16

Theory

In this chapter we present a brief overview of some of the approaches taken to analysing and modelling the behaviour of evolutionary algorithms. The Holy Grail of these efforts is the formulation of predictive models describing the behaviour of an EA on arbitrary problems, and permitting the specification of the most efficient form of optimiser for any given problem. However, (at least in the authors' opinions) this is unlikely ever to be realised, and most researchers will currently happily settle for techniques that provide *any* verifiable insights into EA behaviour, even on simple test problems. The reason for what might seem like limited ambition lies in one simple fact: evolutionary algorithms are hugely complex systems, involving many random factors. Moreover, while the field of EAs is fairly young, it is worth noting that the field of population genetics and evolutionary theory has a head start of more than a hundred years, and is still battling against the barrier of complexity.

Full descriptions and analysis of the various techniques currently used to develop EA theory would require both an amount of space and an assumption of prior knowledge of mathematics and statistics that are unsuitable here. We therefore restrict ourselves to a fairly brief description of the principal methods and results which historically informed the field. We begin by describing some of the approaches taken to modelling EAs using a discrete representation (i.e., for combinatorial optimisation problems), before moving on to describe the techniques used for continuous representations. This chapter finishes with a description of an important theoretical result concerning all optimisation algorithms, the No Free Lunch (NFL) theorem.

For further details, we point the interested reader to 'bird's eye overviews' such as [140], and extensive monographs such as [52, 446, 353]. For a good overview of the most promising recent approaches and results we would suggest [234], or collections such as [63].

A.E. Eiben, J.E. Smith, *Introduction to Evolutionary Computing*,
Natural Computing Series, DOI 10.1007/978-3-662-44874-8_16

16.1 Competing Hyperplanes in Binary Spaces: The Schema Theorem

What Is a Schema?

Since Holland's initial analysis, two related concepts have dominated much of the theoretical analysis and thinking about GAs. These are the concepts of **schema** (plural schemata) and **building blocks**. A schema is simply a hyperplane in the search space, and the common representation of these for binary alphabets uses a third symbol — # the "don't care" symbol. Thus for a five-bit problem, the schema 11### is the hyperplane defined by having ones in its first two positions. All strings meeting this criterion are **instances**, or examples, of this schema (in this case there are $2^3 = 8$ of them). The fitness of a schema is the mean fitness of all strings that are examples of it; in practice this is often estimated from samples when there are many such strings. Global optimisation can be seen as the search for the schema with zero "don't care" symbols, which has the highest fitness.

Holland's initial work showed that the analysis of GA behaviour was far simpler if carried out in terms of schemata. This is an example of **aggregation** in which rather than model the evolution of all possible strings, they are grouped together in some way and the evolution of the aggregated variables is modelled. He showed that a string of length l is an example of 2^l schemata. Although in general there will not be as many as $\mu \cdot 2^l$ distinct schemata in a population of size μ, he derived an estimate that a population will usefully process $O(\mu^3)$ schemata. This result, known as **implicit parallelism** is widely quoted as being one of the main factors in the success of GAs.

Two features are used to describe schemata. The **order** is the number of positions in the schema that do not have the # sign. The **defining length** is the distance between the outermost defined positions (which equals the number of possible crossover points between them). Thus the schema H=1##0#1#0 has order $o(H) = 4$ and defining length $d(H) = 8 - 1 = 7$.

The number of examples of a schema in an evolving population depends on the effects of variation operators. While selection operators can only change the relative frequency of pre-existing examples, operators such as recombination and mutation can both create new examples and disrupt current ones. In what follows we use the notation $Pd(H, x)$ to denote the probability that the action of an operator x on an instance of a schema H is to destroy it, and $Ps(H)$ to denote the probability that a string containing an instance of schema H is selected.

Holland's Formulation for the SGA

Holland's analysis applied to the standard genetic algorithm (SGA) using fitness proportionate parent selection, one-point crossover (1X), and bitwise mutation, with a generational survivor selection. Considering a genotype of length l that contains an example of a schema H, the schema may be disrupted if the crossover point falls between the ends, which happens with probability

$$Pd(H, 1X) = \frac{d(H)}{(l-1)}.$$

The chance that bitwise mutation with rate P_m will disrupt the schema H is proportional to the order of the schema: $Pd(H, mutation) = 1-(1-P_m)^{o(H)}$. After expansion, and ignoring high-order terms in P_m, this approximates to

$$Pd(H, mutation) = o(H) \cdot P_m.$$

The probability of a schema being selected depends on the fitness of the individuals in which it appears relative to the total population fitness, and the number of examples present $n(H,t)$. Using $f(H)$ to represent the **fitness of the schema** H, defined as the mean fitness of individuals that are examples of schema H, and $<f>$ to denote the mean population fitness, we obtain:

$$Ps(H,t) = \frac{n(H,t) \cdot f(H)}{\mu \cdot < f >}.$$

μ independent samples are taken to create the next set of parents, so the expected number of instances of H in the population after selection is:

$$n'(H,t) = \mu \cdot Ps(H,t) = \frac{n(H,t) \cdot f(H)}{<f>}.$$

After normalising by μ (to make the result population-size independent), allowing for the disruptive effects of recombination and mutation derived above, and using an inequality to allow for the creation of new instances of H by the variation operators, the proportion $m(H)$ of individuals representing schema H at subsequent time-steps is given by:

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{<f>} \cdot \left[1 - \left(p_c \cdot \frac{d(H)}{l-1}\right)\right] \cdot \left[1 - p_m \cdot o(H)\right], \quad (16.1)$$

where p_c and p_m are the probabilities of applying crossover, and the bitwise mutation probability, respectively.

This is the **schema theorem**, and the original understanding of this result was that schemata of above-average fitness would increase their number of instances within the population from generation to generation. We can quantify this by noting that the condition for a schema to increase its representation is $m(H,t+1) > m(H,t)$ which is equivalent to:

$$\frac{f(H)}{<f>} > \left[1 - \left(p_c \cdot \frac{d(H)}{l-1}\right)\right] \cdot \left[1 - p_m \cdot o(H)\right].$$

Schema-Based Analysis of Variation Operators

Holland's original version of the schema theorem, as formulated above, was for one-point crossover and bitwise mutation. Following the rapid proliferation of alternative variation (particularly recombination) operators as the field expanded and diversified, a considerable body of results was developed to try and understand why some operators gave improved performance on certain problems. Particularly worthy of mention within this were two long-term research programs. Over a number of years, Spears and De Jong developed analytical results for $Pd(H, x)$ as a function of defining length $d(H)$ and order $o(H)$ for a number of different recombination and mutation operators [107, 108, 408, 412, 413, 414], which are brought together in [411].

Meanwhile, Eshelman and Schaffer conducted a series of empirical studies [157, 159, 161, 366] in which they compared the effects of mutation with various crossover operators on the performance of a GA. They introduced the notion of **operator bias** to describe the interdependence of $Pd(H, x)$ on $d(H), o(H)$ and x, which takes two forms:

- If an operator x displays **positional bias** it is more likely to keep together bits that are close together in the representation. This has the effect that given two schemata H_1, H_2 with $f(H_1) = f(H_2)$ and $d(H_1) < d(H_2)$, then $Pd(H_1, x) < Pd(H_2, x)$.
- By contrast, if an operator displays **distributional bias** then the probability that it will transmit a schema is a function of $o(H)$. One example of this is bitwise mutation, where, as we have seen, the probability of disruption increases with the order: $Pd(H, mutation) \approx Pm \cdot o(H)$. Another example is uniform crossover which will on average select half of the genes from one parent, and so is increasingly likely to disrupt a schema as the ratio $o(H)/l$ increases beyond 0.5.

Although these results provided valuable insight and have informed many practical implementations, it is worth bearing in mind that they are only considering the disruptive effects of operators. Analysis of the constructive effects of operators in creating new instances of a schema H are harder, since these effects depend heavily on the constitution of the current population. However, under some simplifying assumptions, Spears and De Jong [414] developed the surprising result that the expected number of instances of a schema destroyed by a recombination operator is equal to the expected number of instances created, for all recombination operators!

Walsh Analysis and Deception

If we return our attention to the derivation of the schema theorem, we can immediately see from an examination of the disruption probabilities given above that *all other things being equal*, short low-order schema have a greater chance of being transmitted to the next generation than longer or higher-order schema of the same mean fitness. This analysis has led to what has become known as the **building block hypothesis** [189, pp. 41–45]: that GAs begin by selecting amongst competing short low-order schemata, and then progressively combine them to create higher-order schemata, repeating this process until (hopefully) a schema of length $l-1$ and order l, i.e., the globally optimal string, is created and selected for. Note that for two schemata to compete they must have fixed bits (1 or 0) in the same positions. Thinking along these lines raised the obvious question: "What happens if the global optimum is *not* an example of the low-order schemata that have the highest mean fitness?".

To give an immediate example, let us consider a four-bit problem that has 0000 as its global optimum. It turns out that it is relatively simple to create the situation where all of the order-n schemata containing 0's in their defining positions are less fit than the corresponding schemata with 1's in those position, i.e., $f(0\#\#\#) < f(1\#\#\#)$, $f(\#0\#\#) < f(\#1\#\#)$, etc., right up to $f(\#000) < f(\#111)$, $f(0\#00) < f(1\#11)$, etc. All that is required to achieve this is that the fitness of a globally optimal string is sufficiently greater than all the other strings in every schema of which it is a member. In this case we might expect that every time the GA makes a decision between two order-n schemata, it is likely to make the wrong decision unless n=4.

This type of problem is known as **deceptive** and has been of great interest since it would appear to make life hard for a GA, in that the necessary building blocks for successful optimisation are not present. However, it has been postulated that if a fitness function is composed of a number of deceptive problems, then at least a GA using recombination offers the possibility that these can be solved independently and mixed via crossover. By comparison, an optimisation technique relying on local search continuously makes decisions on the basis of low-order schema, and so is far more likely to be 'fooled'. Note that we have not provided a formal definition of the conditions necessary for a function to be deceptive; much work has been done on this subject and slightly differing definitions exist [200, 403, 455].

The importance of deceptive problems to GA theory and analysis is debatable. At various stages some eminent practitioners have made claims that "the only challenging problems are deceptive" [93], (although this view may have been modified with hindsight), but others have argued forcibly against the relevance of deception. Grefenstette showed that it is simple to circumnavigate the problem of deception in GAs by looking for the best solution in each new generation and then creating its inverse [200]. Moreover, Smith and Smith created an abstract randomised test problem generator (NKPRS) in which the probability that a landscape was deceptive could be directly manipulated [404]. Their findings did not demonstrate that there was a correlation between the likelihood of deception and the ability of a standard GA to discover the global optimum.

Much of the work in this area makes use of **Walsh functions** to analyse fitnesses. This technique was first used for GA analysis in [51], but became more widely known after a series of important papers by Goldberg [187, 188]. These are a set of functions that provide a natural basis for the decomposition of a binary search landscape. They can be thought of as equivalent to the way that Fourier transforms decompose a complex signal in the time domain into a weighted sum of sinusoidal waves, which can be represented and manipulated in the frequency domain. Just as Fourier transforms form a vital part in a huge range of signal processing and other engineering applications, because sine functions are so easily manipulable, so Walsh transforms form an easily manipulable way of analysing binary search landscapes, with the added bonus that there is a natural correspondence between Walsh partitions (the equivalent of harmonic frequencies) and schemata. For more details on Walsh analysis the reader is directed to [187] or [353].

16.2 Criticisms and Recent Extensions of the Schema Theorem

Despite the attractiveness of the schema theorem as a description for how GAs work, it has come in for a good amount of criticism, and significant quantities of experimental evidence and theoretical arguments have been produced to dispute its importance. This is perhaps inevitable given that early on some rather extravagant claims were made by its adherents, and given the perhaps natural tendency of humans to take pot-shots at 'sacred cows'.

Ironically, empirical counterevidence was provided by Holland himself, in conjunction with Mitchell and Forrest, who created the **Royal Road functions** based on schema ideas in order to demonstrate the superiority of GAs over local search methods. Unfortunately, their results demonstrated that the opposite was in fact true [177]! However, this work did lead to the understanding of the phenomenon of **hitchhiking** whereby an unfavourable allele becomes established in the population because of an early association with an instance of a high-fitness schema.

Theoretical arguments against the value of the schema theorem and associated analysis have included:

- Even if it is correctly estimated, the rate of increase in representation of any given schema is not in fact exponential. This is because its selective advantage $f(H)/<f>$ decreases as its share of the population increases and the mean fitness rises accordingly.
- Eq. (16.1) applies to the *estimated* fitness of a given schema as averaged over all the instances in the current population, which might not be representative of the schema as a whole. Thus although the schema theorem is correct in predicting the frequency of a schema in the next generation, it can tell us almost *nothing* about the frequency in future generations, since

as the proportions of other schema change, so will the composition of the set of strings which represent H, and hence the estimates of $f(H)$.

- Findings that Holland's idea that fitness proportionate selection allocated optimal amounts of trials to competing schemata is incorrect [277, 359].
- The fact that the schema theorem ignores the constructive effects of operators. Altenberg [6] showed that in fact the schema theorem is a special case of Price's theorem in population genetics. This latter includes both constructive and disruptive terms. Whilst exact versions of the schema theorem have recently been derived [418], these currently remain somewhat intractable even for relatively simple test problems, although their use is starting to offer interesting new perspectives.

These arguments and more are summarised eloquently in [353, pp. 74–90]. We should point out that despite these criticisms, schemata represent a useful tool for understanding *some* of how GAs work, and we would wish to stress the vital role that Holland's insights into the importance of schemata have had in the development of genetic algorithms.

16.3 Gene Linkage: Identifying and Recombining Building Blocks

The Building Block Hypothesis offers an explanation of the operation of GAs as a process of discovering and putting together blocks of coadapted genes of increasing higher orders. To do this, it is necessary for the GA to discriminate between competing schemata on the basis of their estimated fitness. The Messy GA [191] was an attempt to explicitly construct an algorithm that worked in this fashion. The use of a representation that allowed variable length strings and removed the need to manipulate strings in the order of their expression began a focus on the notion of gene linkage (in this context gene is taken to mean the combination of a particular allele value and locus).

Munetomo and Goldberg [312] identify three approaches to the identification of linkage groups. The first of these they refer to as the "direct detection of bias in probability distributions", and is exemplified by Estimation of Distribution Algorithms described in Section 6.8. Common to all of these approaches is the notion of first identifying a factorisation of the problem into a number of subgroups, such that a given statistical criterion is minimised, based on the current population. This corresponds to learning a linkage model of the problem. Once these models have been derived, conditional probabilities of gene frequencies within the linkage groups are calculated, and a new population is generated based on these, replacing the traditional recombination and mutation steps of an EA. It should be emphasised that these EDA approaches are based on statistical modelling rather than on a schema-based analysis. However, since they implicitly construct a linkage analysis of the problem, it would be inappropriate not to mention them here.

The other two approaches identified by Munetomo and Goldberg use more traditional recombination and mutation stages, but bias the recombination operator to use linkage information.

In [243, 312] first-order statistics based on pairwise perturbation of allele values are used to identify the blocks of linked genes that algorithms manipulate. Similar statistics are used in a number of other schemes such as [438].

The third approach identified does not calculate statistics on the gene interactions based on perturbations, but rather adapts linkage groups explicitly or implicitly via the adaptation of recombination operators. Examples of this approach that explicitly adapt linkage models can be seen in [209, 362, 393, 394, 395]. A mathematical model of the linkage models of different operators, together with an investigation of how the adaptation of linkage must happen at an appropriate level (see Sect. 8.3.4 for a discussion of the issue of the scope of adaptation), can be found in [385].

16.4 Dynamical Systems

The **dynamical systems** approach to modelling EAs in finite search spaces has principally been concerned with genetic algorithms because of their (relative) simplicity. Michael Voses established the basic formalisms and results in a string of papers culminating in the publication of his book [446]. This work has been taken on and extended by a number of authors (see, for example, the proceedings of the Foundations of Genetic Algorithms workshops [38, 285, 341]). The approach can be characterised as follows:

- Start with an n-dimensional vector $\overline{p}$, where n is the size of the search space, and the component p_i^t represents the proportion of the population that is of type i at iteration t.
- Construct a **mixing matrix** M representing the effects of recombination and mutation, and a **selection matrix** F representing the effects of the selection operator on each string for a given fitness function.
- Compose a genetic operator $G = F \circ M$ as the matrix product of these two functions.
- The action of the GA to generate the next population can then be characterised as the application of this operator G to the current population: $\overline{p}^{t+1} = G\overline{p}^t$.

Under this scheme the population can be envisaged as a point on what is known as the simplex: a surface in n-dimensional space made up of all the possible vectors whose components sum to 1.0 and are nonnegative. The form of G governs the way that a population will trace a trajectory on this surface as it evolves. A common way of visualising this approach is to think of G as defining a ‘force-field’ over the simplex describing the direction and intensity of the forces of evolution acting on a population. The form of G alone determines which points on the surface act as **attractors** towards which the population is drawn; and analytical analysis of G, and its constituents F and M, has led to many insights into GA behaviour.

Vose and Liepens [447] presented models for F and M under fitness proportionate selection, one-point crossover and bitwise mutation, and these have been extended to other operators in [446]. One of the insights gained by analysing the form of M is that schemata provided a natural way of aggregating strings into equivalence classes under recombination and mutation, which provides a nice tie-in to Holland's ideas.

Other authors have examined a number of alternative ways of aggregating the elements in the search space into a smaller number of equivalence classes, so as to make the models more amenable to solution. Using this approach, a number of important results have been derived, explaining facets of behaviour such as the punctuated equilibria effect (described qualitatively in [447] but expanded and including for the first time *accurate* predictions of the time spent between the discovery of new fitness levels in [439]). These ideas have also been applied to model mechanisms such as self-adaptive mutation in binary coded GAs [383, 386].

It is worth pointing out that while this model exactly predicts the *expected* proportions of different individuals present in evolving populations, these values can only be attained if the population size is infinite. For this reason this approach falls into a class known as **infinite population models**. For finite populations, the evolving vectors $\overline{p}$ can be thought of as representing the probability distribution from which μ independent samples are drawn to generate the next population. Because the smallest proportion that can be present in a real population has a size $1/\mu$, this effectively constrains the population to move between a subset of points on the simplex representing a lattice of size $1/\mu$. This means that, given an initial population, the trajectory predicted may not actually be attainable, and corrections must be made for finite population effects. This work is still ongoing.

16.5 Markov Chain Analysis

Markov chain analysis is a well-established technique that is used to study the properties and behaviour of stochastic systems. A good description can be found in many textbooks on stochastic processes [216]. For our purposes it is sufficient to note that we can describe a system as a discrete-time **Markov chain** provided that the following conditions are met:

- At any given time the system is in one of a finite number (N) of states.
- The probability that the system will be in any given state X^{t+1} in the next iteration is solely determined by the state that it is in at the current iteration X^t, regardless of the previous sequence of states.

The impact of the second condition is that we can define a **transition matrix** Q where the entry Q_{ij} contains the probability of moving from state

i to state j in a single step ($i, j \in \{1, \ldots, N\}$). It is simple to show that the probability that after n steps the system has moved from state i to state j is given by the (i, j)th entry of matrix Q^n. A number of well-known theorems and proofs exist for making predictions of the behaviour of Markov chains.

There are a finite number of ways in which we can select a finite sized population from a finite search space, so we can treat any EA working within such a representation as a Markov chain whose states represent the different possible populations, and a number of authors have used these techniques to study evolutionary algorithms.

As early as in 1989 Eiben et al. [1, 129] proposed a Markov model for the abstract genetic algorithm built from a choice, a production, and a selection function, and used it to establish convergence properties. In contemporary terminology it is a general framework for EAs based on parent selection, variation, and survivor selection, respectively. It has been proved that an EA optimising a function over an arbitrary finite space converges to an optimum with probability 1 under some rather permissive conditions. Simplifying and reformulating the results, it is shown that if, in any given population,

- every individual has a nonzero probability of selection as a parent, and
- every individual has a nonzero probability of selection as a survivor, and
- the survival selection mechanism is elitist, and
- any solution can be created by the action of variation operators with a nonzero probability,

then the nth generation certainly contains the global optimum for some n.

Rudolph [357] tightened the assumptions and showed that a genetic algorithm with nonzero mutation and elitism will always converge to the globally optimal solution, but that this would not necessarily happen if elitism was not used. In [358] the convergence theorems are extended to EAs working in arbitrary (e.g., continuous) search spaces.

A number of authors have proposed exact formulations for the transition matrices Q of binary coded genetic algorithms with fitness proportionate selection, one-point crossover, and bit-flipping mutation [99, 321]. They essentially work by decomposing the action of a GA into two functions, one of which encompasses recombination and mutation (and is purely a function of the crossover probability and mutation rate), and the other that represents the action of the selection operator (which encompasses information about the fitness function). These represent a significant step towards developing a general theory; however, their usefulness is limited by the fact that the associated transition matrices are enormous: for an l-bit problem there are $\begin{pmatrix} \mu + 2^l - 1 \\ 2^l - 1 \end{pmatrix}$ possible populations of size μ and this many rows and columns in the transition matrix.

It is left as an exercise for the reader to calculate the size of the transition matrix for a ten-bit problem with ten members in the population, in order to get a feel for how likely it is that advances in computing will make it possible to manipulate these matrices.

16.6 Statistical Mechanics Approaches

The **statistical mechanics** approach to modelling EA behaviour was inspired by the way that complex systems consisting of ensembles of many smaller parts have been modelled in physics. Rather than trying to trace the behaviour of all the elements of a system (the **microscopic** approach), this approach focuses on modelling the behaviour of a few variables that characterise the system. This is known as the **macroscopic approach**. There are obvious links to the aggregating versions of the dynamical systems approach described above; however, the quantities modelled are related to the cumulants of the variables of interest [345, 346, 348, 354].

Thus if we are interested in the fitness of an evolving population, equations are derived that yield the progress of the moments of fitness $<f>, <f^2>$, $<f^3>$, and so on (where the braces $<>$ denote that the mean is taken over the set of possible populations) under the effects of selection and variation. From these properties, cumulants such as the mean ($< f >$ by definition), variance, skewness, etc., of the evolving population can be predicted as a function of time. Note that these predictions are necessarily approximations whose accuracy depends on the number of moments modelled.

The equations derived rely on various 'tricks' from the statistical mechanics literature and are predominantly for a particular form of selection (Boltzmann selection). The approach does not pretend to offer predictions other than of the population mean, variance and so on, so it cannot be used for all the aspects of behaviour one might desire to model. These techniques are nevertheless impressively accurate at predicting the behaviour of real GAs on a variety of simple test functions. In [347] Prügel-Bennett compares this approach with a dynamical systems approach based on aggregating fitness classes and concludes that the latter approach is less accurate at predicting dynamic behaviour of the population mean fitness (as opposed to the long-term limit) because the variables that it tracks are not representative as a result of the averaging process. Clearly this work deserves further study.

16.7 Reductionist Approaches

So far we have described a number of methods for modelling the behaviour of EAs that attempt to make predictions about the composition of the next population by considering the effect of all the genetic operators on the current population. We could describe these as holistic approaches, since they explicitly recognise that there will be interactions between the effects of different operators on the evolving population. An unfortunate side effect of this holistic approach is that either the resulting systems become very difficult to manipulate, as a result of their sheer size, or necessarily involve approximations and may not model all of the variables that we would like to predict.

An alternative methodology is to take a reductionist approach, and examine parts of the system separately. Although ultimately flawed in neglecting interaction effects, this approach is common to many branches of physics and engineering, where it has been used to yield frequently accurate predictions and insights, provided that a suitable decomposition of the system is made.

The advantage of taking a reductionist approach is that frequently it is possible to derive analytical results and insights when only a part of the problem is considered. A typical division is between selection and variation. A great deal of work has been done on characterising the effects of different selection operators, which can be thought of as complementary to the work described in Section 16.1.

Goldberg and Deb [190] introduced the concept of **takeover time**, which is the number of generations needed for a single copy of the fittest string to completely take over the population in a "selecto-EA" (i.e., one in which no variation operators are used). This work has been extended to cover a variety of different mechanisms for parental and survivor selection, using a variety of theoretical tools such as difference equations, order statistics, and Markov chains [19, 20, 21, 58, 78, 79, 360, 400].

Parallel to this, Goldberg, Thierens, and others examined what they called the **mixing time**, which characterises the speed at which recombination brings together building blocks initially present in different members of a population [430]. Their essential insight is that in order to build a well-performing EA, in particular a GA, it is necessary for the mixing time to be less than the takeover time, so that all possible combinations of the building blocks present can be tried before one fitter string takes over the population and removes some of them. While the rigour of this approach can be debated, it does have the immense benefit of providing practical guidelines for population sizing, operator probabilities, choice of selection methods, and so on, which can be used to help design an effective EA for new applications.

16.8 Black Box Analsyis

One of the approaches which has yielded the most promising advances since the first edition of this book was written has been the 'black box complexity' approach introduced by Droste, Jansen and Wegener [120]. A good recent review can be found in [234], or in collections such as [63]. The essence of this approach is to model the run-time complexity of algorithms on specific functions – that is to say on their expected time from an arbitrary starting point to reaching the global optima. This is done by modelling the process as a system of steps whose likelihood can be expressed and then deriving upper and lower bounds on the run-time from these equations. This approach has lead to many useful insights on the behaviour of population-based methods, and has in part settled some long-running debates within the field – for example, by illustrating non-artificial problems on which crossover is provably useful [118].

16.9 Analysing EAs in Continuous Search Spaces

In contrast to the situation with discrete search spaces, the state of theory for continuous search spaces, and evolution strategies in particular, is fairly advanced. As noted in Section 16.5, Rudolph has shown the existence of global proofs of convergence also in such spaces [358], since the evolution of the population is itself a Markov process. Unfortunately, it turns out that the Chapman–Kolmogorov equation describing this is intractable, so the population probability distribution as a function of time cannot be determined directly. However, it turns out that much of the dynamics of ESs can be recovered from simpler models concerning the evolution of two macroscopic variables, and many theoretical results have been obtained on this basis.

The first of the variables modelled is the **progress rate**, which measures the distance of the centre of mass of the population from the global optimum (in variable space) as a function of time. The second is the **quality gain**, which measures the expected improvement in fitness between generations.

Most of this analysis has concerned variants of two fitness functions, the **sphere model**: $f(\overline{x}) = \sum_i x_i^n$ for some n, and the **corridor model** [373]. The latter takes various forms but essentially contains a single direction in which fitness is improving, hence the name. Since an arbitrary fitness function in a continuous space can usually be expanded (using a Taylor expansion) to a sum of simpler terms, the vicinity of a local optimum of one of these models is often a good approximation to the *local* landscape.

The continuous nature of the search space, coupled with the use of normally distributed mutations and well-known results from order statistics, have permitted a *relatively* straightforward derivation of equations describing the motion of the two macroscopic variables over time as a function of the values of μ, λ, and σ, starting with Rechenberg's analysis of the (1+1) ES on the sphere model, from which he derived the 1/5 success rule [352]. Following from this, the principles of self-adaptation and multimembered strategies have also been analysed. A thorough overview of these results is given in [53].

16.10 No Free Lunch Theorem

By now we hope the reader will have realised that the search for a mathematical model of EAs, which will permit us to make accurate predictions of a given algorithm on any given problem, is still a daunting distance from its goal. Whilst the tools are now in place to make some accurate predictions of some aspects of behaviour on some problems, these are often restricted to simple problems for which an EA is almost certainly not the most efficient algorithm anyway.

However, a recent line of work has come up with a result that allows us to make some statements about the comparative performance of different algorithms across all problems: they are all the same! This result is known as the **No Free Lunch theorem** (NFL) [467]. In layperson's terms it says that if we average over the space of all possible problems, then all nonrevisiting **black box algorithms** will exhibit the same performance.

By nonrevisiting we mean that the algorithm does not generate and test the same point in the search space twice. Although not typically a feature of EAs, this can simply be achieved by implementing an archive of all solutions ever seen, and then each time we generate an offspring discarding it and repeating the process if it already exists in the archive. An alternative approach (taken by Wolpert and Macready in their analysis) is to view performance as the number of distinct calls to the evaluation function. In this case we still need an archive, but we can allow duplicates in the population. By black box algorithms we mean those that do not incorporate any problem or instance-specific knowledge.

There has been some considerable debate about the utility of the No Free Lunch theorem, often centred around the question of whether the set of problems that we are likely to try to tackle with EAs is representative of all problems, or forms some special subset. However, they have come to be widely accepted, and the following lessons can be drawn:

- If we invent a new algorithm and it appears to be the best ever at solving some particular class of problems, then it will pay for this by performing poorly at some others. This suggests that a careful strategy is required to evaluate new operators and algorithms, as discussed in Chap. 9.
- *For a given problem* we can circumvent the NFL theorem by incorporating problem-specific knowledge. This of course leads us towards memetic algorithms (cf. Chap. 10).

For exercises and recommended reading for this chapter, please visit www.evolutionarycomputation.org.