all the traces. The first sum is over the traces and the second sum is over the spike times in individual traces.

$\theta(i,n)$ equals one if both traces n have spike i , otherwise it is equal to zero.

$$A_{(i,n)} = \left(\max(0.010, o_{(i,n)}) \right)^{pow1}$$

$$B_{(i,n)} = \left(1 - \theta(i,n) \right) * P / \left(TC_n \right)^{pow2}$$

$$C_n = Q * \sum \left(h \left(ISI_{(t-1)} - ISI_i \right) \right)$$

Term A weighs spike mismatches more heavily the closer they are to the start of the current injection. Pow1 is a free parameter and is assigned 0.4. Term B penalizes missing or extra spikes in the simulated trace as compared to the original trace. P was set to 0.5sec. Tcn is the total number of spikes in target trace n . Pow2 is a free variable we set to 0.6. Term C is a non monotonicity penalty penalises consequent inter spike intervals that are not monotonically increasing. function $h(x)$ is zero if $x < 0$ and is x otherwise. The parameter Q is a penalty value for non monotonically increasing ISI and is set to 200 sec^{-1}

Waveform match function(Vernier and Bower 1999)

$$\text{waveform match} = 1/(M*N)\text{sum}(\text{sum}(t_i - o_i)^2) \quad \text{match} = 1/(M*N) * \sum \left(\sum (t_i - o_i)^2 \right)$$

where o_i, t_i represent the target and test waveforms sampled at 0.1 msec, N is the total number of points in each waveform sampled and M is the total number of waveforms generated by the model.

Model:

The model of a single cell of an algae or higher plant consists of a single compartment with a variety of ion channels. A voltage dependant calcium channel and a calcium dependant chloride channel form the depolarizing terms, two potassium channels one delayed rectifier and another, an M current of non inactivating channels form the hyperpolarizing terms. The calcium concentration is modeled to exponentially decay in concentration. $d[Ca]/dt = B * I_{Ca} - [Ca]/\tau$ where τ and B are parameters. (Vanier and Bower 1999)

Results:

Param	Varied?	Type	Range	Best	Label
-----	-----	----	----	-----	
0	YES	mult	4	1.69447	Ca Gbar scaling
1	YES	mult	4	1.03044	Ca tau scaling
2	YES	add	0.01	-0.000308393	Ca minf offset
3	YES	mult	4	1.71906	Cl Gbar scaling
4	YES	mult	4	0.263561	Kdr Gbar scaling
5	YES	mult	4	0.27342	KM Gbar scaling
6	YES	mult	4	3.67349	KM tau scaling
7	YES	add	0.01	-6.85149e-05	KM minf offset

Table 1: The table indicates the parameters varied in the search and publishes the best parameters of the genetic algorithm.

Param	Correct value	

0	1.285714	Ca Gbar
1	3.0	Ca tau
2	0.05	Ca minf

3	1.568	Cl Gbar
4	0.526316	Kdr Gbar
5	0.4285714	Km Gbar
6	3.00	Km tau
7	0.005	Km minf

Table 2: The parameters of the original cell that is used in the spike comparison.

Fig 2 shows the waveforms of the genetic algorithm's best matching solution and the solution from the original model. The similarity of the best waveforms for various current injection from 0.1 to one nanoampere to the original data proves the genetic algorithms efficacy. An analysis of the population of the genetic algorithm follows.

Fig 3: Non elitist simulation to 100 iterations

Fig 4: elitist of five members, over 100 generations.

Fig 5: Graph of fitness versus generations for ten generations with the cross over probability of 0.5

Fig 6: Graph of fitness versus generations for ten generations with the cross over probability of 0.6

Fig 7: Graph of fitness versus generations for ten generations with the cross over probability of 0.7

Fig 8 : Graph of fitness versus generations for ten generations with the cross over probability of 0.8

Fig 9: Graph of fitness versus generations for ten generations with the cross over probability of 0.9

Fig 10: Graph of fitness versus generations for ten generations with the cross over probability of 1.0

Population Size

Fig 11, fitness versus no of generations for a population size of 10

Fig 12, fitness versus no of generations for a population size of 20

Fig 13, fitness versus no of generations for a population size of 30

Fig 14, fitness versus no of generations for a population size of 40

Fig 15 fitness versus no of generations for a population size of 50

Fig 16, fitness versus no of generations for a population size of 60

Fig 17, fitness versus no of generations for a population size of 70

Fig 18, fitness versus no of generations for a population size of 80

Fig 19, fitness versus no of generations for a population size of 90

Fig 20, fitness versus no of generations for a population size of 100

Fig 21 : Fitness versus generations for a probability 1.0 of mutation per bit.

Fig 22: Fitness versus generations for cross over type 0 and cross over number 2.

Tables 1 & 2 list the goal parameters of the ideal cell and the parameters of the best of the candidate solutions. A difference in the values indicates that the space of fitness is large and has many solutions in parameters of the resulting voltage waveform.

The table also lists the search parameters, being the maximum conductances of all the channels and the Tau and minf parameters of the Ca channel and the K channel.

Parameters of the genetic algorithm.

The genetic algorithm consists on a population size parameter number of candidate solutions, These candidate solutions are a table of parameters of the neuronal model. Each parameter is a floating point number four bytes in size.

The parameters are so arranged such that each ions parameters for the channel or concentration term are adjacent to each other. These parameter tables are randomly seeded at the start of the simulation. The neuronal model is computer per table of parameters and a fitness function as defined above used to calculate the fitness of each candidate solution. An Elitist method is used to retain one of the best solution and the

best fitness solutions are used to generate by cross over the next generation of solutions. The crossover is of order N meaning that there are N crossovers per two chosen tables, crossovers are only at byte boundaries meaning that a floating point number has no crossover on it, but only at byte boundaries in the parameter table.

There are two parameters to the crossover function, crossover type and cross over number, Cross over type means there are many cross overs of order one and more and an average of N crossovers is produced where N is the cross over number for the crossover type equal to zero, for crossover type equal to one exactly N crossovers take place per pair of tables, where N is the crossover table.

Besides these parameters are the crossover probability and mutation probability constants, These determine the probability of crossover and the probability of mutation per bit of the chromosomes.

Simulations were run for population sizes from 10 to 100 in multiples of 10, for various values of the crossover probability, from 0.5 to 1.0 in units of 0.1, for a mutation rate of 1.0 and with a cross over type of zero with two crossovers per pair of chromosomes, also were run, a non elitist simulation and an elitist with five conserved solutions for a comparison.

Population size:

The population size was varied from 10 to 100 in steps of 10 for a period of 10 generations. The trade off in the population size is in the lack of biodiversity in small populations to the computational overhead of large population sizes. It was found in the simulations run that small populations do not have the diversity of solutions and consequently take a larger time to saturate to a good solution. Larger populations take a shorter time to converge to an optimal solution. Premature convergence in larger populations could also happen. It is elsewhere (??) recommended that a population size of 30-50 is optimal , we have however found 100 to be optimal without causing too high a load in computation cycles. Figures 11 to 20 indicate the fitness versus generation number for a population size from 10 to 100 in steps of 10. Smaller population sizes like 30-50 have a lower fitness measure in 10 simulations, the solutions they converge to are also sub optimal. Population sizes in the 70 to 100 range are much better and rate in higher fitness numbers in the convergence.

Figure 3 is a non elitist simulation where none of the best fit candidate solutions are retained. It is obvious from Figure 3 that elitism is essential or the genetic algorithm does not converge to a solution but instead fluctuates in the process losing all best fit candidates, on the other hand the degree of elitism , whether five or one does not make much of a significant difference. Figure 4 is a fitness versus generations plot with an elitism of five best fitting candidate solutions, The performance of the genetic algorithm is much to the default of one candidate solution retained across generations.

Figure 5 to figure 10 are simulations run with various values of crossover probabilities. Starting from 0.5 in figure 5 to 1.0 in figure 10, in steps of 0.1.

The cross over probability is a trade off between retaining good solutions in the genetic pool rather than reproducing and changing the entire genetic pool every generation. It is essentially a preservation of individuals versus modifying them in crossovers. The simulation from figure 5 to ten indicate a best performance with a probability between 0.7 and 0.9.

Figure 21 is a simulation run with the mutation per bit equal to 1.0 , this makes the search into a kind of steepest hill ascent. From the figure it is obvious that the local maxima alone is achieved, and more global maximum in fitness is not achieved. The mutation is useful in altering the genetic pool through mutations and the best value for mutation so that local maxima are not converged into is about 0.01-0.05.

The mutation rate does not seem to affect the performance of the genetic algorithm.

Figure 21 is a simulation with cross over type zero and crossover number equal to 2.

This means that there exactly 2 crossovers between any two candidate solutions.

From the figure we deduce that there is not much of a difference in the functioning of the genetic algorithm.

References

- Baker J E 1987 Reducing bias and inefficiency in the selection algorithm. In JJ Grefensette ed, Genetic Algorithms and their applications
 Proceedings of the Second International conference on genetic algorithms, Erlbaum.
- Bielby M J , Action potential in Charophytes, www.phys.unsw.edu.au/~mjb/AP%20review.pdf
- Biskup B, Gradmann D and Thiel G (1999) Calcium released from IP3 sensitive stores initiates action potential in Chara. FEBS Letters 453, 72-76
- Connor J.A and C.F Stevens, 1971a Inward and delayed outward membrane currents in isolated neural somata under voltage clamp. J. Physiol. (Lond.) 213, 1-19 [2,5]
- Connor J.A and C.F Stevens, 1971b Voltage clamp studies of a transient outward membrane current in gastropod neural somata. J.Physiol.(Lond)213, 21-30[5]
- Connor J.A and C.F Stevens, 1971c. Prediction of repetitive firing behaviour from voltage clamp data on an isolated neuron soma. J. Physiol. (Lond.) 213, 31-53
- Davis L D 1989 Adapting operator probabilities in genetic algorithms. In J.D.Schaffer, ed., Proceedings of the third international conference on genetic algorithms. Morgan Kaufmann.
- Davis.L.D, ed 1991,, Handbook of genetic algorithms. Van Nostrand Reinhold.
- De Jong K.A. 1975. An analysis of the behavior of a class of genetic adaptive systems, PhD thesis, University of Michigan
 Ann Arbor.
- Eckert. R.and H.D. Lux, 1976 A voltage sensitive persistent calcium conductance in neuronal somata of Helix, J. Physiol.(Lond) 254,129-151.
- Goldberg D.E. and Deb K. 1991 A comparative analysis of selection schemes used in genetic algorithms,in Rawlis G.,ed,Foundations of genetic algorithms Morgan Kaufmann.
- Gorman [A.L.F](#), A. Hermann and M.V Thomas, 1981, Intracellular Calcium and the control of neuronal; pacemaker activity. Fed Proc. 40,2233-2239
- Gorman [A.L.F](#) and [M.V Thomas](#), 1978. [Changes in the intracellular concentration of free calcium ions in a pace maker neuron, measured with the metallo-chromic indicator dye arsenazo III](#) J.Physiol (Lond) 275, 357-376
- Grefenstette J.J 1986. Optimization of control parameters of genetic algorithms. IEEE

transactions on Systems, man and Cybernetics

16,no 1,, 122-128

Grefenstette J.J 1993 Deception considered harmful. In L.D. Whitley, ed, Foundations of genetic algorithms 2.Morgan kaufmann.

Hagiwara S , K. Kusano and N.Saito, 1961, Membrane changes of Onchidium nerve cell in potassium rich media. J. physiol. (Lond) 155, 470-489

Hille Bertil, 1984, Ionic channels of excitable membranes. Sinauer associates inc.

Holland J.H 1975. Adaptation in natural and artificial systems.University of Michigan press.

Lindgren K. 1992 Evolutionary phenomena in simple dynamics. In C.G.Langton, C.Taylor, J.D Farmer and S.Rasmussen, eds, Artificial Life II. Addison-Wesley
Meech R.W and N.B Standen,1975 Potassium activation in *Helix aspersa* neurons under voltage clamp; A component mediated by Calcium influx. J.Physiol. (Lond) 249m 311-239.

Mitchell M. 1996 An Introduction to Genetic Algorithms. MIT press

Neher E 1971, Two fast transient current components during voltage clamp on snail neurones. J. Gen. Physiol. 58, 36-53.

Othmer H.G. (1997) Signal transduction and second messenger systems. In 'Case studies in Mathematical modelling- Ecology , Physiology amd Cell Biology" Prentice Hall, Englewood Cliffs, NJ.

Schaffer J.D, Caruana R.A. Eshelman L.J. and DAs, R. 1989 A study of control parameters affecting online performance of genetic algorithms for function optimization. In J.D Schaffer, ed Proceedings of the third international conference on genetic algorithms. Morgan Kaufmann.

Smith S.J 1978, The mechanism of bursting pace maker activity in neurons of the mollusc *Tritonia diomedea*. PhD thesis. University of washington,University microfiles, Ann Arbor Mich.

Thiel G , MacRobbie E. A. C, and Hanke D. E Raising the intracellular level of inositol 1,4,5 – triphosphate changes plasma membrane ion transport in characean algae. EMBO J. 9, 1731-1741

Thierens, D. and Goldberg D.E. 1993, Mixing in genetic algorithms. In S Forrest, ed, Proceedings of the fifth international conference on genetic algorithms. Morgan kaufmann

Vanier M, Bower J (1996) A comparison of automated parameter searching methods for neural models. In: JM Bower, ed. Proceedings of the 1995 Computational Neuroscience Conference (CNS*95), Monterey (CA). Academic Press, New York.

Wacke M, Thiel G, (2001)Electrically triggered all or none Ca^{2+} liberation during action potential in the gaint algae *Chara*. J Gen. Physiol 118, 11-12

Wacke M , Thiel G and Hutt M. T.(2003) Ca^{2+} dynamics during membrane excitation of gaint algae *Chara*. Model simulations and experimental data. J Membr. Biol. 191, 179-192.