Chapter 4

Sunburn: Coevolving Strings

Sunburn is a model for designing a video-game spacecraft, which was proposed by John Walker. The model has a fitness function that is predicated on winning battles with other spacecraft. The fitness function explores the space of designs for spaceships without using actual numbers; it just uses win, lose, or tie results. The genes for spaceships are character strings augmented by an integer. There are two important new ideas in this chapter. First, the model of evolution used for Sunburn involves a strange variant on tournament selection, termed *gladiatorial tournament selection*. Second, the strings used as genes in Sunburn are tested against other strings. This means that Sunburn is a *coevolving* evolutionary algorithm.

Somewhat surprisingly, in many cases the Sunburn model behaves like a unimodal optimizer rather than a coevolving system; evolution does not appear to be contingent on the initial conditions. This suggests that for those choices of parameters the system is quite likely to be very close to a unimodal optimization problem.

4.1 Definition of the Sunburn Model

The gene for a spacecraft in the Sunburn model has a fixed number of slots for ships systems (the character string) together with a desired range the spacecraft wishes to be from an opponent during a battle (the integer). There are five types of systems, guns, lasers, missiles, drives, and shields. Figure 4.1 shows a typical gene. We will use a value of 20 system slots and choose ranges between 1 and 20.

Systems	Preferred Range
GLMDSSMGLSDSMLGMLSDS	16

Figure 4.1: A Sunburn gene

In biology, the map from genes to creature can be quite complex; in most artificial life systems, it is quite direct. The process of transforming a gene into a creature is called the *developmental biology* of a creature. In Sunburn we have out first nontrivial developmental biology. The sense behind this developmental biology will become apparent as we describe the rules for combat and is discussed explicitly in Section 4.3.

To derive a spaceship from a Sunburn gene, you count the number of loci in which an S appears and then put 3 times that many shields on the front of the ship. The remaining systems, G(uns), L(asers), M(issles), and D(rives) are placed behind the shields in the order in which they appear in the gene. Subscript the ship with its preferred range. The ship described by the gene in Figure 4.1 looks like:

$$SSSSSSSSSSSSSSSSSSGLMDMGLDMLGMLD_{16}$$

with the front of the ship being to the left. The *shield multiplication factor* is the number of shields a ship receives for each genetic loci in which it has an S. In this example and in most of this chapter, the shield multiplication factor is 3.

Combat between two ships is conducted as follows. Combat is initiated at a starting range. (We will explore different methods of generating starting range.) Once combat has started, the ships iteratively go through turns consisting of shooting and then moving. This loop continues until a winner and loser are found or until a draw occurs. The two ships shoot simultaneously using each gun, missile, or laser system once each turn. Each of the three types of weapons has an *effectiveness curve* that specifies its probability of scoring a hit at each possible range. These effectiveness curves are a very important feature of the Sunburn model. Examples of gun, laser, and missile effectiveness curves are shown in Figure 4.2.

The design of the effectiveness curves was inspired by imagining how space weapons might work. A gun emits a physical missile fired at very high velocity but with no guidance beyond initial efforts at aiming. The chance of hitting a target with a gun thus drops off with distance, and guns are most effective at short range. A laser moves at an incredible velocity (the speed of light) in a very straight line and so is equally effective at all ranges where a target can be detected. It does not have a high energy content compared to the kinetic energy locked in a physical projectile like a missile or shell and so has a relatively low kill rate. A missile, like the shells fired by a gun, is a physical projectile. Unlike a shell, it picks up speed throughout its attack run and has target seeking qualities. At short range (low velocity), missiles are relatively easy to intercept. Their effectiveness thus climbs as they approach top velocity. This conception is helpful in framing the initial analysis of the behavior of the system. Other conceptions and sets of weapons effectiveness curves that match them are certainly possible.

Each hit scored removes one system from the front of a ship. After shooting, the ships move. Each drive enables a ship to move a distance of one. The ships take turns moving a distance of one towards their preferred ranges, dropping out when they run out of drives.

In order to win a combat, a ship must destroy all the systems on the other ship and have at least one drive left itself. In the event that neither ship has remaining drives or neither

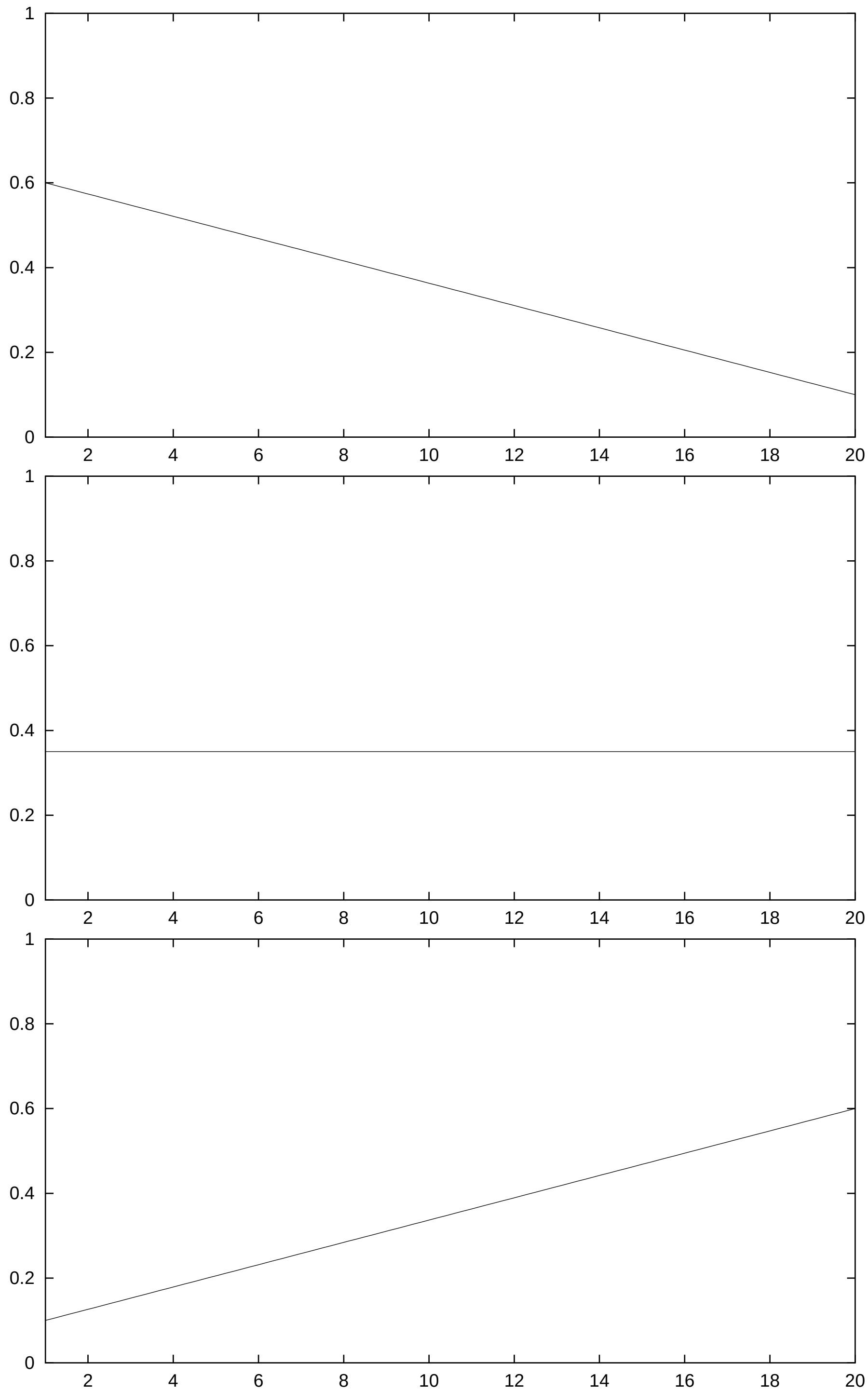

Figure 4.2: From the top, gun, laser, and missile weapons effectiveness

ship has remaining weapons systems, the fight is a draw. If two ships have not fought to a draw or victory in 100 combat turns, their combat is a draw by fiat. (Sometimes ships appear that take forever to resolve a fight, slowing down evolution without substantial benefit. This happens, for example, when the ship's preferred range is badly mismatched with its weapons mix.)

One point that may need emphasis: the genes of the creatures are separate from the ships built from them. When a ship loses systems, its gene remains intact. When a ship wins, its *gene*, not itself, is used in reproduction.

Now that we have the gene, developmental biology, and combat described, we can define *gladiatorial tournament selection*. In this model of evolution, instead of generations there are mating events that asynchronously update the population. This is a form of steady state genetic algorithm. A mating event is performed by conducting combat on randomly selected pairs of distinct ships, returning pairs that draw to the population untouched, until two winners and two losers are obtained. The losers are deleted from the population, and the winners reproduce as in the algorithms in previous chapters. The two children they produce replace the losers.

The fact that there is a group of 4, in which the two best mate and produce children that replace the two worst, make this sort of selection reminiscent of tournament selection, but there are some important differences. The measure of fitness is extremely local: all you really know is that a ship was good enough to beat one other randomly selected ship in one particular combat. The measure of fitness is not a heuristic estimation of quality, but entirely objective within the bounds of luck. All that Sunburn ships do is fight other Sunburn ships, and we are testing them on exactly this task. Since ships are tested against other ships, Sunburn is a clearly a coevolutionary genetic algorithm.

The gladiatorial tournament selection model of evolution can be used whenever two creatures are playing some sort of game against one another. Since it grants fitness for beating one other creature, it is a model of evolution that rewards skill in pairwise interactions, unaffected by other pairwise actions in the population. One might expect that it would be very bad for the emergence of cooperation in a group; we shall test this expectation in Chapter 6 when we study the iterated prisoner's dilemma.

Problems

Problem 4.1 *With gladiatorial tournament selection, a creature may survive indefinitely by accident by simply not being chosen for combat. Compute the probability that a creature remains untested after k mating events in a population of $n = 2m$ creatures. Compute also the expected waiting time until a test.*

Problem 4.2 *Reread Problem 4.1. Suppose that we want to know the time until a creature undergoing evolution with gladiatorial tournament selection breeds or dies. Compute the* *expected time to death or reproduction in terms of the given probability p that any given encounter is a draw.*

Problem 4.3 *The gladiatorial tournament selection described in Section 4.1 uses groups of size 4. Give a reasonable description for how to do gladiatorial tournament selection with larger size groups. Avoid situations in which some of the creatures are not at risk.*

Problem 4.4 *Gladiatorial tournament selection, as portrayed in this section, is a steady state evolutionary algorithm. Give a selection algorithm for a generational version in which every ship engages in combat each generation. Relative to your algorithm, identify at least two details that could have been done in a different way and justify your choice.*

Problem 4.5 Essay. *One of the advantages of tournament selection in general is its ability to preserve population diversity. If we define a generation to be m mating events where there are $n = 2m$ creatures in the population, is gladiatorial tournament selection better or worse at preserving diversity than single tournament selection? Justify your answer logically or experimentally. Be sure to fix a problem domain in which to make your judgment as to what is better. The general answer is likely to be "it depends," which is both correct and uninformative. Possible problem domains are the various example targets of evolutionary computation in the earlier chapters of the book.*

Problem 4.6 Short Essay. *Is gladiatorial tournament selection elitist when it is used in Sunburn? If the answer was "yes" or "no" this would not be an essay question.*

Problem 4.7 *Come up with one or more techniques for converting an optimization that uses tournament selection to one that uses gladiatorial tournament selection. For each such scheme, answer the following questions. Is the scheme elitist? If there are $n = 2m$ creatures in the population, what is the expected number of children a creature will have in a mating event as a function of its rank in the population?*

Problem 4.8 Essay. *In gladiatorial tournament selection one strategy that avoids death is to fight to a draw every time. While this prevents you from having children, it would seem to be a survival strategy. Under what sorts of circumstances is it an effective survival strategy?*

Problem 4.9 *Suppose that we wish to redesign the Sunburn genetic algorithm to evolve ships that beat a fixed, existing design. Give an outline of such a genetic algorithm. Explain the degree to which the algorithm is evolutionary or coevolutionary.*

4.2 Implementing Sunburn

Now that Sunburn is defined, we can play with it. What is the effect of changing the weapons effectiveness curves? How does where you set the initial range affect the outcome? Which kinds of weapons are most effective? Which model of evolution works best with Sunburn?

Experiment 4.1 *Write or obtain software for a Sunburn evolutionary algorithm using gladiatorial tournament selection on a population of 200 ship designs. Use the weapons effectiveness curves from Figure 4.2 which are given by the following formulae. At a range r the probability of hitting for guns, lasers, and missiles is,*

$$P_G(r) = 0.6 - (r-1)/38,$$

$$P_L(r) = 0.35, \textit{ and}$$

$$P_M(r) = 0.1 + (r-1)/38.$$

Use two point crossover where the preferred range is treated as the last character in the string making up the ship design. Use single point mutation, where each ship's system loci are replaced with new, uniformly distributed systems. The preferred range is incremented or decremented by 1 when it is selected for mutation. Assume the starting range is always 20.

In place of generations, do 100 mating events. After each 100 mating events you should report the fraction of the population's genes devoted to each type of system, the fraction of ships that have a drive in the last position (something the rules favor), and the ratio of shields (genetic loci that are "S") to weapons. In addition, report the mean and standard deviation of the preferred ranges and the fraction of combats that are draws. Run each population for 100 "generations", i.e., 10,000 mating events, and run 20 populations with distinct initial populations for comparison.

If you have implemented Experiment 4.1 correctly, it should often converge to a design we call "the starbase" fairly quickly. A "starbase" has lots of missiles, few other weapons, a respectable number of shields, and a single drive on the end of the ship. Preferred ranges will tend to be large, but are essentially irrelevant because of the speed with which combat is resolved. Let us now make a foray into exploring other weapons effectiveness curves.

Experiment 4.2 *Modify the software from Experiment 4.1 to use the following weapons effectiveness curves.*

$$P_G(r) = \frac{0.7}{(r-3)^2+1}$$

$$P_L(r) = \frac{0.6}{(r-6)^2+1}$$

$$P_M(r) = \frac{0.5}{(r-9)^2+1}$$

Gather the same data as before. These weapons effectiveness curves should provide some small incentive for ships to have drive systems other than because of the technicality that a drive system is required to win. Do they? Why or why not? In addition to answering those questions report the various statistics mentioned in Experiment 4.1. A graph of these weapons effectiveness curves is shown in Figure 4.3.

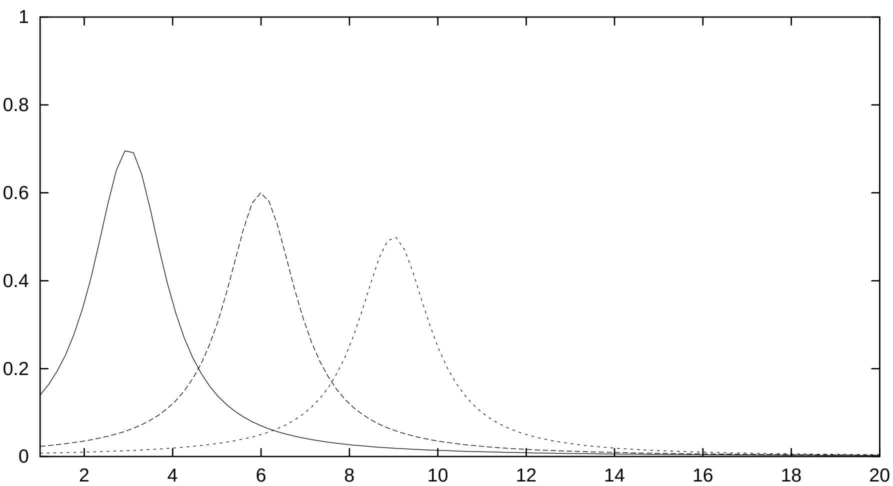

Figure 4.3: Modified weapons effectiveness curves

Next, we will experimentally explore the effect of varying the initial range. Intuitively, starting at long range favors starbases which can launch barrages of highly effective missles. A very short starting range should create a sort of opposite of a starbase, bristling with guns, but otherwise similar to a starbase. We will test the effects of intermediate starting ranges where it is not clear which weapons are best.

Experiment 4.3 *Take either (consult your instructor) of the two sets of weapons effectiveness curves we've examined so far and change the range to be normally distributed with a mean of 10 and a standard deviation of 2. You will need to round to the nearest integer and throw out values larger than 20 or smaller than 1. Report the same statistics as in Experiment 4.1. Does changing the starting range have an effect on evolution? (If you've forgotten the formula to generate normally distributed random numbers, refer to Equation 3.1.)*

The dynamics of selection in Sunburn are strongly controlled by the weapons effectiveness curves. In the next experiment, we have a more than usually open-ended setup in which you are sent off into the wilderness to find interesting weapons effectiveness curves.

Experiment 4.4 *If evolution is* contingent *the fractions of various types of systems at the end of the evolution should vary non-trivially from run to run. Using the software from Experiment 4.1, with appropriate modifications, try to find a set of weapons effectiveness curves for which evolution is more strongly contingent. State in your report how you will document contingency and state the degree to which you found contingency. Did initial findings cause you to revise your method of documenting contingency?*

One issue in the Sunburn system is the question of what makes evolution favor a given type of weapon. The *initial effectiveness* of a weapon is its expected probability of hitting at the initial range. The *total effectiveness* of a weapon is the area under its weapons effectiveness curve. The *maximal effectiveness coefficient* of a weapon is the fraction of the possible ranges where the weapon is more likely to hit than any other. The *range weighted effectiveness* is the sum over possible ranges r of the weapons effectiveness curve multiplied by a range weighting function $\omega(r)$. All of these measures are related to effectiveness but it isn't very clear which one is closest to being the thing optimized for by evolution. When reason does not speak to a question, experimentation may help.

Experiment 4.5 *Let* $\omega(r) = \frac{1}{(r-s)^2+1}$ *where s is the mean starting range for ships. Give and logically justify a hypothesis about which of initial effectiveness, total effectiveness, maximal effectiveness coefficient, or range weighted effectiveness the Sunburn genetic algorithm is in fact optimizing. Then design and perform an experiment to test that hypothesis for a variety of weapons effectiveness curves and starting ranges. Advanced students may also wish to test other possible choices of* $\omega(r)$.

One of the themes that appears over and over in this book is that the model of evolution can have a substantial effect. We will now experimentally test the effect of changing the model of evolution on Sunburn.

Experiment 4.6 *If you have not done Problem 4.4 yet, do so. Now modify the code from Experiment 4.1 to use the generational version of gladiatorial tournament selection. Perform the experiment again, keeping the number of mating events roughly constant (within 2x). In your write up, discuss the changes in the dynamics of evolution and the difference in final designs, if any.*

There are a *lot* of other experiments we could perform. Sunburn has been a rich source of interesting student projects over the years (see Section 4.4). We will cut off the parade of experiments with the basic Sunburn model here and look at less than basic models in the next section.

Problems

Problem 4.10 *There are several ways to detect that a combat is over without fighting to the bitter end. Give at least one set of circumstances where a combat must end in a draw and one where it is clear which ship will win even though both ships still have working systems.*

Problem 4.11 *Suppose we have a simplification of the Sunburn model in which there are only missiles and shields. Ships don't move and the winner is the last ship with any systems. If the probability that a missile hit will destroy a system is* p*, then what is the optimal number of shields in a gene with 20 systems slots and a shield multiplication factor of 3? Attack this problem by thinking of the ships as pairs of numbers that add to 20 and saying one pair dominates another if there is a probability of more than 0.5 that the corresponding ship will win a fight. This gives you a domination relationship on these pairs of numbers which you can compute by hand or machine. If there are ships not dominated by any other they are the optimal designs, otherwise there is no optimum. If the problem seems too hard, try working with a smaller number of system slots until you get a feel for the problem.*

Problem 4.12 *In a given turn, a ship in the Sunburn genetic algorithm may fire several weapons which all have the same probability* p *of hitting. Each weapon needs its own random number. Each such event is a* Bernoulli trial *as defined in Appendix A. If you have several, say more than 3, independent Bernoulli trials they are very close to being normally distributed with mean* np *and standard deviation* $\sqrt{np(1-p)}$*. This means we could substitute a single normal random variable for a large number of uniform random variables. In order to get a normal random variable with mean* μ *and standard deviation* σ*, denoted* $G(\mu, \sigma)$*, you set*

$$G(\mu, \sigma) = \sigma \cdot G(0, 1) + \mu$$

where the formula for $G(0, 1)$ *is given after Equation 3.5.*

Assuming we wish to replace any set of 4 or more weapon resolutions of weapons of the same type with a normal random variable, give pseudocode for doing so for missiles, lasers, and guns. Truncate the normal random variable when it produces unreasonable values. Advanced students may wish to experimentally measure the amount of time saved by this technique.

Problem 4.13 *Please compute the total effectiveness, maximal effectiveness coefficient, and range weighted effectiveness of missiles, lasers, and guns for the weapons effectiveness curves in experiments 4.1 and 4.2. Assume that* $\omega(r) = \frac{1}{(r-20)^2+1}$.

Problem 4.14 *Come up with a better algorithm than the one given in the text for figuring out the change in position of Sunburn ships during the movement part of a turn. You should produce pseudocode that accepts current positions, desired ranges, and number of drives for two ships and returns their new positions.*

4.3 Discussion and Generalizations

Why does the developmental biology of Sunburn have a shield multiplication factor? Well, shields do nothing except soak up damage. If one shield locus in the gene corresponded to one shield in the finished ship, then evolution would quickly replace the shields with a system that did something. This system, whatever it was, could still absorb one hit and the ship would get some additional use out of it. If a shield gene can soak up more than one hit, it has some nontrivial usefulness compared to other systems, and there is reason for a gene to contain some nontrivial number of shield loci.

Experiment 4.7 *Using the software from Experiment 4.1, with appropriate modifications, repeat the experiment with shield multiplication factors of 2 and 4. Compare the results to the original experiment with a shield multiplication of 3.*

Drives, which are far less useful in the basic Sunburn model than the designer thought they would be, are used to move ships to where their weapons are more effective. A large part of the reason drives, other than the one required for victory, are not useful is that ships quickly evolve to be maximally effective at the mean initial range. Experiment 4.2 tries to prevent this by making all weapons woefully ineffective at the starting range. If you are interested, you could try a more extreme tactic by making weapons completely useless at the initial range.

The three sorts of weapons in the initial Sunburn work done by John Walker were thought of as effective at short range (guns), effective at long range (missiles), and somewhat effective at all ranges (lasers) which is reflected in the weapons effectiveness curves shown in Figure 4.2. Changing the weapons effectiveness curves can radically affect the outcome of evolution. Also somewhat surprising is the apparent unimodality of the coevolving space of designs for Sunburn for almost all the weapons effectiveness curves tried thus far. For many choices of such curves the evolution of the resulting populations seems to close in on a single optimal design with a few ineffective forays into dead ends. A few choices of weapons effectiveness curves do yield contingent evolution, but these sets of curves seem rare.

It is obvious that one could increase or decrease the number of types of weapons in Sunburn to get different systems. A model with a single type of weapon is easier to analyze theoretically as we saw in Problem 4.11. In the remainder of this section we will give other possible modifications of the Sunburn model that make good term projects.

Limited Shots

One completely unrealistic feature of the Sunburn model is the unlimited use of the ship's weapons. The ships fire as many shots as they like, one per turn per weapons system. Removing this implausible feature leads to several different variations. First one might try simply imposing a maximum on the total uses of each weapon type, perhaps with less effective weapon types having more shots. Choose these limits carefully. If they are too high, they do not change the behavior of the model at all. If they are too low, they cause a high number of stalemates.

It is plausible that limiting weapon use would encourage the creatures to learn to move to effective ranges as fast as possible before those weapons were used up. The need to use weapons effectively so as not to waste shots also suggests another possible addition to the Sunburn model: a way of deciding when to fire. The technology for such a decision maker could be quite simple, e.g., a real number for each weapon type that serves as a probability threshold. The creature only fires a given type of weapons system when the range is such that the probability of hitting equals or exceeds the threshold. Another natural choice for a decision to fire device is a small neural net. The net might take as inputs range, remaining charges, distance from the weapons system to the front of the ship, hit probability at current range, or number of shields remaining. Its output would be a fire/don't fire decision. Even a very small neural network (see Section 11.1) with one to three neurons could make fairly complex decisions. The ship should have three neural nets, one for each weapon type, rather than one net per weapons system. Ships without a given type of weapon might have vestigial neural control systems for that weapon type. The connection weights for these neural nets would need to be incorporated into the data structure holding a spaceship design.

Experiment 4.8 *Suppose we are doing a Sunburn model with a single type of weapon and that we have augmented the genetics with the probability threshold described above. Running several populations with starting range 20 and weapons effectiveness curve*

$$f_w(r) = \frac{0.7}{(r-10)^2+1},$$

ascertain if there is a single optimum weapons threshold or if the weapons threshold is contingent on the initial conditions. Describe carefully the design of trials you use to settle this question. Does it matter how you incorporate the threshold into the data structure used to hold the spaceship design?

Another control technology that would be interesting but much more difficult to implement is genetic programs (see Section 1.3). As with neural control structures, there should be one GP parse tree for each weapon type which takes integer parameters and outputs a fire/no fire decision. Since the constant function "fire" is a tolerable program for some parts of the parameter space, there is room for evolution to take an extremely simple control strategy and improve it. This is typically an evolution-friendly situation where an evolutionary algorithm can shine.

For each of the three possible control technologies we mention above and any other that you think up on your own, it may be best to start with a single type of weapon until the control strategy evolving software is working properly. Only after this software is working in a predictable fashion should you diversify the weapons mix. Keep in mind that the "always fire" weapons control of the basic Sunburn model, when using one weapon type, is an invitation for evolution to solve the optimization problem given in Problem 4.11. With the control systems in place, however, nontrivial evolutionary behavior is possible even with a single weapons type.

Another variation that could make the Sunburn model more plausible is to add a type of ships system called a magazine which contains some fixed number of missiles, lasers charges, and shells for guns. This way the evolutionary algorithm would have to figure out how many magazines there are and where they should be placed. Taking this variation farther, there could be three types of magazines: one for missiles, one for laser charges, and one for shells. Other variations might include only being able to use magazines if they are adjacent to appropriate weapons types or destroying systems adjacent to magazines when the magazine is hit.

Sensible Movement

The movement in the original Sunburn model is as simple as possible and could easily be made more realistic and difficult. One could change the drives from devices that churn out distance to devices that churn out acceleration. This would probably require placing the ships on a real number line. Acceleration requires three variables, position (s), velocity (v), and acceleration (a). A simple algorithm for acceleration is as follows.

```
For each drive available do
Begin
   If we are too close a := a-1 else a := a+1
end;
v := v+a;
s := s+v;
```

This model of acceleration is very primitive. When farther away than its preferred range, the ship simply accelerates toward the other ship; otherwise, it accelerates away. This will result in very little time spent at optimum range and, interestingly, with the ships having their highest velocity when they are near their desired range. This doesn't have to be a bad thing. Our semantic interpretation of the number we call the ships "preferred range" is the range the ship wants to be at during combat. With the more complex movement system in place, evolution may well find another use for that number, in essence changing the "meaning" of the number. Instead of using this number as a range to maintain, it will become the useful mark for starting to decelerate. It is not unintuitive that this range be chosen so that the ship is slowing down and turning around at a point where its weapons are most effective.

From our knowledge of basic physics, we can guess that a much larger space will be needed for ships with drives that generate acceleration than for ones that just generate position changes. Once you have chosen your larger board size, you may wish to make moving beyond maximum range a condition that leads to stalemate.

As with the weapons systems, it may also be desirable to have a neural net or genetic program that decides if it wants to use the ship's drives in a given turn. The inputs to this net or GP could include current range, number of ships systems left, number of drives left, number of weapons left, and current velocity. This opens up the possibility that a ship will try to run away and achieve stalemate when victory has become impossible, an action that will require some revision to the methods of early detection of stalemates. The more sophisticated drive controls could be used in either the position or acceleration generating environments. Given that a severely damaged ship would want to flee, one also might simply add the following rule to the basic Sunburn model: when a ship is out of weapons systems it will use all remaining drives to increase range. The thought here is that the ship is attempting to achieve stalemate.

Finally, if you implement a more complex model of movement it might be sensible to factor the ship's relative velocity into the weapons effectiveness curves. A faster moving ship might be harder to hit (recall that velocity is a relative *not* an absolute quantity). Some thought should be given to determining how speed affects weapons effectiveness. It may be that it affects different weapons to different degrees and it is almost certainly not linear.

Variations in Taking Damage

It is implausible that a ships systems would be destroyed *in order*. The position of systems would provide a strong bias for the order of destruction but not utterly control it. One simple method of dealing with this is to place a distribution on the systems that favors the front. Pick some probability p and then, each time a ship is hit, do the following. With probability p the first system remaining is destroyed; with probability $1-p$ you skip the first system and go on to the rest of the ship. If $p = 1/2$ then the probability of systems being destroyed is, starting from the front, $1/2$, $1/4$, $1/8$, ...

Experiment 4.9 *Modify the software from Experiment 4.1 to use the probabilistic hit evaluation described above. For $p = 0.75$ and $p = 0.5$, rerun the experiment and report on the differences that this type of hit location assessment causes. In particular, does the number of drives go up and does the ratio of shields to weapons change?*

Another implausible feature of damage assessment in Sunburn is that the same generic shield can stop three very different sorts of weapons. This implausibility can be remedied in a fashion that will also make the evolution more complex and hopefully more interesting. Instead of a single type of shields, there will be one type of shield per type of weapons system. In the original Sunburn model this would mean a missile shield, a laser shield, and a gun shield. These three types of shields would be represented in the genome by distinct letters. The developmental biology is modified as follows: take the pattern of shield genes in the order they appear in the gene; place this pattern of shields on the front of the ship repeated a number of times equal to the shield multiplication factor.

Example 4.1 *If we had a Sunburn simulation with multiple shield types, S_M, S_L, and S_G, and the following gene,*

$$GLMDS_MS_MMGLS_LDS_GMLGMLS_MDS_L\ 16$$

then the resulting ship would look like

$$S_MS_MS_LS_GS_MS_LS_MS_MS_LS_GS_MS_LS_MS_MS_LS_GS_MS_LGLMDMGLDMLGMLD_{16}.$$

Here is one possible way to modify the combat rules in order to accommodate multiple shield types. In combat, a shield is transparent to the type of weapons it is not intended for and is still destroyed by them. Suppose we has a ship with 3 missile shields and then a laser shield as its front 4 systems. If a laser hit the ship, then those 3 missile shields would be destroyed without doing anything and the laser shield would be destroyed, stopping the laser. Notice that having different types of shields makes the order in which various attacks hit important. Assume that the the enemy's hits are generated from the front to the back of his ship. If his forward missile launcher and aft laser turret hit, the missile hit is processed first. This gives the ordering of weapons systems a new importance.

Experiment 4.10 *Modify the software from Experiment 4.1 to have multiple shield types as described above. What differences do the multiple shield types make?*

There are many, many other modifications one could make to the Sunburn model, but we leave these to your inventiveness.

Problems

Problem 4.15 *There is at least one obvious and stupid method of choosing weapons effectiveness curves that will force the fraction of the gene devoted to each type of weapons system to be a contingent feature of evolution. Find an example of such a method.*

Problem 4.16 *Give a choice of weapons effectiveness curves and a fixed starting range for a Sunburn model that will coerce the use of drives. Make your example substantially different from those discussed in the text.*

Problem 4.17 *Using the effectiveness curves given in Figure 4.2, design 3 neural nets with at least one neuron and no more than 6 that are used to decide if the ship should fire guns, missiles, or lasers. Assume the ship has 8 missiles, 20 laser charges, and 20 shells for its* *guns (any weapons system may fire remaining ordinance of an appropriate type) and that the starting range is 20. The nets may use the range, effectiveness curves, position of the weapons system in the ship, and remaining ordinance as inputs. Write a few paragraphs describing how the systems are supposed to work and work together.*

Problem 4.18 *For p equal to each of 0.9, 0.75, 0.5, and 0.3, graph on the same axes the probability of systems being destroyed starting from the front of the ship using the probabilistic hit location system described in Section 4.3 (Variations in Taking Damage).*

Problem 4.19 *In example 4.1, we repeat the pattern of shields in the gene 3 times. Why is this better or worse than taking the pattern once and triplicating each shield as we go? Give an example.*

Problem 4.20 Essay. *Describe a Sunburn variation in which there are two or more weapon types and for which weapons, shields, and drives all draw off a common reserve of energy. Make predictions about the behavior of your system. Advanced students should test these predictions experimentally.*

4.4 Other Ways of Getting Burned

One objection that has been raised to the Sunburn model as an example of coevolving strings is that it is violent. The author, being a fairly typical recovering video and war game addict, simply did not imagine that abstract warfare between character strings was offensive. Once made aware that Sunburn was sufficiently violent to offend some portion of the public, the author put it as a challenge to his students to take the basic notion of Sunburn and find a nonviolent fitness function that could be used to experiment with a similar set of issues. This section is inspired by one of the better attempts to meet the challenge, invented by Mark Joenks as a term project for the class.

The essence of Sunburn is to have strings that compete in some fashion and whose genes spell out how that competition is approached by the entities derived from those genes. In this section, we will develop such a model for individuals participating in political campaigns. It is for you to decide if this is more or less offensive than warfare.

In Sunburn, the character strings had five letters: drive, gun, laser, missle, and shield. These character strings were developed into abstract models of fighter-craft. For our model of political campaigns we need a new alphabet as follows: **A**dopting popular programs, **B**ribing, **D**oing what your opponent did in the last move of the campaign, **F**undraising, **L**aying low, **N**egative campaigning, **P**andering, and having a **S**candal. The artificial agents in this simulation of politics will be called **vi**rtual **p**oliticians or *VIP*s.

Of prime importance in evaluating a character string representing a VIP is our model of the behavior of the electorate. Out fitness evaluation, still used for gladiatorial tournament

Credentials	C	the candidates standing with his single issue voters
Credibility	R	the candidates perceived ability to serve competently
Name Recognition	N	related to being recognized by a voter
Scandal Factor	S	the degree to which the candidate is tainted by scandal
Finances	B	powers everything else

Table 4.1: State variables for VIPs

selection, will run through the campaign season. At regular intervals, the genes of each of our virtual politicians will be expressed, one location at a time, in order. A series of variables will change value during the campaign, depending on the actions of the competitors. At the end the electorate will vote, probabilistically, deciding the contest. The state variables stored for each candidate are given in Table 4.1.

The way we initialize the candidate's state variables describe how the candidates start the campaign season. We update the variables according to the following rules, as we scan down the candidate's gene strings.

Rule 1. Credentials, credibility, name recognition, and scandal factor all undergo exponential decay. At the beginning of each period of the campaign, they are multiplied by r_C, r_R, r_N, and r_S, all smaller than 1. Since the voters have short memories for anything complicated, we insist that

$$r_N > r_R > r_C > r_S.$$

Rule 2. Finances grow exponentially. At the beginning of each period of the campaign, a candidate's money is multiplied by r_F, bigger than 1. This represents fund-raising by the candidate's campaign organization.

Rule 3. Adopting a popular program adds 2 to a candidate's credibility and subtracts 1 from her credentials. If she has at least half as much money as her opponent, it adds 2 to her credibility and subtracts 1 from her credentials. Otherwise, it adds 2 to her credibility and subtracts 1 from her credentials (she swiped the idea).

Rule 4. Bribing either subtracts 5 from a candidate's finances or cuts them in half if his total finances are less than 5. Bribing adds 5 to his credentials, 2 to his scandal factor, and 1 to his name recognition.

Rule 5. Doing what a candidate's opponent did last time is just what it sounds like. On the first action, this action counts as laying low.

Rule 6. Fund-raising adds 3 to a candidate's finances and 1 to her name recognition. It represents a special, personal effort at fund-raising by the candidate.

Rule 7. Laying low has no effect on the state variables.

Rule 8. Negative campaigning subtracts 1 from a candidate's credibility and credentials and adds 1 to the other candidate's credentials. If he has at least half as much money as his opponent, then this goes his way. Otherwise, it goes the other candidate's way.

Rule 9. Pandering adds 5 to a candidate's credentials, 1 to her name recognition, and subtracts 1 from her credibility.

Rule 10. Scandal adds 4 to a candidate's name recognition and subtracts 1 from his credentials and credibility.

Once we have used the rules to translate the VIP's genes into the final version of the state variables, we have an election. In the election, we have 25 special interest voters aligned with each candidate and 50 unaligned voters. Each voter may choose to vote for a candidate or refuse to vote at all. The special interest candidates will vote for their man or not vote. For each voter, check the following probabilities to tally the vote.

A special interest voter will vote for his candidate with probability:

$$P_{special} = \frac{e^{C-S}}{2 + e^{C-S}} \tag{4.1}$$

An unaligned voter will chose a candidate to consider first in proportion to name recognition. He will vote for the first candidate with probability:

$$P_{unaligned} = \frac{e^{R-S}}{3 + e^{R-S}} \tag{4.2}$$

If he does not vote for the first candidate, then he will consider the second candidate using the same distribution. If the unaligned voter still has not voted, then he will repeat this procedure two more times. If, at the end of 3 cycles of consideration, he has still not picked a candidate, he will decline to vote. The election (and the gladiatorial tournament selection) are decided by the majority of voters picking a candidate. If no one votes then the election is a draw.

Experiment 4.11 *Using the procedure outlined in this section, create an evolutionary algorithm for VIPs using gladiatorial tournament selection on a population of 200 VIPs. Use two point crossover on a string of 20 actions with two point mutation. Set the constants as follows:* $r_N = 0.95$, $r_R = 0.9$, $r_C = 0.8$, $r_S = 0.6$, *and* $r_F = 1.2$. *Use uniform initial conditions for the VIPs with the state variables all set to 0, except finances which is set to 4. Perform 100 runs lasting for 20,000 mating events each. Document the strategies that arise. Track average voter turnout and total finances for each run.*

There are an enormous number of variations possible on the VIP evolutionary algorithm. If you find one that works especially well, please send the author a note.

Problems

Problem 4.21 Essay. *Compare and contrast the Sunburn and VIP simulators as evolving systems.*

Problem 4.22 *The choices of constants in this section were pretty arbitrary. Explain the thinking that you imagine would lead to the choices for the four decay constants in Experiment 4.11.*

Problem 4.23 *Explain and critique the rules for the VIP simulator.*

Problem 4.24 *In terms of the model, and referring to the experiment if you have performed it, explain how scandals might help. At what point during the campaign might they be advantageous?*

Problem 4.25 Essay. *The VIPs described in this section have a pre-programmed set of actions. Would we obtain more interesting results if they could make decisions based on the state variables? Outline how to create a data structure that could map the state variables onto actions.*

Problem 4.26 *Cast you mind back to the most recent election in your home state or country. Write out and justify a VIP gene for the two leading candidates.*

Problem 4.27 *The VIP simulator described in this section is clearly for a two-sided contest. Outline how to modify the simulator to run a simulation of a primary election.*

Problem 4.28 *We have the electorate divided 25/50/25 in Experiment 4.11. Outline the changes required to simulate a 10/50/40 population in which one special interest group outnumbers another, but both are politically active. Refer to real world political situations to justify your design.*

Problem 4.29 *Analyze Equations 4.1 and 4.2. What are the advantages and disadvantages of those functions? Are they reasonable choices given their place in the overall simulation? Hint: graph* $f(x) = \frac{e^x}{c+e^x}$ *for* $c = 1, 2, 3$.

Problem 4.30 *Should the outcome of some actions depend on what the other candidate did during the same campaign period? Which ones, why, and how would you implement the dependence?*