Chapter 1

Introduction

The field of evolutionary computation is itself an evolving community of people, ideas, and applications. Although one can trace its genealogical roots as far back as the 1930s, it was the emergence of relatively inexpensive digital computing technology in the 1960s that served as an important catalyst for the field. The availability of this technology made it possible to use computer simulation as a tool for analyzing systems much more complex than those analyzable mathematically.

Among the more prominent groups of scientists and engineers who saw this emerging capability as a means to better understand complex evolutionary systems were evolutionary biologists interested in developing and testing better models of natural evolutionary systems, and computer scientists and engineers wanting to harness the power of evolution to build useful new artifacts. More recently there has been considerable interest in evolutionary systems among a third group: artificial-life researchers wanting to design and experiment with new and interesting artificial evolutionary worlds.

Although these groups share a common interest in understanding evolutionary processes better, the particular choices of what to model, what to measure, and how to evaluate the systems built varies widely as a function of their ultimate goals. It is beyond the scope of this book to provide a comprehensive and cohesive view of all such activities. Rather, as the title suggests, the organization, selection, and focus of the material of this book is intended to provide a clear picture of the state of the art of evolutionary computation: the use of evolutionary systems as computational processes for solving complex problems.

Even this less ambitious task is still quite formidable for several reasons, not the least of which is the explosive growth in activity in the field of evolutionary computation during the past decade. Any attempt to summarize the field requires fairly difficult choices in selecting the areas to be covered, and is likely be out of date to some degree by the time it is published!

A second source of difficulty is that a comprehensive and integrated view of the field of evolutionary computation has not been attempted before. There are a variety of excellent books and survey papers describing particular subspecies of the field such as genetic algorithms, evolution strategies, and evolutionary programming. However, only in the past few years have people begun to think about such systems as specific instances of a more

general class of evolutionary algorithms. In this sense the book will not be just a summary of existing activity but will require breaking new ground in order to present a more cohesive view of the field.

A final source of difficulty is that, although the goals of other closely related areas (such as evolutionary biology and artificial life are quite different, each community has benefitted significantly from a cross-fertilization of ideas. Hence, it is important to understand to some extent the continuing developments in these closely related fields. Time and space constraints prohibit any serious attempt to do so in this book. Rather, I have adopted the strategy of scattering throughout the book short discussions of related activities in these other fields with pointers into the literature that the interested reader can follow.

For that strategy to be effective, however, we need to spend some time up front considering which ideas about evolution these fields share in common as well as their points of divergence. These issues are the focus of the remainder of this chapter.

1.1 Basic Evolutionary Processes

A good place to start the discussion is to ask what are the basic components of an evolutionary system. The first thing to note is that there are at least two possible interpretations of the term *evolutionary system*. It is frequently used in a very general sense to describe a system that changes incrementally over time, such as the software requirements for a payroll accounting system. The second sense, and the one used throughout this book, is the narrower use of the term in biology, namely, to mean a Darwinian evolutionary system.

In order to proceed, then, we need to be more precise about what constitutes such a system. One way of answering this question is to identify a set of core components such that, if any one of these components were missing, we would be reluctant to describe it as a Darwinian evolutionary system. Although there is by no means a consensus on this issue, there is fairly general agreement that Darwinian evolutionary systems embody:

- one or more populations of individuals competing for limited resources,
- the notion of dynamically changing populations due to the birth and death of individuals,
- a concept of fitness which reflects the ability of an individual to survive and reproduce, and
- a concept of variational inheritance: offspring closely resemble their parents, but are not identical.

Such a characterization leads naturally to the view of an evolutionary system as a process that, given particular initial conditions, follows a trajectory over time through a complex evolutionary state space. One can then study various aspects of these processes such as their convergence properties, their sensitivity to initial conditions, their transient behavior, and so on.

Depending on one's goals and interests, various components of such a system may be fixed or themselves subject to evolutionary pressures. The simplest evolutionary models focus on

the evolution over time of a single fixed-size population of individuals in a fixed environment with fixed mechanisms for reproduction and inheritance. One might be tempted to dismiss such systems as too simple to be of much interest. However, even these simple systems can produce a surprisingly wide range of evolutionary behavior as a result of complex nonlinear interactions between the initial conditions, the particular choices made for the mechanisms of reproduction and inheritance, and the properties of the environmentally induced notion of fitness.

1.2 EV: A Simple Evolutionary System

To make these notions more concrete, it is worth spending a little time describing a simple evolutionary system in enough detail that we can actually simulate it and observe its behavior over time. In order to do so we will be forced to make some rather explicit implementation decisions, the effects of which we will study in more detail in later chapters. For the most part these initial implementations decisions will not be motivated by the properties of a particular natural evolutionary system. Rather, they will be motivated by the rule "Keep it simple, stupid!" which turns out to be a surprisingly useful heuristic for building evolutionary systems that we have some hope of understanding!

The first issue to be faced is how to represent the individuals (organisms) that make up an evolving population. A fairly general technique is to describe an individual as a fixed length vector of L features that are chosen presumably because of their (potential) relevance to estimating an individual's fitness. So, for example, individuals might be characterized by:

$$< \textit{hair color, eye color, skin color, height, weight} >$$

We could loosely think of this vector as specifying the genetic makeup of an individual, i.e., its genotype specified as a chromosome with five genes whose values result in an individual with a particular set of traits. Alternatively, we could consider such vectors as descriptions of the observable physical traits of individuals, i.e., their phenotype. In either case, by additionally specifying the range of values (alleles) such features might take on, one defines a five-dimensional space of all possible genotypes (or phenotypes) that individuals might have in this artificial world.

In addition to specifying the "geno/phenospace", we need to define the "laws of motion" for an evolutionary system. As our first attempt, consider the following pseudo-code:

```
EV:

  Generate an initial population of M individuals.

  Do Forever:

    Select a member of the current population to be a parent.
```

```
    Use the selected parent to produce an offspring that is
    similar to but generally not a precise copy of the parent.

    Select a member of the population to die.

    End Do
```

Although we still need to be more precise about some implementation details, notice what enormous simplifications we already have made relative to biological evolutionary systems. We have blurred the distinction between the genotype and the phenotype of an individual. There is no concept of maturation to adulthood via an environmentally conditioned development process. We have ignored the distinction between male and female and have only asexual reproduction. What we have specified so far is just a simple procedural description of the interacting roles of birth, death, reproduction, inheritance, variation, and selection.

The fact that the population in EV never grows or shrinks in size may seem at first glance to be rather artificial and too restrictive. We will revisit this issue later. For now, we keep things simple and note that such a restriction could be plausibly justified as a simple abstraction of the size limitations imposed on natural populations by competition for limited environmental resources (such as the number of scientists funded by NSF!).

In any case, undaunted, we proceed with the remaining details necessary to implement and run a simulation. More precisely, we elaborate EV as follows:

```
EV:

  Randomly generate the initial population of M individuals
  (using a uniform probability distribution over the entire
  geno/phenospace) and compute the fitness of each individual.

  Do Forever:

    Choose a parent as follows:

      - select a parent randomly using a uniform probability
        distribution over the current population.

    Use the selected parent to produce a single offspring by:

      - making an identical copy of the parent, and then
        probabilistically mutating it to produce the offspring.

    Compute the fitness of the offspring.

    Select a member of the population to die by:
```

```
    - randomly selecting a candidate for deletion from the
      current population using a uniform probability
      distribution; and keeping either the candidate or the
      offspring depending on which one has higher fitness.

End Do
```

There are several things to note about this elaboration of EV. The first is the decision to make the system stochastic by specifying that the choice of various actions is a function of particular probability distributions. This means that the behavior of EV can (and generally will) change from one simulation run to the next simply by changing the values used to initialize the underlying pseudo-random number generators. This can be both a blessing and a curse. It allows us to easily test the robustness and the range of behaviors of the system under a wide variety of conditions. However, it also means that we must take considerable care not to leap to conclusions about the behavior of the system based on one or two simulation runs.

The second thing to note is that we have used the term "fitness" here in a somewhat different way than the traditional biological notion of fitness which is an *ex post facto* measure based on an individual's ability to both survive and produce viable offspring. To be more precise, in EV we are assuming the existence of a mechanism for measuring the "quality" of an individual at birth, and that "quality" is used in EV to influence an individual's *ex post facto* fitness.

A standard approach to defining such measures of quality is to provide a function that defines a "fitness landscape" over the given geno/phenospace. This is sometimes referred to as *objective fitness* since this measurement is based solely on an individual's geno/phenotype and is not affected by other factors such as the current makeup of the population. This confusion in terminology is so deeply ingrained in the literature that it is difficult to avoid. Since the evolutionary computation community is primarily interested in how evolutionary systems generate improvements in the quality of individuals (and not interested so much in *ex post facto* fitness), the compromise that I have adopted for this book is to use the term *fitness* to refer by default to the assessment of quality. Whenever this form of fitness is based on an objective measure of quality, I generally emphasize this fact by using the term *objective fitness*. As we will see in later chapters, the terminology can become even more confusing in situations in which no such objective measure exists.

Finally, note that we still must be a bit more precise about how mutation is to be implemented. In particular, we assume for now that each gene of an individual is equally likely to be mutated, and that on the average only one gene is mutated when producing an offspring. That is, if there are L genes, each gene has an independent probability of 1/L of being selected to undergo a mutation.

If we assume for convenience that the values a feature can take on are real numbers, then a natural way to implement mutation is as a small perturbation of an inherited feature value. Although a normally distributed perturbation with a mean of zero and an appropriately scaled variance is fairly standard in practice, we will keep things simple for now by just using

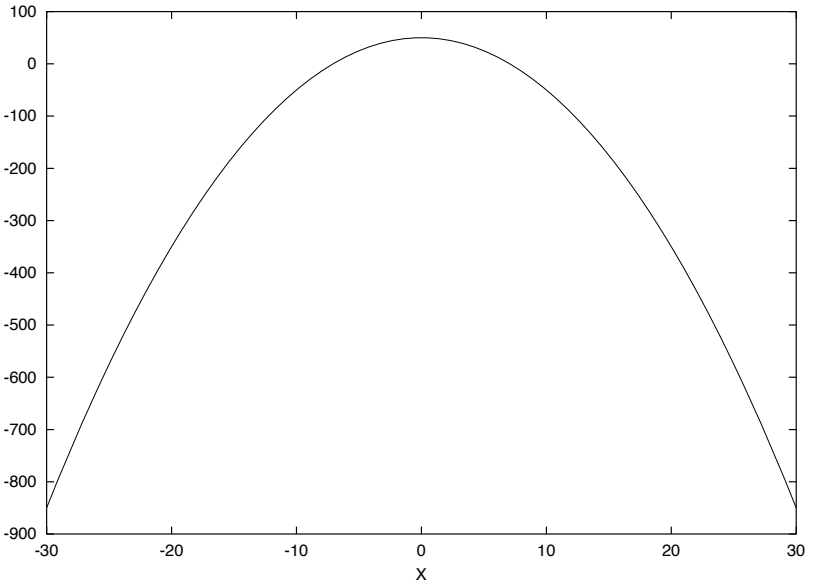

Figure 1.1: L1, a simple 1-D objective fitness landscape.

a fixed size delta to be added or subtracted with equal probability to the value inherited from a parent.

1.3 EV on a Simple Fitness Landscape

We will examine additional EV implementation issues in more detail later in this chapter and, for the interested reader, the source code for EV is described in appendix A.1. For our purposes here we will ignore these details and focus on the behavior of EV under a variety of different situations. We begin by considering a simple one-dimensional geno/phenospace L1 in which individuals consist of a single "trait" expressed as a real number, and their objective fitness is computed via a simple time-invariant function of that real-valued trait. More specifically, an individual is given by $< x >$ and the fitness of individual $< x >$ is defined by the objective fitness function $f(x) = 50 - x^2$, a simple inverted parabola as depicted graphically in figure 1.1.

Running EV on L1 with a mutation delta of 1.0 and a population of size 10, we observe the following initial output:

```
Simulation time limit (# births): 1000
Random number seed (positive integer): 12345
Using an inverted parabolic landscape defined on 1 parameter(s) with
  offset 50.0,  with parameter initialization bounds of:
     1: -100.0  100.0
```

```
  and with hard parameter bounds of:
     1: -Infinity  Infinity
Using a genome with 1 real-valued gene(s).
Using delta mutation with step size 1.0
Population size: 10

Population data after 10 births (generation 1):
 Global fitness:  max = 43.02033, ave = -4599.12864, min = -8586.50977
 Local fitness:   max = 43.02033, ave = -4599.12793, min = -8586.50977
 Indiv  birthdate   fitness      gene values
    1      1        -713.93585    -27.63939
    2      2          43.02033      2.64191
    3      3       -7449.33398     86.59869
    4      4       -6909.38477     83.42293
    5      5       -4387.99414     66.61827
    6      6       -8499.85352    -92.46542
    7      7       -1154.42651    -34.70485
    8      8       -5584.96094    -75.06638
    9      9       -2747.90723    -52.89525
   10     10       -8586.50977    -92.93282
```

At this point in the simulation, an initial population of 10 individuals has been randomly generated from the interval [-100.0,100.0] and their objective fitnesses have been computed. Notice that exactly 1 new individual is produced at each simulated clock tick. Hence each individual has a unique "birth date".

Since EV maintains a constant size population M, we introduce the notion of a "generation" as equivalent to having produced M new individuals, and we print out various things of interest on generational boundaries. Thus, after 10 more births (clock ticks) we see:

```
Population data after 20 births (generation 2):
 Global fitness:  max = 43.02033, ave = -5350.27546, min = -8586.50977
 Local  fitness:   max = 43.02033, ave = -3689.70459, min = -8499.85352
 Indiv  birthdate   fitness      gene values
    1      1        -713.93585    -27.63939
    2      2          43.02033      2.64191
    3     18        -659.65704    -26.63939
    4      4       -6909.38477     83.42293
    5      5       -4387.99414     66.61827
    6      6       -8499.85352    -92.46542
    7      7       -1154.42651    -34.70485
    8      8       -5584.96094    -75.06638
    9     20        -713.93585    -27.63939
   10     15       -8315.92188    -91.46542
```

By looking at the birth dates of the individuals in generation 2 we can see that only 3 of the 10 new individuals produced (birth dates 11–20) were "fit enough" to survive along with 7 members of the previous generation. Global objective fitness statistics have been updated to reflect the characteristics of all individuals produced during an entire simulation run regardless of whether they survived or not. Local objective fitness statistics have been updated to reflect the individuals making up the current population.

Note that, because survival in EV involves having higher objective fitness than one's competitor, the average objective fitness of the population is monotonically nondecreasing over time. If we now let the simulation run for several more generations, we begin to see more clearly the effects of competitive survival:

```
Population data after 60 births (generation 6):
 Global fitness:  max = 49.58796, ave = -2510.92094, min = -8586.50977
 Local fitness:   max = 49.58796, ave =    41.44871, min =    28.45270
 Indiv  birthdate   fitness      gene values
    1     52          47.30414      1.64191
    2     37          49.58796      0.64191
    3     48          36.73652      3.64191
    4     53          36.73652      3.64191
    5     35          43.02033      2.64191
    6     59          47.30414      1.64191
    7     32          47.30414      1.64191
    8     46          49.58796      0.64191
    9     55          28.45270      4.64191
   10     56          28.45270      4.64191
```

Notice that the gene values of all 10 members of the population are now quite similar, but still some distance from a value of 0.0 which achieves a maximum objective fitness of 50.0 on this landscape. Continuing on, we see further movement of the population toward the region of highest fitness:

```
Population data after 120 births (generation 12):
 Global fitness:  max = 49.87177, ave = -1231.82122, min = -8586.50977
 Local fitness:   max = 49.87177, ave =    49.84339, min =    49.58796
 Indiv  birthdate   fitness      gene values
    1     62          49.87177     -0.35809
    2     89          49.87177     -0.35809
    3     91          49.58796      0.64191
    4     65          49.87177     -0.35809
    5    109          49.87177     -0.35809
    6     72          49.87177     -0.35809
    7     96          49.87177     -0.35809
    8    108          49.87177     -0.35809
    9     77          49.87177     -0.35809
   10    100          49.87177     -0.35809
```

By generation 25, the population has become completely homogeneous:

```
Population data after 250 births (generation 25):
 Global fitness:  max = 49.87177, ave = -565.91903, min = -8586.50977
 Local fitness:   max = 49.87177, ave =   49.87177, min =    49.87177
 Indiv  birthdate   fitness      gene values
    1     62          49.87177     -0.35809
    2     89          49.87177     -0.35809
    3    248          49.87177     -0.35809
    4     65          49.87177     -0.35809
    5    109          49.87177     -0.35809
    6     72          49.87177     -0.35809
    7     96          49.87177     -0.35809
    8    108          49.87177     -0.35809
    9     77          49.87177     -0.35809
   10    100          49.87177     -0.35809
```

And, if we let the system run indefinitely from here, we see that EV has, in fact, converged to a stable fixed point:

```
Population data after 1000 births (generation 100):
 Global fitness:  max = 49.87177, ave = -104.82450, min = -8586.50977
 Local fitness:   max = 49.87177, ave =   49.87177, min =    49.87177
 Indiv  birthdate   fitness      gene values
    1     62          49.87177     -0.35809
    2     89          49.87177     -0.35809
    3    248          49.87177     -0.35809
    4     65          49.87177     -0.35809
    5    109          49.87177     -0.35809
    6     72          49.87177     -0.35809
    7     96          49.87177     -0.35809
    8    108          49.87177     -0.35809
    9     77          49.87177     -0.35809
   10    100          49.87177     -0.35809
```

An immediate question that comes to mind is whether EV will converge this way in general. To see that this is the case for simple static fitness landscapes, recall that EV has no upper limit on the lifetime of an individual. Individuals only die when challenged by new individuals with higher objective fitness, and the only way to increase objective fitness is via the mutation operator which introduces small fixed-size perturbations of existing individuals. In the example above we see that from generation 25 on there has been no change in the population. This is because, at this point, every mutation results in a gene value farther away from 0.0 and thus lower in objective fitness.

This raises an important issue that we will revisit many times: the importance of distinguishing between:

- the conceptual geno/phenospace one has in mind to be explored,
- the actual geno/phenospace that is represented and searched by an evolutionary algorithm, and
- the subset of the represented space that is actually reachable via a particular evolutionary algorithm.

In the example above, the conceptual space is the infinite set of all real numbers, the actual space is the finite set of all real numbers representable using the computer's 32-bit floating point representation, and the set of reachable points are those which are can be produced by repeated applications of the mutation operator on members of the randomly generated initial population. In this particular case, each of the 10 members of the initial population can be mutated only by using a mutation delta of 1.0, implying that only a relatively small number of points are reachable during a particular simulation!

To illustrate these ideas, we run EV again changing only the initial random number seed, which results in a different initial population and a different stable fixed point:

```
Simulation time limit (# births): 1000
Random number seed (positive integer): 1234567
Using an inverted parabolic landscape defined on 1 parameter(s) with
  offset 50.0, with parameter initialization bounds of:
     1: -100.0  100.0
   and with hard parameter bounds of:
     1: -Infinity  Infinity
Using a genome with 1 real-valued gene(s).
Using delta mutation with step size 1.0
Population size: 10

Population data after 10 births (generation 1):
 Global fitness:  max = -59.71169, ave = -2588.03008, min = -8167.49316
 Local fitness:   max = -59.71169, ave = -2588.02979, min = -8167.49316
 Indiv  birthdate   fitness      gene values
    1       1       -2595.38550    -51.43331
    2       2        -669.55261     26.82448
    3       3       -2490.53003     50.40367
    4       4         -59.71169    -10.47433
    5       5        -118.75333    -12.99051
    6       6       -1659.58362    -41.34711
    7       7       -1162.48181     34.82071
    8       8       -7870.33447     88.99626
    9       9       -8167.49316     90.65039
   10      10       -1086.47461    -33.71164
```

```
Population data after 100 births (generation 10):
 Global fitness:  max = 43.87767, ave =  -659.01723, min = -8349.79395
 Local fitness:   max = 43.87767, ave =    34.34953, min =    20.03166
 Indiv  birthdate   fitness      gene values
    1     57          20.03166     -5.47433
    2     96          37.92900     -3.47433
    3     93          37.92900     -3.47433
    4     66          29.98033     -4.47433
    5     65          29.98033     -4.47433
    6     84          37.92900     -3.47433
    7     77          37.92900     -3.47433
    8     94          29.98033     -4.47433
    9     78          37.92900     -3.47433
   10    100          43.87767     -2.47433

Population data after 500 births (generation 50):
 Global fitness:  max = 49.77501, ave =   -93.92220, min = -8349.79395
 Local fitness:   max = 49.77501, ave =    49.77500, min =    49.77501
 Indiv  birthdate   fitness      gene values
    1    154          49.77501     -0.47433
    2    158          49.77501     -0.47433
    3    323          49.77501     -0.47433
    4    251          49.77501     -0.47433
    5    306          49.77501     -0.47433
    6    155          49.77501     -0.47433
    7    188          49.77501     -0.47433
    8    151          49.77501     -0.47433
    9    145          49.77501     -0.47433
   10    144          49.77501     -0.47433
```

Note that on this run convergence occurred slightly farther away from 0.0 than on the previous run. Looking at the initial population and recalling that mutation only can change gene values by adding or subtracting 1.0, it is easy to see that this convergence point is the result of a sequence of successful mutations of individual 4. It is also interesting to note that a sequence of successful mutations of individual 6 would have gotten even closer to the boundary. However, for that to happen during the run, each of the individual mutations in the sequence would have to occur and also to survive long enough for the next mutation in the sequence to also occur. Hence we see that there are, in general, reachable points that may have higher fitness than the one to which EV converges, because the probability of putting together the required sequence of mutations in this stochastic competitive environment is too low.

A reasonable question at this point is whether much of what we have seen so far is just an artifact of a mutation operator that is too simple. Certainly this is true to some extent.

Since the genes in EV are real numbers, a more interesting (and frequently used) mutation operator is one that changes a gene value by a small amount that is determined by sampling a Gaussian (normal) distribution $G(0, \sigma)$ with a mean of zero and a standard deviation of σ. In this case the average step size s (i.e., the absolute value of $G(0, \sigma)$) is given by $\sqrt{2/\pi} * \sigma$ (see the exercises at the end of this chapter), allowing for a simple implementation of the Gaussian mutation operator $GM(s) = G(0, s/\sqrt{2/\pi})$.

To see the effects of this, we rerun EV with a Gaussian mutation operator $GM(s)$ using a step size $s = 1.0$, and compare it to the earlier results obtained using a fixed mutation delta of 1.0:

```
Simulation time limit (# births): 1000
Random number seed (positive integer): 12345
Using an inverted parabolic landscape defined on 1 parameter(s) with
  offset 50.0, with parameter initialization bounds of:
     1: -100.0  100.0
  and with hard parameter bounds of:
     1: -Infinity  Infinity
Using a genome with 1 real-valued gene(s).
Using Gaussian mutation with step size 1.0
Population size: 10

Population data after 10 births (generation 1):
 Global fitness:  max = 43.02033, ave = -4599.12864, min = -8586.50977
 Local fitness:   max = 43.02033, ave = -4599.12793, min = -8586.50977
 Indiv  birthdate   fitness      gene values
    1      1        -713.93585    -27.63939
    2      2          43.02033      2.64191
    3      3       -7449.33398     86.59869
    4      4       -6909.38477     83.42293
    5      5       -4387.99414     66.61827
    6      6       -8499.85352    -92.46542
    7      7       -1154.42651    -34.70485
    8      8       -5584.96094    -75.06638
    9      9       -2747.90723    -52.89525
   10     10       -8586.50977    -92.93282

Population data after 100 births (generation 10):
 Global fitness:  max = 49.98715, ave = -1078.09662, min = -8586.50977
 Local fitness:   max = 49.98715, ave =    49.80241, min =    49.50393
 Indiv  birthdate   fitness      gene values
    1     89          49.84153     -0.39809
    2     70          49.85837      0.37634
    3     90          49.50393     -0.70433
    4     75          49.66306     -0.58047
    5     96          49.79710     -0.45044
```

```
    6     74          49.89342      0.32647
    7     22          49.85916      0.37529
    8     48          49.98715      0.11336
    9     83          49.98008      0.14113
   10     76          49.64036     -0.59970

Population data after 500 births (generation 50):
 Global fitness:  max = 49.99962, ave =  -176.97845, min = -8586.50977
 Local fitness:   max = 49.99962, ave =    49.99478, min =    49.98476
 Indiv  birthdate   fitness      gene values
    1    202          49.99303     -0.08348
    2    208          49.99561      0.06628
    3    167          49.99860      0.03741
    4    415          49.99311     -0.08297
    5    455          49.98476      0.12345
    6    275          49.98920      0.10392
    7    398          49.99706     -0.05423
    8    159          49.99914     -0.02935
    9    131          49.99767     -0.04832
   10    427          49.99962     -0.01945
```

Notice how much more diverse the population is at generation 10 than before, and how much closer the population members are to 0.0 in generation 50. Has EV converged to a fixed point as before? Not in this case, since there is always a small chance that mutations still can be generated that will produce individuals closer to 0.0. It is just a question of how long we are willing to wait! So, for example, allowing EV to run for an additional 500 births results in:

```
Population data after 1000 births (generation 100):
 Global fitness:  max = 49.99999, ave = -64.39950, min = -8586.50977
 Local fitness:   max = 49.99999, ave =  49.99944, min =    49.99843
 Indiv  birthdate   fitness      gene values
    1    794          49.99901      0.03148
    2    775          49.99989     -0.01055
    3    788          49.99967      0.01822
    4    766          49.99949      0.02270
    5    955          49.99927      0.02703
    6    660          49.99991      0.00952
    7    762          49.99999      0.00275
    8    159          49.99914     -0.02935
    9    569          49.99843      0.03962
   10    427          49.99962     -0.01945
```

Another obvious question concerns the effect of the mutation step size on EV. Would increasing the probability of taking larger steps speed up the rate of convergence to regions of

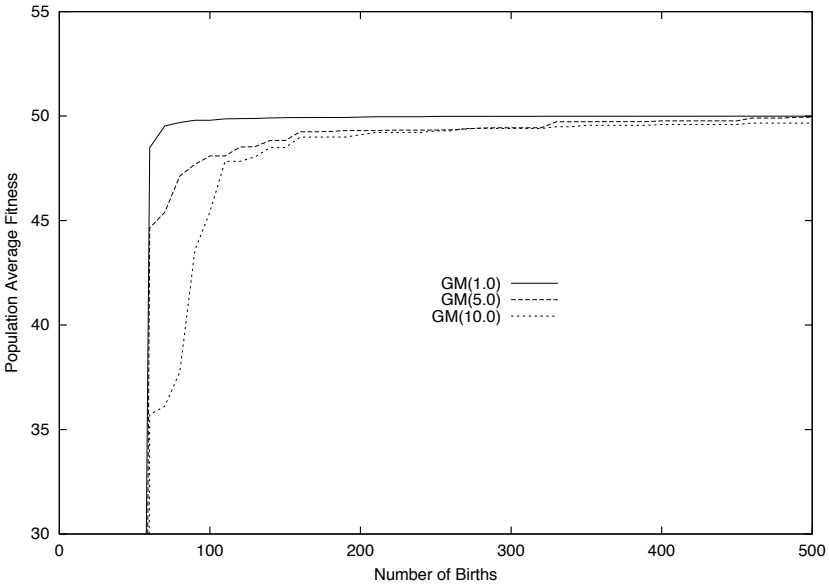

Figure 1.2: Average population fitness in EV using different Gaussian mutation step sizes.

high fitness? One way to answer this question is to focus our attention on more macroscopic properties of evolutionary systems, such as how the average fitness of the population changes over time. EV provides that information for us by printing the "average local fitness" of the population at regular intervals during an evolutionary run. Figure 1.2 plots this information for three runs of EV: one with mutation set to $GM(1.0)$, one with $GM(5.0)$, and one with $GM(10.0)$.

What we observe is somewhat surprising at first: increasing the mutation step size actually slowed the rate of convergence slightly! This is our first hint that even simple evolutionary systems like EV can exhibit surprisingly complex behavior because of the nonlinear interactions between their subcomponents. In EV the average fitness of the population can increase only when existing population members are replaced by new offspring with higher fitness. Hence, one particular event of interest is when a mutation operator produces an offspring whose fitness is greater than its parent.

If we define such an event in EV as a "successful mutation", it should be clear that, for landscapes like L1, the "success rate" of mutation operators with larger step sizes decreases more rapidly than those with smaller step sizes as EV begins to converge to the optimum, resulting in the slower *rates* of convergence that we observe in figure 1.2.

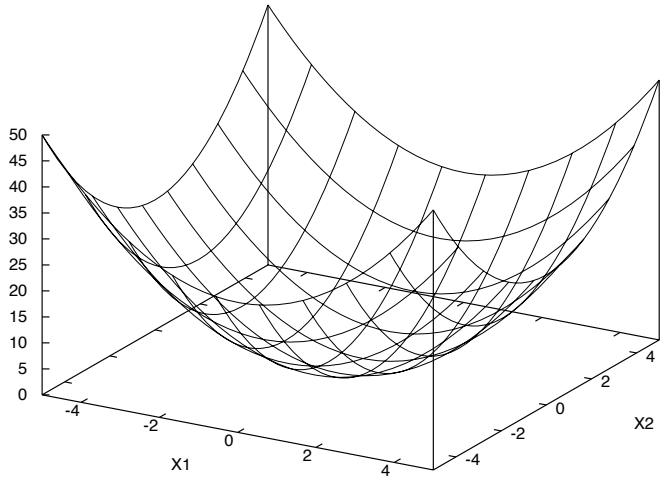

Figure 1.3: L2, a 2-D objective fitness landscape with multiple peaks.

1.4 EV on a More Complex Fitness Landscape

So far we have been studying the behavior of EV on the simple one-dimensional, single-peak fitness landscape L1. In this section, the complexity of the landscape is increased a small amount in order to illustrate more interesting and (possibly) unexpected EV behavior.

Consider a two-dimensional geno/phenospace in which individuals are represented by a feature vector $< x1, x2 >$ in which each feature is a real number in the interval $[-5.0, 5.0]$, and $f2(x1, x2) = x1^2 + x2^2$ is the objective fitness function. That is, this fitness landscape L2 takes the form of a two-dimensional parabola with four peaks of equal height at the four corners of the geno/phenospace as shown in figure 1.3.

What predictions might one make regarding the behavior of EV on this landscape? Here are some possibilities:

1. EV will converge to a homogeneous fixed point as before. However, there are now four peaks of attraction. Which one it converges near will depend on the randomly generated initial conditions.

2. EV will converge to a stable fixed point in which the population is made up of subpopulations (species), each clustered around one of the peaks.

3. EV will not converge as before, but will oscillate indefinitely among the peaks.

4. The symmetry of the fitness landscape induces equal but opposing pressures to increment and decrement both feature values. This results in a dynamic equilibrium in

which the average fitness of the population changes very little from its initial value.

As we will see repeatedly throughout this book, there are enough subtle interactions among the components of even simple evolutionary systems that one's intuition is frequently wrong. In addition, these nonlinear interactions make it quite difficult to analyze evolutionary systems formally and develop strong predictive models. Consequently, much of our understanding is derived from experimentally observing the (stochastic) behavior of an evolutionary system. In the case of EV on this 2-D landscape L2, we observe:

```
EV:
  Simulation time limit (# births): 1000
  Random number seed (positive integer): 12345
  Using a parabolic landscape defined on 2 parameter(s)
   with parameter initialization bounds of:
     1: -5.0  5.0
     2: -5.0  5.0
   and with hard parameter bounds of:
     1: -5.0  5.0
     2: -5.0  5.0
  Using a genome with 2 real-valued gene(s).
  Using gaussian mutation with step size 0.1
  Population size: 10

Population data after 100 births (generation 10):
 Global fitness:  max = 45.09322, ave = 34.38162, min =  1.49947
 Local fitness:   max = 45.09322, ave = 42.91819, min = 38.90914
 Indiv  birthdate   fitness      gene values
    1    100       44.09085      4.77827      4.61074
    2     98       43.89545      4.70851      4.66105
    3     99       45.09322      4.73010      4.76648
    4     61       40.24258      4.36996      4.59848
    5     82       43.03299      4.66963      4.60734
    6     73       43.03299      4.66963      4.60734
    7     89       43.65926      4.66963      4.67481
    8     93       42.74424      4.58463      4.66105
    9     38       38.90914      4.29451      4.52397
   10     95       44.48112      4.82220      4.60734

Population data after 500 births (generation 50):
 Global fitness:  max = 49.93839, ave = 32.42865, min =  1.49947
 Local fitness:   max = 49.93839, ave = 49.92647, min = 49.91454
 Indiv  birthdate   fitness      gene values
    1    457       49.91454      4.99898      4.99246
    2    349       49.91454      4.99898      4.99246
    3    373       49.93839      4.99898      4.99485
    4    201       49.91454      4.99898      4.99246
```

```
    5    472       49.93839      4.99898      4.99485
    6    322       49.91454      4.99898      4.99246
    7    398       49.93839      4.99898      4.99485
    8    478       49.93839      4.99898      4.99485
    9    356       49.91454      4.99898      4.99246
   10    460       49.93839      4.99898      4.99485

Population data after 1000 births (generation 100):
 Global fitness:  max = 49.96095, ave = 30.10984, min =  1.49947
 Local fitness:   max = 49.96095, ave = 49.96095, min = 49.96095
 Indiv  birthdate   fitness      gene values
    1    806       49.96095      4.99898      4.99711
    2    737       49.96095      4.99898      4.99711
    3    584       49.96095      4.99898      4.99711
    4    558       49.96095      4.99898      4.99711
    5    601       49.96095      4.99898      4.99711
    6    642       49.96095      4.99898      4.99711
    7    591       49.96095      4.99898      4.99711
    8    649       49.96095      4.99898      4.99711
    9    632       49.96095      4.99898      4.99711
   10    566       49.96095      4.99898      4.99711
```

On this particular run we see the same behavior pattern as we observed on the simpler 1-D landscape L1: fairly rapid movement of the entire population into a region of high objective fitness. In this particular simulation run, the population ends up near the peak at $< 5.0, 5.0 >$. If we make additional runs using different random number seeds, we see similar convergence behavior to any one of the four peaks with equal likelihood:

```
EV:
  Simulation time limit (# births): 1000
  Random number seed (positive integer): 1234567
  Using a parabolic landscape defined on 2 parameter(s)
   with parameter initialization bounds of:
     1: -5.0  5.0
     2: -5.0  5.0
   and with hard parameter bounds of:
     1: -5.0  5.0
     2: -5.0  5.0
  Using a genome with 2 real-valued gene(s).
  Using gaussian mutation with step size 0.1
  Population size: 10

Population data after 500 births (generation 50):
 Global fitness:  max = 49.80204, ave = 33.45935, min =  4.69584
 Local fitness:   max = 49.80204, ave = 49.50858, min = 48.44830
```

```
 Indiv  birthdate   fitness      gene values
    1     494       49.76775     -4.99155     -4.98520
    2     486       49.30639     -4.99155     -4.93871
    3     446       49.35423     -4.99155     -4.94355
    4     437       49.55191     -4.99155     -4.96350
    5     491       49.55191     -4.99155     -4.96350
    6     469       49.76775     -4.99155     -4.98520
    7     445       49.76775     -4.99155     -4.98520
    8     410       48.44830     -4.99155     -4.85105
    9     497       49.76775     -4.99155     -4.98520
   10     471       49.80204     -4.99155     -4.98863

Population data after 1000 births (generation 100):
 Global fitness:  max = 49.91475, ave = 31.25352, min =  4.69584
 Local fitness:   max = 49.91475, ave = 49.91109, min = 49.90952
 Indiv  birthdate   fitness      gene values
    1     663       49.90952     -4.99155     -4.99940
    2     617       49.90952     -4.99155     -4.99940
    3     719       49.90952     -4.99155     -4.99940
    4     648       49.90952     -4.99155     -4.99940
    5     810       49.90952     -4.99155     -4.99940
    6     940       49.91475     -4.99155     -4.99992
    7     903       49.90952     -4.99155     -4.99940
    8     895       49.91475     -4.99155     -4.99992
    9     970       49.91475     -4.99155     -4.99992
   10     806       49.90952     -4.99155     -4.99940
```

So, these simulations strongly support scenario 1 above. What about the other scenarios? Is it possible for stable subpopulations to emerge? In EV a new individual replaces an existing one *only* if it has higher fitness. Hence, the emergence of stable subpopulations will happen only if they both involve identical 32-bit fitness values. Even small differences in fitness will guarantee that the group with higher fitness will take over the entire population. As a consequence, scenario 2 is possible but highly unlikely in EV.

Scenario 3, oscillation between peaks, can happen in the beginning to a limited extent as one peak temporarily attracts more members, and then loses them to more competitive members of the other peak. However, as the subpopulations get closer to these peaks, scenario 1 takes over.

Finally, scenario 4 (dynamic mediocrity) never occurs in EV, since mutations which take members away from a peak make them less fit and not likely to survive into the next generation. So, we see strong pressure to move away from the center of L2 and toward the boundaries, resulting in significant improvement in average fitness.

It should be clear by now how one can continue to explore the behavior of EV by presenting it with even more complex, multi-dimensional, multi-peaked landscapes. Alternatively, one can study the effects on behavior caused by changing various properties of EV, such as the population size, the form and rates of mutation, alternative forms of selection and repro-

duction, and so on. Some of these variations can be explored easily with EV as suggested in the exercises at the end of the chapter. Others require additional design and implementation decisions, and will be explored in subsequent chapters.

1.5 Evolutionary Systems as Problem Solvers

EV is not a particularly plausible evolutionary system from a biological point of view, and there are many ways to change EV that would make it a more realistic model of natural evolutionary systems. Similarly, from an artificial-life viewpoint, the emergent behavior of EV would be much more complex and interesting if the creatures were something other than vectors of real numbers, and if the landscapes were dynamic rather than static. However, the focus of this book is on exploring evolution as a computational tool. So, the question of interest here is: what computation (if any) is EV performing?

From an engineering perspective, systems are designed with goals in mind, functions to perform, and objectives to be met. Computer scientists design and implement algorithms for sorting, searching, optimizing, and so on. However, asking what the goals and purpose of evolution are immediately raises long-debated issues of philosophy and religion which, though interesting, are also beyond the scope of this book. What is clear, however, is that even a system as simple as EV appears to have considerable potential for use as the basis for designing interesting new algorithms that can search complex spaces, solve hard optimization problems, and are capable of adapting to changing environments.

To illustrate this briefly, consider the following simple change to EV:

```
EV-OPT:

  Generate an initial population of M individuals.

  Do until a stopping criterion is met:

    Select a member of the current population to be a parent.

    Use the selected parent to produce an offspring which is
    similar to but generally not a precise copy of the parent.

    Select a member of the population to die.

    End Do

  Return the individual with the highest global objective fitness.
```

By adding a stopping criterion and returning an "answer", the old EV code sheds its image as a simulation and takes on a new sense of purpose, namely one of searching for highly fit (possibly optimal) individuals. We now can think of EV-OPT in the same terms

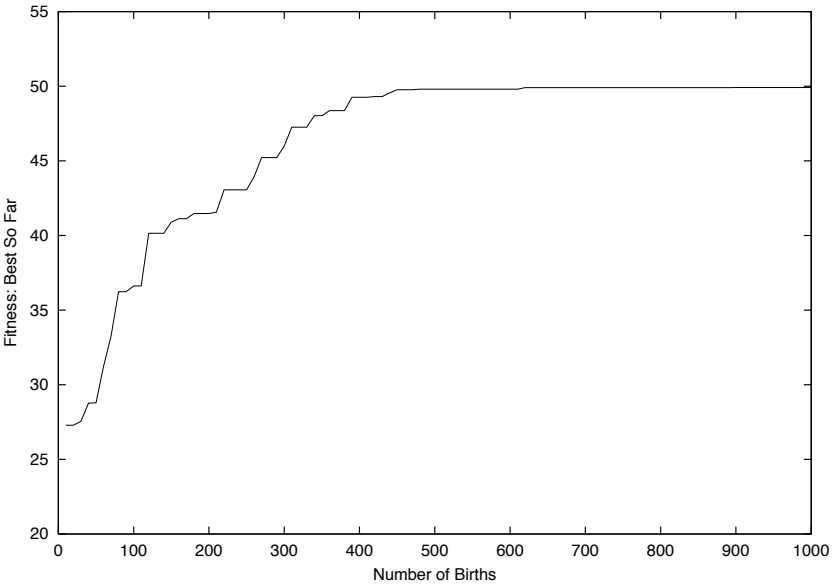

Figure 1.4: A best-so-far curve for EV on landscape L2.

as other algorithms we design. Will it always find an optimum? What kind of convergence properties does it have? Can we improve the rate of convergence?

One standard way of viewing the problem-solving behavior of an evolutionary algorithm is to plot the objective fitness of the best individual encountered as a function of evolutionary time (i.e., the number of births or generations). Figure 1.4 illustrates this for one run of EV on landscape L2. In this particular case, we observe that EV "found" an optimum after sampling about 500 points in the search space.

However, as we make "improvements" to our evolutionary problem solvers, we may make them even less plausible as models of natural evolutionary systems, and less interesting from an artificial-life point of view. Hence, this problem-solving viewpoint is precisely the issue which separates the field of evolutionary computation from its sister disciplines of evolutionary biology and artificial life.

However, focusing on evolutionary computation does not mean that the developments in these related fields are sufficiently irrelevant to be ignored. As we will see repeatedly throughout this book, natural evolutionary systems are a continuing source of inspiration for new ideas for better evolutionary algorithms, and the increasingly sophisticated behavior of artificial-life systems suggests new opportunities for evolutionary problem solvers.

1.6 Exercises

1. Explore the behavior of EV on some other interesting one-dimensional landscapes such as x^3, $sin(x)$, and $x * sin(x)$ that involve multiple peaks.

2. Explore the behavior of EV on multi-dimensional landscapes such as $x1^2 + x2^3$, $x1 * sin(x2)$, $(x1 - 2)(x2 + 5)$ that have interesting asymmetries and interactions among the variables.

3. Explore the behavior of EV when there is some noise in the fitness function such as $f(x) = x^2 + gauss(0, 0.01x^2)$.

4. The average value of the Gaussian (normal) distribution $G(0, \sigma)$ is, of course, zero. Show that the average *absolute* value of $G(0, \sigma)$ is $\sqrt{2/\pi} * \sigma$.

Hint: $Ave(|(G(0, \sigma)|) = \int_{-\infty}^{\infty} prob(x)\ |x|\ dx = 2\int_{0}^{\infty} prob(x)\ x\ dx$
where $prob(x) = \frac{1}{\sigma\sqrt{2\pi}}\ e^{-\frac{x^2}{2\sigma^2}}$