Chapter 1
Evolutionary Computation: Centralized, Parallel or Collaborative

Heinz Mühlenbein

Abstract This chapter discusses the nature and the importance of spatial interactions in evolutionary computation. The current state of evolution theories is discussed. An interaction model is investigated we have called Darwin's *continent-island cycle* conjecture. Darwin argued that such a cycle is the most efficient for successful evolution. This bold conjecture has not yet been noticed in science. We confirm Darwin's conjecture using an evolutionary game based on the iterated prisoner's dilemma. A different interaction scheme, called the stepping-stone model is used by the Parallel Genetic Algorithm PGA. The PGA is used to solve combinatorial optimization problems. Then the Breeder Genetic Algorithm BGA used for global optimization of continuous functions is described. The BGA uses competition between subpopulations applying different strategies. This kind of interaction is found in ecological systems.

1.1 Introduction

Modeling evolutionary principles found in nature in order to make the development of powerful problem solving programs easier, or even to create human-like intelligence was tried already in the beginning of the computer area. There have been followed at least three approaches to achieve this goal.

1. **Use a theory** - develop a theory of problem solving and implement it on a computer
2. **Copy the brain** - analyze the human brain and make a copy of it on a computer
3. **Copy natural evolution** - analyze natural evolution and implement the most important evolutionary forces on a computer

In the history of artificial intelligence research one of the three approaches was dominant at any one time. The second and third approach depend on

Fraunhofer Institut Autonomous intelligent Systems Schloss Birlinghoven 53757 Sankt Augustin, Germany, e-mail: heinz.muehlenbein@online.de

biological knowledge. In this chapter I will concentrate on the third approach. It relies on theories of evolution and of computation. The theory of computation is well advanced, so the problems of evolutionary computation lie in implementing theories of evolution. If a convincing constructive (or even mathematical) theory of evolution would exist, then evolutionary computation would be just a matter of implementation - which of the major evolutionary forces to implement in what detail.

But does biology possess a constructive theory of evolution? Here the opinions differ extremely. The main stream theory of evolution is called *New* or *Modern Synthesis*. Its followers claim that it reconciles Darwin's idea of continuous small variations with gene flows derived from population genetics. The second major force of the Modern Synthesis is Darwin's concept of *natural selection*. But are these two forces sufficient to explain the wonders of evolution at least in some broad terms?

There is no doubt that the modern synthesis is able to explain the change of gene frequencies on a small time scale. If there is enough diversification, then the theory correctly predicts further changes for a short time. But can it explain the evolution on a large time scale with new species arising and old species vanishing?

The outline of the chapter is as follows. First I recall the state of art of evolution theories, because they are used as models for evolutionary computation. Then I describe different genetic algorithms, centralized, parallel and co-evolutionary. The major part of the chapter deals with the investigation of Darwin's conviction that space is as important as time for evolution to take place. Especially we[1] will analyze an important conjecture of Darwin which has been unnoticed sofar. We have called it the *Continent-island cycle conjecture*. This conjecture is analyzed using evolutionary games. Here a number of different spatial distributions are compared. Then I describe the parallel genetic algorithm PGA and its use in combinatorial optimization. In the final section co-evolution of sub-populations is used for continuous optimization.

The term collaboration does not appear in textbooks of biology. In a restricted form collaboration is investigated in ecology. Collaboration in the general sense is considered to be an important component of human societies and therefore part of sociology. Biology researches cooperation driven by interactions - between individuals, between species, between geographic distributed sub-populations, within insect colonies etc. In this chapter we investigate spatial distributions which vary over time. The most interesting distribution is the continent-island cycle. This might also be a model for successful collaboration in human societies.

1.2 Darwinism - the unfinished theory

Darwin's book "The Origin of Species by Means of Natural Selection" had undoubtedly an immense influence on biology and modern thinking in general. But it should have been seen as the beginning of a new theory of evolution, not a solution. The further development of Darwin's ideas has been

[1] The research has been done together with many researchers. Therefore I use mainly we throughout the chapter.

hindered especially in the Anglo-Saxon countries because of the battle of its supporters against some orthodox believers of religion[2].

In Germany Ernst Haeckel was s strong advocate of Darwin's theory. Nevertheless he wrote as early as 1863 - only four years after the publication of the Origin: "Darwin's evolution theory is in no ways finished, instead it gives only the first outline of a future theory. On one hand we do not know the certainly many relations, which are probably not of a smaller weight for the origin of species than natural selection, which was emphasized far too one-sided by Darwin. And in many cases the external conditions of existence of an-organic nature like climate and habitat, geographic and topographic conditions, to which the organism have to adapt, should be not less important than these relations....Another, and no doubt the most important shortcoming of Darwin's theory lies in the fact, that it gives *no indication of the spontaneous creation or the first genesis of the one or the few oldest organisms from which all other organisms developed (19)*[3]".

It is outside the scope of this paper to discuss the above problems in detail. They are still controversial in biology. In order to refresh the memory of the reader I recall some important terms in evolutionary biology

- **Genotype:** The molecular basis of inheritance as defined by genes and chromosomes.
- **Phenotype:** The actual appearance of the living beings.
- **Species:** A group of organisms capable of inter-breeding and producing fertile offspring. More precise measures are based on the similarity of DNA or morphology.

Another important concept in population genetics is the *fitness*. It describes the capability of an individual of certain genotype to reproduce, and usually is equal to the proportion of individual genes in the *next* generation. An individual's fitness is manifested through its phenotype. As phenotype is affected by both genes *and* environment, the fitness of different individuals with the same genotype are not necessarily equal, but depend on the environment on which the individuals live. However, the fitness of the genotype is considered to be an averaged quantity, it will reflect the outcomes of all individuals with that genotype.

This is a very careful definition, but how can this fitness be measured? Its needs the next generation! I will not discuss this complicated issue further. All mathematical models of population genetics assume that the fitness is given. In the simplest case of a single gene with two alleles a and A, we have the genotypes with genotypes aa, aA, AA with corresponding fitness values w_{00}, w_{01}, w_{11}.

In order to illustrate the current state of art of evolution theories, I shortly describe two representative examples. The first one is expressed in the book of Maynard Smith and Szathmary (48). They see evolution as the evolution of complexity in terms of genetic information and how it is stored, transmitted, and translated. This approach has led them to identify several *major transitions*, starting with the origin of life and ending with the origin of human language (see table 1.1).

[2] This controversy is still not settled, if one considers the many supporters of "intelligent design" in the US.

[3] translation by the author

before	→ after
replicator molecules	→ population of molecules in compartments
independent replicator	→ chromosomes
RNA as gene and enzyme	→ DNA and protein
procaryote	→ eucaryote
asexual clones	→ sexual population
protist	→ plants, animals, fungi
solitary individuals	→ colonies
societies of primates	→ human societies

Table 1.1 Major transitions in evolution (48)

The authors "solve" some of the transition problems with a very narrow version of the modern synthesis. "We are supporters of the gene centered approach proposed by Williams and refined by Dawkins." In the gene centered approach, also called the *selfish gene concept* (7), the genes are the major actors. They possess an internal force to proliferate as much as possible.

Let me illustrate the gene centered approach with the *kin selection* concept. In the gene centered approach fitness measures the quantities of copies of the genes of an individual in the next generation. It doesn't really matter how the genes arrive in the next generation. That is, for an individual it is equal beneficial to reproduce itself, or to help relatives with similar genes to reproduce, as long as at least the same number of copies of the individual's genes get passed on to the next generation. Selection which promotes this kind of helper behavior is called kin selection. It has even been put into a mathematical rule by Hamilton (20)! An altruistic act will be done if

$$C < r * B \tag{1.1}$$

Here C means the cost in fitness to the actor, B the benefit in fitness and r the relatedness. Let us discuss a simple example. Consider a father and his children, which are drowning. Here $r = 0.5$. Let us assume that the father has to make a sacrifice, this means $C = 1$. First assume $B = 1$. Then the father will not make a sacrifice for a single child, but it needs three children at least. But is the father is not able to father new children, then he makes the sacrifice for a single child! (see also the discussion of altruism in (48).

The selfish gene concept has been opposed by a small group in biology, most notably the late Stephen J. Gould. Recently even philosophers of science formulate a basic critic. "The synthetic theory bypassed what were at the time intractable questions of the actual relationship between stretches of chromosomes and phenotypic traits. Although it was accepted that genes must, in reality, generate phenotypic differences through interaction with other genes and other factors in development, genes were treated as *black boxes* that could be relied on to produce phenotypic variation with which they were known to correlate (17)."

A more complex theory of evolution is contained in the book by Jablonka and Zeligowski. The title is: "Evolution in Four Dimension" (23). The four dimensions are

- The genetic inheritance system
- The epigenetic inheritance system
- The behavioral inheritance system
- The symbolic inheritance system

The genetic dimension dominates the current research. It culminated in sequencing the human genome. The epigenetic inheritance system is less understood and varies considerably. Although their DNA sequences remain unchanged during development, cells nevertheless acquire information that they can pass to their progeny. Let me give a simple example. Liver cells, skin cells, and kidney cells, look different, behave differently, and function differently, yet they contain all the same genetic information. The difference are the consequences of events that occurred during the developmental history of each type of cell and determined which genes are turned on. Now, when liver cells divide their daughter cells are liver cells, and so on. Although their DNA sequences remain unchanged during development, cells nevertheless acquire information that they can pass to their progeny. This information is transmitted by the *epigenetic inheritance system*. Sometimes the information can be transmitted even transgenerationally. The existence of epigenetic inheritance was barely recognized until the mid-1970s.

Behavioral inheritance is still controversial. Most evolutionists stress the genetic basis of behavior (see Maynard Smith discussed earlier). They maintain that behavioral strategies are to a large extent genetically determined and almost independent of each other. Each behavior has been shaped through natural selection of genes that led to the construction of a specific behavioral module in the brain.

The fourth dimension, the symbolic interaction system, is unique to man. There exists only a few attempts of a mathematical theory of the symbolic inheritance system. Most notably is the theory of "cultural transmission" (4).

But what makes even this very condensed theory so difficult is the *interaction of all four inheritance systems*. The genetic inheritance system does not dominate or even is foundation of the other three systems. The four dimensions cannot be analyzed separately, it is their complex interaction which defines the success of the individual.

Metaphorically speaking: Each organism travels on a unique trace in this four dimensional space.

1.2.1 The system view of evolution

One of the major weakness of the modern synthesis is the *separation of the individuals and the environment*. The fitness is averaged over individuals and environments. Let $\mathbf{O}(\mathbf{t}) = (O_1(t), \dots O_N(t))$ denote the vector of individuals at generation t. Then we can formulate a simple system model. Each individual $O_i(t)$ (mainly characterized by its genotype) is assigned a fitness f predicting the performance of this individual within the environment E and given the other individuals. Then the evolution can be written as:

$$O_i(t+1) = f(\mathbf{O}(\mathbf{t}), E(t)) \tag{1.2}$$

$$E(t+1) = g(E(t) \tag{1.3}$$

It seems impossible to obtain numerical values for this fitness. Therefore theoretical biology has made many simplifications. The environment is kept fixed, i.e $g(E(t)) = const$, the influence of other individuals is described by some averages of the population, etc.. The above model is still too simple, because each individual is *developing in a close interaction with its environment*.

The model given by 1.2 has not yet been used in population genetics, but specialized cases are applied commonly in *population dynamics* (21) or *ecology*. Given two species with population sizes N and M, the following equations are used

$$N(t+1) = F(N(t), M(t)) \tag{1.4}$$
$$M(t+1) = G(N(t), M(t)) \tag{1.5}$$

The population sizes of the next generation depends on the interaction of the two species at generation t. The interaction can be positive, this means that both species are supporting each other. If the interaction is negative we have the classical predator-prey system (18).

The development problem in evolutionary models has been addressed recently by the *developmental system theory* (41). Unfortunately the theory is very informal, it has been formulated from a philosopher's point of view. Therefore I will describe the nucleus of an evolution theory as it has been stated by Anatol Rapaport (44).

The theory is based on the concept of an *organism*. "According to a soft definition, a system is a portion of the world that is perceived as a unit and that is able to maintain its identity in spite of changes going on in it. An example of a system par excellence is a living organism. But a city, a nation, a business firm, a university are organisms of a sort. These systems are too complex to be described in terms of succession of states or by mathematical methods. Nevertheless they can be subjected to methodological investigations (44)."

Rapaport then defines: "Three fundamental properties of an organism appear in all organism-like systems. Each has a structure. That is, it consists of inter-related parts. It maintains a short-term steady state. That is to say, it reacts to changes in the environment in whatever way is required to maintain its integrity. It functions. It undergoes slow, long term changes. It grows, develops, or evolves. Or it degenerates, disintegrates, dies. Organisms, ecological systems, nations, institutions, all have these three attributes: *structure, function, and history*, or, if you will, *being, acting, and becoming*."

Taking the importance of individual development into account, I divide becoming into *developing and evolving*. Development is the process creating a grown-up individual in a given environment from an fertilized egg . Evolution is the change of the species in many generations.

The neuro-biologist Humberto Matura (32) also tried to find criteria for what characterizes life. He proposed a new concept which he called *autopoiesis*. It is derived from Greek "autos=self" and "poiein=to make". The theory remained rather informal and speculative. Therefore the concept was not very successful in biology, but it is very popular in other sciences, especially in sociology.

The major conclusion of this section is: *There exists no theory of evolution today which could serve as a computer model for evolving artificial creatures with problem solving capacities.* Given this situation it is no surprise that early models of evolutionary computation are not based on detailed biological models, but use biological terms only metaphorically. This observation is still valid.

1.3 Evolutionary algorithms - centralized, parallel or collaborative

There exist a myriad of evolutionary algorithms which model parts of general evolutionary models described in the previous section. The most popular algorithm is the genetic algorithm GA which models evolution by sexual reproduction and natural selection. The GA was invented by Holland (22). The optimization problem is given by a fitness function $F(x)$.

Genetic Algorithm

```
STEP0: Define a genetic representation of the problem; set $t = 0$
STEP1: Create an initial population $P(0) = x_1^0, \dots x_N^0$
STEP2: Compute the average fitness $\overline{F} = \sum_i^N F(x_i)/N$. Assign each individual the normalized fitness value $F(x_i^t)/\overline{F}$
STEP3: Assign each $x_i$ a probability $p(x_i, t)$ proportional to its normalized fitness. Using this distribution, randomly select $N$ vectors from $P(t)$. This gives the set $S(t)$
STEP4: Pair all of the vectors in $S(t)$ at random forming $N/2$ pairs. Apply crossover with probability $p_{cross}$ to each pair and other genetic operators such as mutation, forming a new population $P_{t+1}$
STEP5: Set $t = t + 1$, return to STEP2
```

In the simplest case the genetic representation is just a bit-string of length n, the *chromosome*. The positions of the strings are called *locus* of the chromosome. The variable at a locus is called *gene*, its value *allele*. The set of chromosomes is called the *genotype* which defines a *phenotype* (the individual) with a certain fitness. The *crossover operator* links two searches. Part of the chromosome of one individual (search point) is inserted into the second chromosome giving a new individual (search point). We will later show with examples why and when crossover helps the search.

A genetic algorithm is a *parallel random search with centralized control*. The centralized part is the selection schedule and the mating. For the selection the average fitness of the population is needed. The result is a highly synchronized algorithm, which is difficult to implement efficiently on parallel computers.

In our parallel genetic algorithm PGA we use a *distributed selection scheme*. This is achieved as follows. Each individual does the selection by itself. It looks for a partner in its neighborhood only. The set of neighborhoods defines a spatial population structure. Thus the PGA runs *totally asynchronous*, there is no synchronization

Our second major change can now easily be understood. Each individual is active and not acted on. It may improve its fitness during its lifetime by performing a *local search*. The generic PGA can be described as follows

Parallel genetic algorithm

```
STEP0: Define a genetic representation of the problem
STEP1: Create an initial population and its population structure
STEP2: Each individual does local hill-climbing
STEP3: Each individual selects a partner for mating in its neighborhood
STEP4: An offspring is created with genetic crossover of the parents
```

STEP5: The offspring does local hill-climbing. It replaces the parent, if it is better than some criterion (acceptance)
STEP6: If not finished, return to STEP3.

Because each individual runs independently on a processor, it may use a specific local hill-climbing method. This feature will be important for problems, where the efficiency of a particular hill-climbing method depends on the problem instance.

The PGA can be described as a parallel search with information exchange between neighbors in the space. Because the neighborhoods overlap, a *diffusion process* takes place. In the PGA all decisions are made by the individuals themselves. Therefore the *PGA is a totally distributed algorithm without any central control.*

There have been several other attempts to implement a parallel genetic algorithm. Most of the algorithms run *k* identical standard genetic algorithms in parallel, one run per processor. They differ in the linkage of the runs. Tanese (49) introduces two *migration* parameters: the *migration interval*, the number of generations between each migration, and the *migrationrate*, the percentage of individuals selected for migration. The subpopulations are configured as a binary n-cube. In the implementation of Cohoon (5) it is assumed that each subpopulation is connected to each other. The algorithm from Manderick et al. (30) has been derived from our PGA. In this algorithm the individuals of the population are placed on a planar grid and selection and crossing-over are restricted to small neighborhoods on that grid.

All but Manderick's algorithm use subpopulations that are densely connected. We have shown in (34) why restricted connections like a ring are better for the parallel genetic algorithm. All the above parallel algorithms do not use hill-climbing, which is one of the most important parts of our PGA.

An extension of the PGA, where subpopulations are used instead of single individuals, has been described in (40). This algorithm outperforms the standard GA by far in the case of function optimization. The idea has been refined in (45) where competing sub-populations of varying sizes are used.

Before we report applications of the PGA, we discuss the importance of spatial structures and collaboration in biology.

1.4 Co-evolution and collaboration in evolution

Collaboration is not part of evolution theories, it plays the dominant role in a separate field, today called *ecology*. Evolution theories try to understand how collaboration evolves and changes, whereas ecology models the existing collaboration in a habitat, trying to understand its influence on the size of the populations. A famous example are predator-prey cycles described in equation 1.4. In this section we discuss the interaction of populations in spatial distributions and its importance for the speed of evolution.

1.4.1 Darwin revisited

First we will show that Darwin was aware of the importance of external conditions of an-organic nature like the geographic distribution. In the section "Circumstances favourable and unfavourable to Natural Selection" Darwin tries to describe the influence of intercrossing, isolation and number of individuals on the production of new organic forms. Darwin mentions that this is an extremely intricate subject. He argues very carefully, for example: "For to ascertain whether a small isolated area or a large open area like a continent has been the most favourable for the production of new organic forms, we ought to make comparisons within equal times; and this we are incapable of doing".

Simulation makes such a comparison possible. It is of course impossible to simulate the real evolution of nature, so we have to find an artificial environment which is nevertheless complex enough to model the important aspects of evolution. As a first step we have decided to simulate an artificial population where each individual plays a two-person game against the other individuals. We have selected the Iterated Prisoner's Dilemma IPD, because it is surprisingly complex. The simulation will be discussed in section 1.5.

After a lot of reasoning Darwin arrives at the following conclusion. "I conclude that a large continental area, which will probably undergo many oscillations of level, and which consequently will exist for long periods in a broken condition, will be the most favourable for the production of many new forms of life, likely to endure long and spread widely." Darwin argues as follows:

In a large continent, there is severe competition. This leads to the extinction of over-specialized species. But it is highly improbable that something new arises on a large continent. This happens much easier on small islands. But if the islands are isolated for a long time, then over-specialized forms will develop.

So Darwin postulates, that the islands should reconvert to a large continent. There will be again severe competition eliminating the specialized forms. This briefly sketches Darwin's derivation of his hypothesis. The interested reader is highly recommended to read the above mentioned chapter in Darwin's book. I found this conjecture so important that I have named it *Darwin's continent-island cycle* conjecture.

Mainstream biological science seems not to have noticed or to have deliberately neglected this section in Darwin's book. Darwin's evolution model is a non-equilibrium model, whereas all the popular Darwinian, Neo-Darwinian and Synthesis theories are equilibrium models.

Darwin's arguments in favor of the continent cycle can also, with some minor changes, be applied to other areas like the *invention of successful scientific ideas* or the *efficient organization of companies*. Take the organization of companies as an example. If the market (the environment) is stable, a large centralized company with severe internal competition is most effective. If the company has to adapt to a changing market, the large company should be subdivided into small companies which can adapt much faster.

1.4.2 Spatial population structures in evolution theories

Several researchers in biology have tried to investigate the importance of spatial population structures for evolution - without ever referring to Darwin. Space is an important element of the *shifting balance theory* of evolution proposed by Wright (56). He argued that the best way to avoid a species being hung up on a low fitness peak is to have the population broken up into many nearly isolated subpopulations. Wright's theory has three phases (57). *Phase 1* consists of the *differentiation* of innumerable small local populations by more or less random processes that occasionally lead to higher peaks. *Phase 2* is the occupation of higher peaks by *local mass selection*. *Phase 3* is the *diffusion* of these successful subpopulations throughout the species, followed by the appearance of still more successful centers of diffusion at points of contact. Then the whole process starts again.

Fisher (11), in contrast, argued that no such theory is needed. In a highly multidimensional fitness surface, the peaks are not very high and are connected by fairly high ridges, always shifting because of environmental changes. According to Fisher, the analogy is closer to waves and troughs in an ocean than in a static landscape. Alleles are selected because of their average effects, and a population is unlikely to be ever in such a situation that it can never be improved by direct selection based on additive variance.

The difference between these two views is not purely mathematical, but physiological. Does going from one favored combination of alleles to another often necessitate passing through genotypes that are of lower fitness? Fisher argued that evolution typically proceeds in a succession of small steps, leading eventually to large differences by the accumulation of small ones. According to this view, the most effective population is a large panmictic one in which statistical fluctuations are slight and each allele can be fairly tested in combination with many others alleles. According to Wright's view, a more favorable structure is a large population broken up into subgroups, with migration sufficiently restricted (less than one migrant per generation) and size sufficiently small to permit appreciable local differentiation.

Four different models for spatially structured populations have been investigated mathematically

- the one-island model
- the island model
- the stepping stone model
- the isolating by distance model

In the one-island model, an island and a large continent are considered. The large continent continuously sends migrants to the island. In the island model, the population is pictured as subdivided into a series of randomly distributed islands among which migration is random.

In the stepping-stone model migration takes place between neighboring islands only. One and two dimensional models have been investigated. The isolation by distance model treats the case of continuous distribution where effective demes are isolated by virtue of finite home ranges (neighborhoods) of their members. For mathematical convenience it is assumed that the position of a parent at the time it gives birth relative to that of its offspring when the latter reproduces is normally distributed.

Felsenstein (9) has shown that the isolating by distance model leads to unrealistic clumping of individuals. He concluded, that this model is biologically irrelevant. There have been many attempts to investigate spatial population structures by computer simulations, but they did not have a major influence on theoretical biology. A good survey of the results of the different population models can be found in (10). Population models with oscillation like Darwin's continent-island cycle have not been dealt with.

The issue raised by Wright and Fisher is still not settled. Phase 3 of Wright's theory has been recently investigated by Crow (6). He concludes: "The importance of Wright's shifting-balance theory remains uncertain, but we believe whatever weaknesses it may have, they are not in the third phase."

The problem of spatial population structures is now reappearing in the theory of genetic algorithms. The plain GA is based on Fisher's model. It is a well known fact, that the GA suffers from the problem of premature convergence. In order to solve this problem, many genetic algorithms enforce diversification explicitly, violating the biological metaphor. A popular method is to accept an offspring only if it is genetically more than a certain factor different from all the members of the population.

Our parallel genetic algorithm PGA tries to introduce diversification more naturally by a spatial population structure. Fitness and mating is restricted to neighborhoods. In the PGA we have implemented the *isolation by distance model and the stepping stone model*. The three phases of Wright's theory can actually be observed in the PGA. But the relative importance of the three phases are different than Wright believed. The small populations do not find better peaks by random processes. The biggest changes of the population occur at the time after migration between the subpopulations. Recombinations between immigrants and native individuals occasionally lead to higher peaks which were not found by any of the subpopulations during isolation. This behavior can easily be demonstrated in the application function optimization (see (34) for details). We can therefore state the following observation.

The creative forces of evolution take place at migration and few generations afterwards. Wright's argument that better peaks are found just by chance in small subpopulations is wrong.

In our opinion the most important part of Wright's theory is what Wright postulated as "the appearance of still more successful centers of diffusion at points of contact". The difference of the evolution in a large continent and small isolated islands, has been recently investigated by (43).

We believe that static fitness functions cannot model natural evolution. In a real environment the fitness of an individual depends on the outcome of its interactions with other organisms in the environment. The fitness cannot be specified in advance. Therefore we used for a simulation of complex spatial population structures an evolutionary game.

1.5 The iterated prisoner's dilemma as an evolutionary game

In our artificial ecology the interactions of the individuals are modeled by a game. The fitness of the individual is the sum of the payoffs the individual gets during its lifetime. We have chosen the Iterated Prisoner's Dilemma

(IPD), because it has been investigated from a number of different viewpoints.

The major emphasis of our research is on methodological questions for at least two reasons. First we believe that methodological questions are of utmost importance in a scientific field where it is almost impossible to compare simulation results with actual experiments. Second, a convincing simulation to support or disprove Darwin's continent cycle theory would need a tremendous effort.

Over its 60-year lifespan, the Iterated Prisoner's Dilemma has been one of the most frequently studied phenomena in economics, political science, sociology and psychology (see Axelrod (1) for a survey). The basic prisoner's Dilemma is a two-person game, with each player having a choice of either cooperating (C) or defecting (D). A typical set of payoffs is presented below.

Move	C	D
C	3/3	0/5
D	5/0	1/1

Given these payoffs, it is easily shown that mutual defection is the only Nash equilibrium. Of course, the intrigue of the Prisoner's Dilemma is that this unique equilibrium is Pareto inferior to the mutual cooperation outcome. If the basic Prisoner's Dilemma is iterated, the resulting super game is an Iterated Prisoner's Dilemma IPD. If the number of iterations is a known finite number, then a simple backward induction argument implies that the only equilibrium is mutual defection in every round. However, if the game is repeated a finite, but unknown number of times, then cooperative behavior can theoretically emerge.

The *ecological* approach to experimental games has added another dimension to the study of conflict and cooperation in societies. John Maynard Smith (47) introduced the *evolutionary game theory*, where the games are played by a population of individuals. The higher the payoff of an individual, the more offspring he will get. In this manner the most effective strategies survive. A strategy is called *evolutionary stable* (47), if it cannot be invaded by a single mutant strategy.

The theory assumes that the strategies are not changed during the course of evolution. In our simulations the strategies are coded by genes. The strategies are constantly changed by the parallel genetic algorithm, which uses mutation and crossing-over for generating offspring.

1.5.1 *The simulation of spatial structures using the Iterated Prisoner's Dilemma*

There have been many attempts to investigate the IPD with genetic algorithms. The first simulation was performed by Axelrod (2). Axelrod considered strategies where the moves are based on the game's past three-move history. The major focus of Axelrod's study was on strategies evolving against a fixed environment. Each individual played against eight representative strategies. Marks (31) extended the investigation to *bootstrap* evolution, where the individuals play against each other. Miller (33) used finite automata to represent strategies. Furthermore he investigated the effect of informational accuracy on the outcome of the simulation. All three researchers used the plain genetic algorithm for evolving the population. They have been interested in equilibrium states and "optimal" strategies. We concentrate on the evolution of the behavior of the total population.

The PGA has been extended to simulate different population structures. The major enhancements of the PGA to the plain genetic algorithm are the spatial population structure, the distributed selection and the local hill-climbing. The individuals are active. They look for a partner for mating in their neighborhood. The partner is chosen according to the preference of the individuals. The best individual in a neighborhood has the chance to get as many offspring as the global best individual of the population. The PGA therefore has a very "soft" selection scheme. Each individual has the chance that on average 50% of its genes are contained in the chromosome of an offspring. The offspring replaces the parent. In order not to complicate the simulations the individuals are not allowed to improve their fitness by learning. This means their strategy is fixed during their lifetime.
We now turn to the problem of genetic representation of strategies.

1.5.2 The genetic representation

There are at least two obvious ways to represent strategies as a genetic chromosome, one is based on a simple table lookup, the other on finite automaton. We will discuss in this paper deterministic table lookup strategies. A k-lookback strategy can be defined as a mapping of the outcome of the last k moves into a new move. In the simplest case of just looking one play back, a strategy can be defined by four entries in a table symbolizing the four possible moves of the last game - DD,DC,CD,CC. In addition two bits are necessary to specify the first move. The genetic representation of one-lookback thus consists of six bits. This gives 2^6 different genotypes. Three popular strategies are given below

						strategy
C	*	*	*	C	C	ALL-C
D	*	D	D	*	*	ALL-D
C	*	D	C	D	C	TIT-FOR-TAT

The sign * denotes that the allele on this locus does not have any influence on the performance of the strategy. The ALL-C strategy in row one is defined as follows. The player starts with C, then only two outcomes are possible, CD or CC. In both cases the player plays C. The outcomes DD and DC are not possible, therefore the entries in these columns are irrelevant. Altogether there are twelve different bit-strings which define an ALL-C strategy. The problem of this straightforward genetic representation is that we have a distinction between the *representation* and the *interpretation*. The program which interprets the representation is not part of the genetic specification and therefore not subjected to the evolution process.

But we have a clear distinction between genotype, phenotype and behavior. The genotype is mapped into some phenotype, the phenotype together with the environment (in our case the other phenotypes) defines the strategy. Let us take the famous TIT-FOR-TAT as an example. In TIT-FOR-TAT the player makes the move the opponent made the game before. In an environment where only C is played, TIT-FOR-TAT cannot be distinguished from an ALL-C player. A different behavior can only be recognized if there exists an individual who occasionally plays D.

The mapping from genotype to phenotype is many-to-one. This makes a behavior oriented interpretation of a given genetic representation very difficult. There exist no simple structure of the genotype space. The Hamming distance between two ALL-C genetic representations can be as large as four, whereas the Hamming distance between two very different strategies like ALL-C and ALL-D can be as small as one. An example is shown below

						strategy
C	C	D	D	C	C	ALL-C
C	C	D	D	C	D	ALL-D

If we assume that the genetic operators *mutation* and *crossing-over* uniformly explore the genotype space, then strategies like ALL-C and ALL-D will have a much higher chance to be generated than other strategies which are less often represented. The genetic search is therefore biased by the genetic representation. We believe that this effect is not a shortcoming of the chosen representation, but that this feature models real life evolution. The evolution has always to work within the constraints it creates for itself.

The complex mapping between genotype and phenotype makes it difficult to estimate the outcome of a genetic operator. For example, a winning strategy may be crossed with a losing strategy, giving in most cases a new strategy. An ALL-D strategy which is crossed-over with an ALL-C strategy gives with probability 0.2 ALL-D and with probability 0.2 ALL-C. With probability 0.6 we get a strategy which is different from the strategies of the parents.

We believe that in our artificial ecology the crossover operator is too disruptive compared to real evolution. The same problem occurs if the genetic representation is based on a finite automaton. In order to solve this problem we have to find a genetic representation which is based on a more complex genetic machinery than simple bit-strings. It is outside the scope of this paper to discuss this genetic machinery. We only want to mention that we have to incorporate some ideas of genetic models of self-reproduction proposed already in the 60's.

The influence of spatial population structures is independent of the genetic representation, therefore we will concentrate on this subject.

1.5.3 Mathematical analysis of structured populations in evolutionary games

Before we discuss some of the simulation results in detail we want to show by a simple analysis how a spatial population structure influences the development of strategies. For simplicity we assume that we have a population, consisting of *inhabitants* playing strategy I and *invaders* playing strategy J.

Let $0 < s < 1$ be the proportion of invaders. We assume that s is very small. Furthermore the invaders are clustered. We model this fact by a clustering factor $0 < k \leq 1/s$.

Let $P(I, J)$ denote the payoff of an individual playing strategy I against an individual playing strategy J. After invasion the fitness of the inhabitants $F(I)$ is given by the outcomes of plays with each other and against the invaders. Computing the fraction of each of the games we obtain

$$F(I) = (1 - s * \frac{1 - ks}{1 - s}) * P(I, I) + s * \frac{1 - ks}{1 - s} P(I, J) \tag{1.6}$$

The fitness of the invaders is given by

$$F(J) = (1 - ks) * P(J, I) + ks * P(J, J) \tag{1.7}$$

We see that for $k = 0$ the invaders play against the inhabitants only, the case $k = 1$ gives the panmictic population normally considered in the theory of evolutionary games. Here the plays are performed according to the frequency of the actors. In the case of $k > 1$ we have a clustering effect. The players play more often within their groups (inhabitants, invaders). For $k = 1/s$ the effect is most dramatic. The mixed terms with $P(i, J)$ vanish, thus the invaders and the inhabitants play within their group only. This is a very crude model of a structured population, but it can be used to show some important points.

A strategy is called *collective stable* if no strategy can invade it. A new strategy is said to *invade* if the newcomer gets a higher score than the native strategy, this means that $F(I) < F(J)$. In order to obtain a simple formula, we assume that s is small approximate F(I) by $P(I, I)$. Thus the small number of plays between inhabitants and invaders is ignored. We get

$$P(I, I) < (1 - ks)P(J, I) + ks * P(J, J) \tag{1.8}$$

It is now easily seen that even ALL-C can withstand the invasion of ALL-D, if there is a strong preference for each strategy to play only against each other. With our payoff values we obtain that ALL-C will not be invaded by ALL-D if $k > 0.5s^{-1}$. But also the other invasion is possible. ALL-C can invade an ALL-D population as long as they "stick together". This means they play, even after the invasion, much more against each other than against ALL-D. [4]

In a one-dimensional spatial population structure with fixed neighborhoods the situation is more difficult. The contest between the strategies happens at the boundary of the neighborhoods, whereas the individuals in the interior play only against members of their own group. In this spatial structure the success of the invasion is therefore totally determined by the outcomes at the boundary.

It is almost impossible to investigate realistic spatial population structures by analytical methods, one has to use simulations. This was first done by Axelrod ((1), pp. 158-168). Axelrod investigated a simple 2-D structure where each player had four neighbors. The selection was very strong. If a player had one or more neighbors which had been more successful, the

[4] If we set $p = ks$ then the above inequality is identical to Axelrod's *p-cluster invasion* ((1),p.212).

player converted to the strategy of the most successful of them. Axelrod's major conclusion was that mutual cooperation can be sustained in a (not too highly connected) territorial system at least as easy as it can be in a freely mixing system. We will extend Axelrod's work. First, different population structures are compared and second, the strategies evolve controlled by the genetic algorithm.

1.5.4 Simulation results

In our simulation we have investigated the following population structures

- a small panmictic population
- a large panmictic population(500 individuals)
- a one-dimensional population (ring structure with four neighbors)
- the continent-island cycle: a cycle between ten islands populations and a panmictic population
- a competition between five population structures

In a panmictic population each individual plays against each other, in a spatial population structure the individuals play only against their neighbors. Most of the experiments have been done with a small population of 50 individuals. Detailed simulation results can be found in (3). We outline in this paper only the major facts supporting or disproving Darwin's argument. In our simulations we used 2-lookback strategies. They can be coded by 20 bits. This gives 2^{20} different genotypes.

The figures show individual runs which are "representative". Each individual experiment has been repeated 10 times. The results are qualitative as described. The figures are what we believe "average" runs. Because of the stochastic nature of the individual runs (especially the runs with small populations) it makes no sense to average the fitness over the 10 different runs. A small panmictic population for instance occasionally changes to a non-cooperative behavior for a certain time. The time when this will happen cannot be predicted.

For a qualitative investigation we used the average fitness of the population. Batz (3) has used more difficult measures like the distribution of classes of strategies. Given the pay-off table theoretical research indicates that strategies like TIT-FOR-TAT (TFT) are evolutionary stable, leading to an average fitness of three.

The difference between a large panmictic population and a small one is shown in figures 1.1 and 1.2. The simulation of the large population started with three predefined strategies - 5 ALL-D, 490 ALL-C and 5 TFT. The population heads first to non-cooperative behavior. Then ALL-D is beaten by TFT. The population arrives at cooperation in generation 45.

Similarly the small population started with 5 ALL-D, 40 ALL-C and 5 TFT. The small panmictic population oscillates, but is also heads first to non-cooperative behavior. It arrives at cooperation at generation 65.

The result of this experiment can be explained mathematically. The initial fitness of the strategies can easily be computed. In the small population the fitness of ALL-D is given by

$$F(ALL-D) = (4 + 200 + 5)/49 = 4.27$$

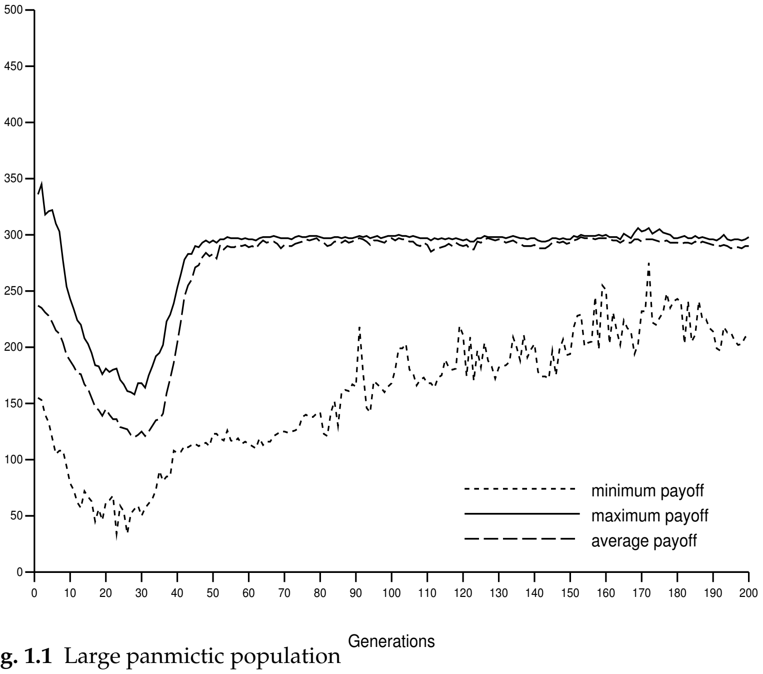

Fig. 1.1 Large panmictic population

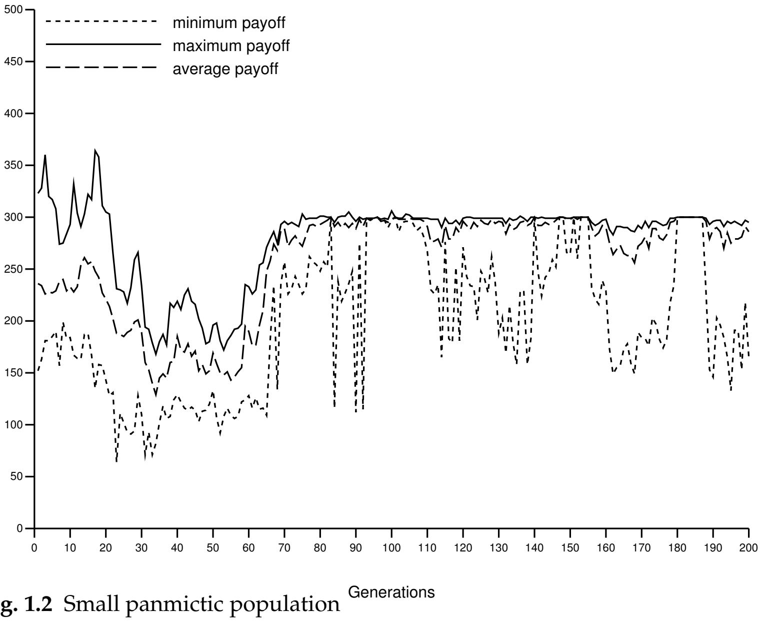

Fig. 1.2 Small panmictic population

Similarly we obtain $F(TFT) = 2.84$ and $F(ALL - C) = 2.69$. Thus ALL-D first increases by eliminating ALL-C. Then ALL-D is conquered by TFT. For the large population we get $F(All - D) \approx 5$, $F(TFT) \approx 3$, $F(ALL - C) \approx 3$. The difference of the fitness between ALL-D and TFT is larger in the large population. This explains the rapid increase of ALL-D in the large population at the beginning.

The ring population oscillates as shown in figure 1.3. The selection scheme of the PGA is too soft for this population structure. In order to implement a higher selection pressure we introduced an acceptance test of the offspring. In the first scheme the offspring replaced the parent only if it won the IPD against the parent. The effect was dramatical. Now the population always settled on non-cooperative behavior. The situation changed with our second scheme. This extension we called the *family game*. Each mating produces two offspring. After the mating the family consisting of the two parents and the two offspring plays an IPD tournament. The winner replaces the parent. With this selection scheme the population settled on cooperative behavior.

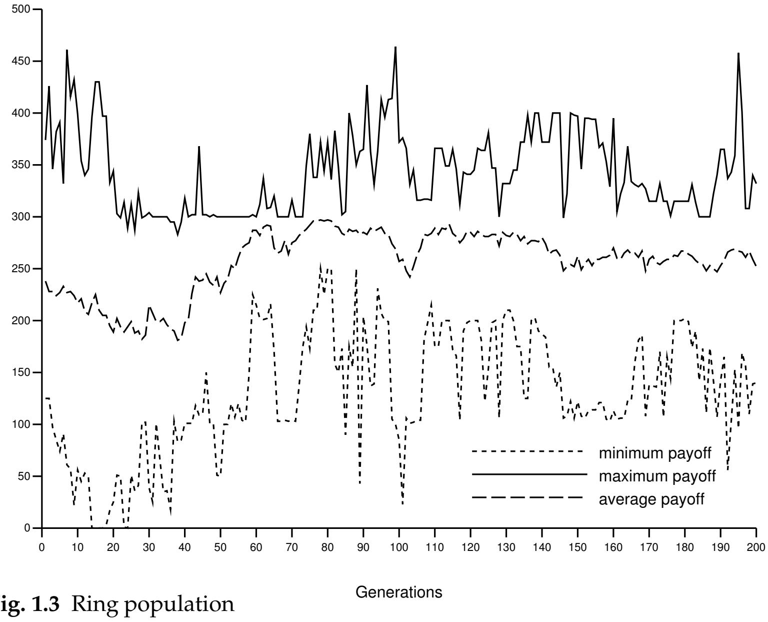

Fig. 1.3 Ring population

The explanation of this result is simple. In the IPD non-cooperative strategies can be eliminated if the cooperative individuals stick together. In a single contest, ALL-D can never be beaten. It is outside the scope of this paper to compare the family game with *kin selection* proposed in sociobiology (55).

In figure 1.4 the continent-island cycle is shown. One easily recognizes the cycle (20 generations island, 20 generations continent). During the continent phase the variance is reduced, during the island phase it is increased.

In figure 1.5 the average fitness of the population is shown for five different population structures. The simulation started with a homogeneous

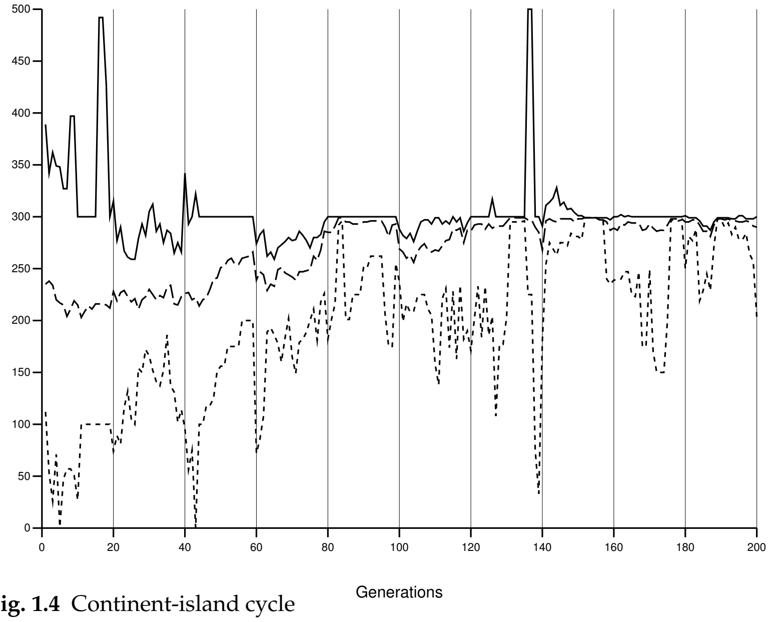

Fig. 1.4 Continent-island cycle

ALL-D population. We investigated whether the populations will change to cooperation. We see that the population which is subjected to the continent-island cycle is first to arrive at cooperation. This result was consistent in ten runs. A closer analysis of the strategies showed that the winning cooperative strategies are not naive like ALL-C, but they resemble TIT-FOR-TAT.

In a further set of experiments we changed the game during the course of the simulation, for instance we changed the IPD to the chicken game. The spatial structured populations adapted much faster to the new game than a large panmictic population. This is one of the extensions that have been already proposed by Axelrod for investigation ((1), p.221).

1.5.5 The punctuated equilibrium theory

In our simulation, we observe rapid changes in fitness, followed by long intervals of small fluctuations. This offers some support for a view of evolution called *punctuated equilibrium* (16). This concept was controversial in evolutionary biology for some time, but it has now many supporters. Let us describe the theory and its historical context.

During the years of Darwin's apprenticeship in science there has been an intense conflict in geological circles between adherents of *rapid* and *gradual* changes of the fossil record. There is no doubt that Darwin strictly favored gradualism in his writing.

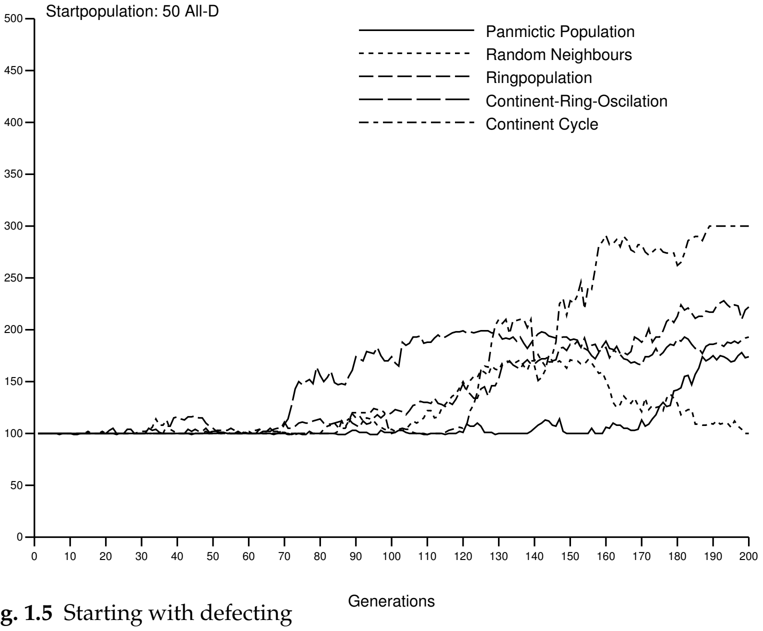

Fig. 1.5 Starting with defecting

This posed a dilemma to paleontologist. In order to favor gradual changes by natural selection, they had to view the empirical data as so incomplete that the process of natural selection cannot be observed.

Gould and Eldredge abandoned this view by rejecting gradualism. They observed in the history of most fossil species two features particularly inconsistent with gradualism:

- **Stasis:** Most species exhibit no directional change during their tenure on earth. They appear in the fossil record looking much the same as when they disappear; morphological change is usually limited and direction less.
- **Sudden appearance:** In any local area, a species does not arise gradually by the steady transformation of its ancestors; it appears all at once and 'fully formed'.

This behavior they called punctuated equilibrium. It explains the fossil record. The evolutionary forces responsible for this behavior are still controversial discussed. In all simulations the time of changes (indicated by a change in average fitness) is small compared to the time of the equilibrium. We next turn to a different application of the PGA, the solution of difficult combinatorial problems.

1.6 Combinatorial optimization by the PGA

Most applications of genetic algorithms are optimization problems. Here a static fitness function is used derived from the optimization problem. The search strategy of the PGA is driven by three components - the spatial population structure, the crossover operator and the hill-climbing strategies. The spatial structure has been discussed in the previous section. In this section we concentrate on the problem dependent aspect, the crossover operator and the local hill-climbing.

There have been attempts to "prove" that genetic algorithms make a nearly optimal allocation of trials. This result is called the "Fundamental Theorem of Genetic Algorithms" (Goldberg (13)) We have shown already in (34) that the above claim is only valid for simple optimization problems. In fact, in (37) we have proven a correct schema theorem, based on Boltzmann selection and our Estimation of Distribution family of algorithms (29; 36).

The search strategy of a genetic algorithm can be explained in simple terms. The crossover operator defines a *scatter search* (12) where new points are drawn out of the area which is defined by the old or "parent" points. The more similar the old parent are, the smaller will be the sampling area. Thus crossing-over implements an adaptive step-size control.

But crossing-over is also exploring the search space. Let us assume that the combinatorial problem has the *building block feature*. We speak of a building block feature if the substrings of the optimal solutions are contained in other good solutions. In this case it seems a good strategy to generate new solutions by patching together substrings of the old solutions. This is exactly what the crossover operator does.

1.6.1 *The Traveling Salesman Problem*

The major difficulty for applying the PGA to combinatorial problems is to define a crossover operator which creates valid solutions i.e. solutions which fulfill the constraints of the problem. We will explain this problem first with the TSP.

THE TRAVELING SALESMAN PROBLEM

OPT 1 (TSP) *Given are n cities. The task of the salesman is to visit all cities once so that the overall tour length is minimal.*

This problem has been investigated in (35; 14; 15; 34) with the PGA. The genetic representation is straightforward. The gene at locus i of the chromosome codes the edge (or link) which leaves city i. With this coding, the genes are not independent from each other. Each edge may appear on the chromosome only once, otherwise the chromosome would code an invalid tour. A simple crossing-over will also give an invalid tour. This is the reason why this simple genetic representation has not been used in genetic algorithms. The researchers tried to find a more tricky representation in order to apply a simple crossover operator.

We take the opposite approach. We use a simple representation, but an intelligent crossover operator. Our crossover operator for the TSP is straightforward. It inserts part of chromosome A into the corresponding location at chromosome B, so that the resulting chromosome is the most similar to A and B. A genetic repair operator then creates a valid tour.

We call our crossover operator MPX, the maximal preservative crossover operator. It preserves sub-tours contained in the two parents. The pseudocode is given below.

```
PROC    crossover (receiver, donor, offspring)
    Choose position $0 <= i < nodes$ and length $b_{low} <= k <= b_{up}$ randomly.
    Extract the string of edges from position $i$ to position $j = (i + k)$ MOD $nodes$ from the
    mate (donor). This is the crossover string.
    Copy the crossover string to the offspring.
    Add successively further edges until the offspring represents a valid tour.
    This is done in the following way:
        IF an edge from the receiver parent starting at the last city in the offspring is possible
        (does not violate a valid tour)
        THEN    add this edge from the receiver
        ELSE    IF an edge from the donor starting at the last city in the offspring is possible
        THEN    add this edge from the donor
        ELSE    add that city from the receiver which comes next in the string, this adds a
        new edge, which we will mark as an implicit mutation.
```

We want to recall, that in the PGA the crossover operator is not applied to all TSP configurations, but only to configurations which are a local minima. Our local search is a fast version of the 2-opt heuristic developed by Lin (28). It is a 2-opt without checkout. It gives worse solutions than 2-opt, but the solution time scales only linearly with the number of cities.

We have later found that the efficiency of the PGA increases with the quality of the local search. But the major goal of the PGA work on the TSP was to investigate the problem independent aspects i.e. the population structure and the selection schedule. Therefore many generations were needed, which could only be obtained by a fast local search method.

We turn to a popular benchmark problem, the ATT-532 problem solved to optimality in (42). The PGA with a population size of 64 and truncated 2-opt as local search method got a tour length of 0.10% above optimal in ten runs of $t = 1080s$ (1000 generations,15000 local searches) on a 64 processor system, the average final tour length was 0.19% above optimal (14). This is a substantial improvement over the results in (50) for genetic 2-opt search. It demonstrates the robustness of the parallel genetic algorithm. The PGA finds good solutions with a simple local search also.

This implementation had some influence in the development of heuristics for the TSP. There is a section about the different PGA implementation in Johnson and McGeoch's seminal paper (26).

We will compare our heuristic with a very fast and efficient heuristic proposed by Johnson (24). It is called *iterated Lin-Kernighan* search. In his implementation a new start configuration is obtained by an unbiased 4-opt move of the tour at hand. Then a new L-K search is started. If the search leads to a tour with a smaller tour length, the new tour is accepted.

Johnson reports the following results. In time $t = 2700s$ (500 L-K searches) the optimal tour (length 27686) was output 6 of 20 IterL-K runs, the average final tour-length was 0.05% above optimal. Multiple L-K runs gave much worse results. A single L-K run averages 0.98% above optimal in time $t = 120s$. 100 L-K runs gave a tour length of 0.25% above optimal. It needed 20000 L-K runs ($t = 530hours$) to obtain a tour of length 27705.

Why is IterL-K more efficient than independent L-K runs? The success of IterL-K depends on the fact that good local L-K minima are clustered together and not randomly scattered. The probability to find a good tour is higher nearby a good tour than nearby a bad tour.

We have shown in (34) that 2-opt local minima are clustered. Furthermore we could show the following relation: The better the solutions are, the more similar they are. This relation is the reason for the success of Johnson's IterL-K. The relation holds, if the problem has the *building block feature*, which is necessary for the success of the crossover operator of our genetic algorithm.

Iterated hill-climbing needs a fine tuned mutation rate to get with high probability to the attractor region of a new local minimum. In the TSP case Johnson found that a simple 4-opt move is sufficient. In other combinatorial problems it is more difficult to find a good mutation rate and a good local heuristic like the Lin-Kernighan search for the TSP. Therefore we share the opinion of Johnson that the TSP is in practice much less formidable than its reputation would suggest (24). An in-depth evaluation of heuristics for the solution of large TSP problems can be found in (26).

1.6.2 The Graph Partitioning Problem

We will now turn to another combinatorial problem, the graph partitioning problem GPP. Here the Lin-Kernighan heuristic is not as good as in the TSP case. We will show that the genetic search is very efficient for this problem. The major obstacle is to find a suitable crossover operator.

THE GRAPH PARTITIONING PROBLEM

The m graph partitioning problem (m-GPP) is a fundamental problem which arises in many applications. The GPP is to divide a given graph into a number of partitions (m) in order to optimize some criterion e.g. to minimize the number of edges between partitions. More formally:
Let a graph $G = (V, E, w)$ be given. $V = \{v_1, v_2, ..., v_n\}$ is the set of nodes, $E \subseteq V \times V$ is the set of edges and $w : E \mapsto I\!N$ defines the weights of the edges.
The m-GPP is to divide the graph into m disjunct parts, such that some optimization criteria will be fulfilled. In this paper we will consider the following optimization criteria:

OPT 2 (m-GPP) *Let $\mathcal{P} = \{P_1, ..., P_m\}$ be a partition. Let $\mathcal{G} = (g_1 g_2 ... g_n)$ denote the partition to which the nodes belong $(1 \leq g_i \leq m)$. Then we look for*

$$\min_{\mathcal{P}} \sum_{\substack{1 \leq i < j \leq n \\ g_i \neq g_j}} w_{ij}$$

such that $\sigma(P)$ is minimal.

$\sigma(P)$ is defined as

$$\sigma^2(P) = \frac{1}{m}\sum_{i=1}^{m}|P_i|^2 - (\frac{1}{m}\sum_{i=1}^{m}|P_i|)^2$$

In order to solve the GPP, we have to define the genetic representation and the genetic operators. In the simplest representation, the value (allele) g_i on locus i on the chromosome gives the number of the partition to which node v_i belongs. But this representation is highly degenerate. The number of a partition does not have any meaning for the partitioning problem. An exchange of two partition numbers will still give the same partition. All together m! chromosomes give the same fitness value.

$$F(\mathcal{G}) = \sum_{\substack{1 \le i < j \le n \\ g_i \neq g_j}} w_{ij}$$

All m! chromosomes code the same partitioning instance, the same "phenotype". The genetic representation does not capture the structure of the problem. We did not find a better genetic representation, so we decided that the crossover operator has to be "intelligent". Our crossover operator inserts complete partitions from one chromosome into the other, not individual nodes. It computes which partitions are the most similar and exchanges these partitions. Mathematically spoken, the crossover operator works on equivalence classes of chromosomes.

Figure 1.6 shows an example. The problem is to partition the 4×4 grid into four partitions.

The crossover operator works as follows. Partition 2 has to be inserted into $\mathcal{B}$. The crossover operator finds, that partition 4 of $\mathcal{B}$ is the most similar to partition 2 in $\mathcal{A}$. It identifies partition 2 of $\mathcal{A}$ with partition 4 of $\mathcal{B}$. Then it exchanges the alleles 2 and 4 in chromosome $\mathcal{B}$ to avoid the problems arising from symmetrical solutions. In the crossover step it implants partition 2 of chromosome $\mathcal{A}$ into $\mathcal{B}$.

After identifying all gen-loci and alleles which lead to a non-valid partition a repair operator is used to construct a new valid chromosome. Mutation is done after the crossover and depends on the outcome of the crossover. In the last step a local hill-climbing algorithm is applied to the valid chromosome.

For local hill-climbing we can use *any* popular sequential heuristic. It should be fast, so that the PGA can produce many generations. In order to solve very large problems, it should be of order $O(n)$ where n is the problem size. Our hill-climbing algorithm is of order $O(n^2)$, but with a small constant. In order to achieve this small constant, a graph reduction is made. The general outline of our hill-climbing algorithm is as follows:

Local search for the GPP

1. Reduce the size of the graph by combining up to r nodes into one hyper-node
2. Apply the 2-opt of Kernighan and Lin to the reduced Graph. For the GPP it is defined as follows:

   a. Select two hyper-nodes

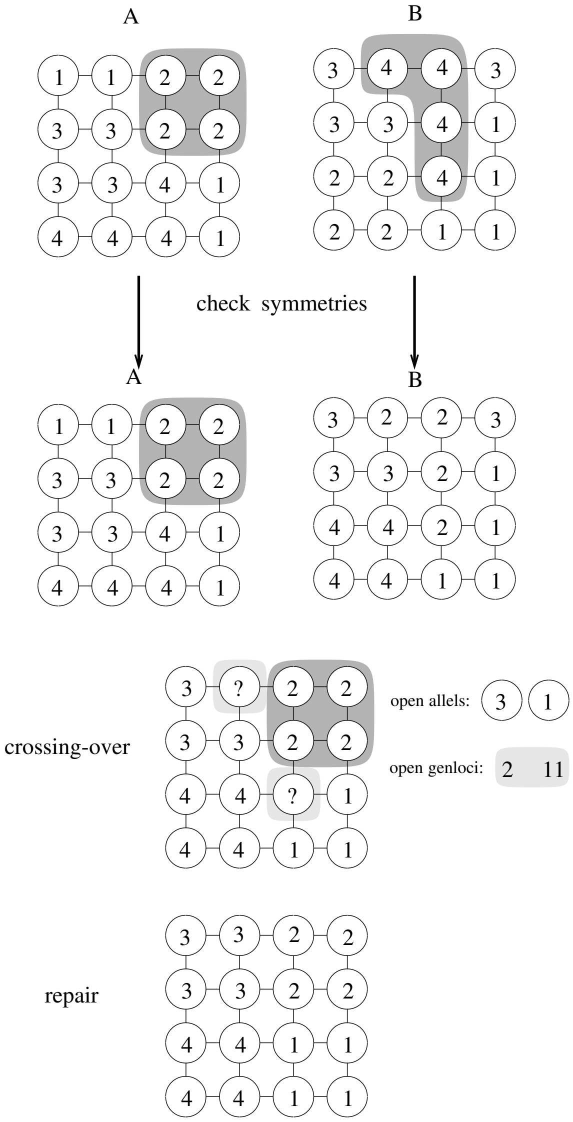

Fig. 1.6 The crossover operator for the m-GPP

b. Test if an exchange of this hyper-nodes gives a lower fitness
c. If this is the case, exchange the nodes
d. Terminate, if no exchange can be made over the set of nodes

3. Expand the resulting graph
4. Create a valid partition
5. Apply a further local hill-climbing algorithm to the valid partition

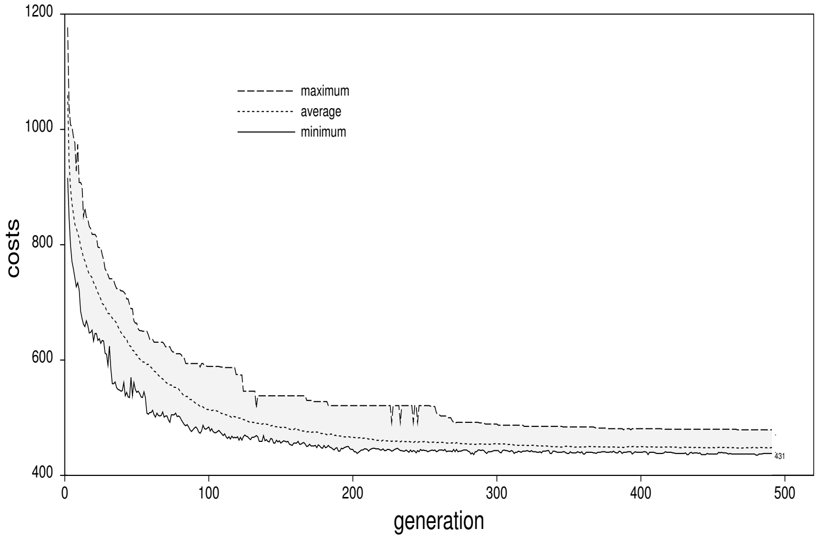

Fig. 1.7 Problem EVER918, 64 individuals

Step2 and step5 are of order $O(n^2)$. The constant is smaller than doing 2-opt search on the original string. The above local search is only done for the initial population. After the first generation we use a still faster search. We apply 2-opt only to the nodes which have connections to outside partitions.

The general m-GPP problem has been studied seldom. More popular is the the bi-partitioning problem. A detailed study of the bi-partitioning problem can be found in (25) In that paper random graphs and random geometric graphs up to 1000 nodes are used to compare different heuristics. We decided to make a performance analysis with real life graphs. Furthermore are we more interested in the general partitioning problem, not in the bi-partitioning problem. Detailed results can be found in (53). Results for the bi-partitioning problem using our new Factorized Distribution Algorithm FDA can be found in (37).

We will give here the computational results for solving a large GPP benchmark problems. The problem is called *EVER918*. It is a 3-D graph which consists of 918 nodes and 3233 edges. It has to be partitioned into 18 partitions.

In figure 1.7 we show the progress of the PGA for problem $EVER918$. The progress is typical. The best progress is made at the beginning, then it decays exponentially.

For EVER918 the PGA found the best solution computed so far (54). Further investigations have indicated that it will be difficult to construct an efficient iterative Lin-Kernighan search for the m-GPP. First, the quality of an average L-K solution is bad for the GPP. Second, it is difficult to determine a good mutation rate which jumps out of the attractor region of the local minimum. This has been demonstrated in (54). There the following relation was shown: the better the local minimum, the larger its attractor region.

In summary: The m-GPP problem is more difficult to solve than the TSP. The PGA got results which are better than other known heuristics.

1.7 Continuous function optimization by competition

Optimization of multi-modal continuous functions was a notoriously difficult task for genetic algorithms. Sometimes good results have been achieved, but for many benchmark problems genetic algorithms have not been competitive with other methods. Therefore we developed an algorithm called the *Breeder Genetic Algorithm* BGA. The BGA uses a real-valued representation (38; 39). Therefore a number of specialized mutation and recombination operators have been implemented. These depend on parameters. In order to automaticly adapt the parameters, *competition* between subpopulations using different parameters has been implemented. Detailed discussions of the competition scheme can be found in (39; 46).

1.7.1 The BGA for Continuous Parameter Optimization

Let an unconstrained optimization problem be given on a domain $D \subset \Re^n$

$$\min(F(\mathbf{x})) \quad a_i \leq x_i \leq b_i \quad i = 1, ..., n \ . \tag{1.9}$$

The breeder genetic algorithm **BGA** was designed to solve the above problem (38). The BGA uses a set of search strategies. An overview of search strategies based on recombination can be found in (52) and (51). In (52) it was shown that a new recombination scheme called **fuzzy recombination** (FR) worked best as a breadth search. In this chapter we only describe the BGA mutation scheme and the BGA line recombination which uses also the mutation scheme.

BGA mutation $\mathsf{BM}(\rho, k, \nu)$

The BGA mutation scheme depends on the *mutation range* ρ, the *precision* k and a new parameter ν which gives the number of neighboring variables to be mutated. The standard BGA mutation ($\nu = 0$) randomly selects just one variable x_i.

Given x_i a new value z_i is computed according to

$$z_i = x_i + \rho_i \cdot \delta(k) \tag{1.10}$$

ρ_i is normally set to the domain of definition of variable x_i. $\delta(k)$ is a random variable which is computed as follows:

$$\delta(k) = \text{sign}(\alpha) \cdot 2^{-k \cdot |\alpha|} \quad \alpha = \mathbf{U}(-1, 1)$$

where $\mathbf{U}(u,v)$ denotes the uniform probability distribution with support $(v,w) \subset \Re$.

k is called the precision constant. The smallest absolute value of $\delta(k)$ is 2^{-k}, the highest value is 1.0. Therefore the step sizes of the BGA-mutation are contained in the interval $[\rho_i \cdot 2^{-k}; \rho_i]$.

The rationale of the BGA mutation scheme has been explained in (38). An extension of the BGA-mutation is specified by its third parameter ν. Now additionally to the randomly chosen variable x_i also adjacent variables are modified. The higher the distance to i the smaller is the change.

$$z_{i-j} = x_{i-j} + \rho_{i-j} \cdot 2^{-j} \tag{1.11}$$

$$z_{i+j} = x_{i+j} + \rho_{i+j} \cdot 2^{-j} \tag{1.12}$$

$$\begin{array}{l} for \;\; j = 1, \ldots \nu \\ and \; i - j > 0, \quad i + j \leq 0 \end{array}$$

where Δ is the value of $\delta(k)$ generated for x_i

The parameter ν defines the size of the neighborhood. It lies in the interval $[0; n]$. For the standard BGA-mutation we have $\nu = 0$. Note that the mutation step decreases exponentially starting from variable x_i. This is in accordance to the design rationale of the BGA mutation.

BGA line recombination $\mathsf{BLR}(\rho, k)$

The BGA line recombination uses components from both, mutation and recombination. It creates new points in a direction given by the two parent points. The placement of the point is done by the BGA mutation scheme. It works as follows: Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ be the parent strings with $\mathbf{x}$ being the one with better fitness. Then the offspring $\mathbf{z} = (z_1, \ldots, z_n)$ is computed by

$$z_i = x_i + \rho_i \cdot \delta(k) \cdot \frac{y_i - x_i}{\|\mathbf{x} - \mathbf{y}\|} \tag{1.13}$$

The BGA line recombination may generate points which are far from the given point x_i.

1.7.2 *Competition between Subpopulations*

In order to automatically adapt the parameters, we had the idea to use subpopulations each with a different set of parameters. The subpopulations with high fitness should increase, the ones with low fitness should decrease. Before implementing a scheme we looked if theoretical models could guide us.

Ecology deals with the interaction of subpopulations and species. Unfortunately even the analysis of the interaction between two species can be quite complicated, involving the effects of exterior and interior parameters.

As a first approximation one may distinguish four basic situations — competition, predator-prey, symbiosis and host-parasite.

The most popular equations for analyzing the interaction of species are generalizations of the famous Lotka-Volterra equations. For two species they are as follows (8)

$$\frac{dN_1}{dt} = r_1 \cdot N_1 \left(1 - \frac{N_1}{K_1} - \alpha_{12}\frac{N_1}{K_2}\right) \tag{1.14}$$

$$\frac{dN_2}{dt} = r_2 \cdot N_2 \left(1 - \frac{N_2}{K_1} - \alpha_{21}\frac{N_1}{K_2}\right) \tag{1.15}$$

Here N_1, N_2 denote the population sizes of the two species, r_1, r_2 are the growth rates, K_1, K_2 the carrying capacities and $\alpha_{12}, \alpha 21$ the interaction coefficient. This equation has been studied intensively (8). It is very useful for understanding the complex patterns which may arise by two interacting species. For a competition scheme to be implemented these equations cannot be used because the interaction coefficients cannot be specified in advance. In analogy to the above model the following model has been implemented.

1.7.3 The Basic Competition Model of the BGA

Our competition scheme requires a *quality criterion* to rate a group, a *gain criterion* to reward or punish the groups, an *evaluation interval*, and a *migration interval*. The evaluation interval gives each strategy the chance to demonstrate its performance in a certain time window. By occasional migration of the best individuals groups which performed badly are given a better chance for the next competition. The sizes of the groups have a lower limit. Therefore no strategy is lost. The number of competing subpopulations (S) depends on the set of strategies used in the competition. Normally the number of groups is between 2 and 8.

Our **quality criterion** (Q) is based on the fitness of the best individual of the group. To avoid an inefficient oscillation of group sizes we use information about the last ω competitions for the evaluation. The vector $\mathbf{w} \in N^{\omega}$ provides the winners of the last ω competitions. $w_k \in \{1, \ldots, S\}$ contains the winner of the k-last competition.

The following formula describes the quality of group i. $k = 0$ denotes the current competition, $k = 1$ the previous one, etc. The time window ω is 10.

$$Q_i(\mathbf{w}) = \sum_{k=0}^{\omega-1} \begin{cases} (\omega - k)/\omega & : \quad i = w_k \\ 0 & : \quad i \neq w_k \end{cases} \tag{1.16}$$

The **gain criterion** (G) defines how the population size of each group is modified according to its quality. Normally, the size of the group with the best quality increases, the sizes of all other groups decrease. The following scheme increases the size of the best group (w_0) by the accumulated loss of the others. The loss of a group is proportional to its population size. The loss factor $\kappa \in [0; 1]$ defines the rate of loss.

The change of the population sizes is computed from the following equations:

$$\Delta N_i = \begin{cases} \sum\limits_{j=1, j \neq i}^{S} N_j^t \cdot \kappa & : \begin{array}{l} Q_i(\mathbf{w}) > Q_j(\mathbf{w}) \\ \forall j, j \neq i \end{array} \\ -N_i^t \cdot \kappa & : \ else \end{cases} \tag{1.17}$$

where N_i^t denotes the size of group i and S denotes the number of groups. The loss factor κ is normally set to 0.125.

The population size of each group of the next generation is given by:

$$N_i^{t+1} = \begin{cases} N_i^t + \Delta N_i & : \ N_i^t + \Delta N_i \geq N^{min} \\ N_i^{min} & : \ else \end{cases} \tag{1.18}$$

The size of the population is only reduced if it is greater than the minimal size N^{min}.

This gain criterion leads to a fast adaptation of the group sizes. Each group looses the same percentage of individuals.

The **evaluation interval** η and the **migration interval** θ are rather robust external parameters. Normally we set $\eta = 4$ and $\theta = 16$.

If one compares equations 1.17 with the generalized Lotka-Volterra equation 1.14 the following major difference can be observed. Our equations are linear whereas the Lotka-Volterra equations contain the nonlinear term $N_i \cdot N_j$. The reason for this difference is that the Lotka-Volterra equations model individual competition. If there are many predators and each one captures two preys, then the reduction of the preys depends on the number of predators. In contrast, our competition scheme evaluates whole groups by taking the best individual as evaluation criterion.

The current competition scheme seems appropriate in cases when the strategies used by the different groups differ substantially. Sometimes a competition model might be better where even the size of the total population may vary.

1.7.4 The Extended Competition Model

If search strategies differ very much they may also require a different population size to be efficient. It has been shown in (38) that mutation is most efficient in small populations whereas recombination needs a larger population size. This can be modeled by introducing growth rates which depend on the group.

In our implementation we introduced a *consumption factor* γ for each subpopulation. Biologically spoken a consumption factor specifies the consumption of the limited resource by one individual of a species — the higher the consumption factors the lower the number of individuals which can be supported by that resource. We implemented this extension by introducing a *normalized population size* $\tilde{N}$.

$$\tilde{N}_i = \gamma_i \cdot N_i \tag{1.19}$$

The gain criterion of equation 1.17 is now applied to the normalized population sizes. The sum of the normalized population sizes remains constant because it is limited by the limited resource K.

$$\sum_{i=1}^{S} \tilde{N}_i = K \tag{1.20}$$

For $\gamma_i = 1.0$ for $i = 1, \ldots, S$ we obtain the basic model.In contrast to the basic model the sum of the real population sizes varies during a simulation. This extended competition scheme is very useful for multi-modal problems where it is useful to locate the region of attraction of the global (or a good local) optimum by a breadth search and to do the fine adaptation by an exploring strategy afterwards. In this case the strategy performing breadth search gets a lower γ than the other strategy. So the total population size is high at the beginning when the breadth search works and low at the end when the fine adaption is done. So the whole population size is adapted during the run by the competition model.

Numerical results for difficult test functions can be found in (46). A discussion about he evolution of the population sizes during a run can be found in (45).

1.8 Conclusion

Complex spatial population structures are seldom used in evolutionary computation. In this chapter we have investigated the stepping-stone model, competing sub-populations, and Darwin's continent-island cycle. For Darwin's conjecture an evolutionary algorithm was used where the fitness of each individual is given by the competition with other individuals. The competition is modeled by evolutionary games. The parallel genetic algorithm PGA uses the stepping-stone interaction. It runs totally in parallel. The selection is distributed and done by each individual in its neighborhood. Faster convergence can be obtained by the Breeder Genetic Algorithm BGA. It models breeding as it is done by a human breeder. For really difficult optimization problems the competing BGA has been developed. It uses competing sub-populations which are bred using different strategies. Occasionally good individuals migrate to other sub-populations. The sizes of the sub-populations are adjusted according to their performance.

Darwin's cycle model seems also a good starting point for investigating the development of new ideas in human societies, be it in science or art. It takes small groups or even a single individual to try out new ideas. But for the ideas to be accepted a large community is needed. In a large community many individuals evaluate the new ideas, only the most promising eventually survive.

References

[1] R. Axelrod. *The evolution of cooperation*. Basic, New York, 1984.

[2] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In L. Davis, editor, *Genetic algorithms and Simulated Annealing*, pages 32–41. Morgan Kaufmann, Los Altos, 1987.

[3] M. Batz. Evolution von Strategien des Iterierten Gefangenen Dilemma. Master's thesis, Universität Bonn, 1991.

[4] L.L. Cavalli-Sforza and M.W. Feldman. *Cultural Transmission and Evolution: A Quantitative Approach*. Princeton University Press, Princeton, 1981.

[5] J.P. Cohoon, S.U. Hedge, W.N. Martin, and D. Richards. Punctuated equilibria: A parallel genetic algorithm. In J.J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 148–154. Lawrence Erlbaum, 1987.

[6] J.F. Crow, W.R. Engels, and C. Denniston. Phase three of wright's shifting balance theory. *Evolution*, 44:233–247, 1990.

[7] R. Dawkins. *The Selfish Gene: Second Edition*. Oxford University Press, Oxford, 1989.

[8] J.M. Emlen. *Population Biology : The Coevolution of Population Dynamics and Behavior*. Macmillan Publishing Company, New York, 1984.

[9] J. Felsenstein. A pain in the torus: Some difficulties with models of isolation by distance. *Amer. Natur.*, 109:359–368, 1975.

[10] J. Felsenstein. The theoretical population genetics of variable selection and migration. *Ann. Rev. Genet.*, 10:253–280, 1976.

[11] R.A. Fisher. *The Genetical Theory of Natural Selection*. Dover, New York, 1958.

[12] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8:156–166, 1977.

[13] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, 1989.

[14] M. Gorges-Schleuter. Asparagos: An asynchronous parallel genetic optimization strategy. In H. Schaffer, editor, *3rd Int. Conf. on Genetic Algorithms*, pages 422–427. Morgan-Kaufmann, 1989.

[15] M. Gorges-Schleuter. *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. PhD thesis, University of Dortmund, 1991.

[16] S.J. Gould and N. Eldredge. Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, 3:115–151, 1977.

[17] P. E. Griffiths. The philosophy of molecular and developmental biology. In *Blackwell Guide to Philosophy of Science*. Blackwell Publishers, 2002.

[18] W.S.C. Gurney and R.M.Nisbet. *Ecological Dynamics*. Oxford University Press, New York, 1998.

[19] E. Haeckel. Über die Entwicklungstheorie Darwin's. In *Gemeinverständliche Vorträge und Abhandlungen aus dem Gebiet der Entwicklungslehre*. Emil Strauss, Bonn, 1902.

[20] W. D. Hamilton. The genetical evolution of social behavior I and II. *Journal of Theoretical Biology*, 7:1–16, 17–52, 1964.

[21] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, 1998.

[22] J.H. Holland. *Adaptation in Natural and Artificial Systems*. Univ. of Michigan Press, Ann Arbor, 1975/1992.

[23] E. Jablonka and M.J. Lamb. *Evolution in Four Dimensions*. MIT Press, Cambridge, 2005.

[24] D.S. Johnson. Local optimization and the traveling salesman problem. In M.S. Paterson, editor, *Automata, Languages and Programming*, Lecture Notes in Computer Science 496, pages 446–461. Springer Verlag, 1990.

[25] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part i, graph partitioning. *Operations Research*, 37:865–892, 1989.

[26] D.S. Johnson and L.A. McGeoch. The traveling salesman problem: a case study. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. Wiley, Chichester, 1997.

[27] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 2:291–307, 1970.

[28] S. Lin. Computer solutions of the traveling salesman problem. *Bell. Syst. Techn. Journ.*, 44:2245–2269, 1965.

[29] Th. Mahnig and H. Mühlenbein. A new adaptive Boltzmann selection schedule SDS. In *Proceedings of the 2001 Congress on Evolutionary Computation*, pages 183–190. IEEE Press, 2001.

[30] B. Manderick and P. Spiessens. Fine-grained parallel genetic algorithm. In H. Schaffer, editor, *3rd Int. Conf. on Genetic Algorithms*, pages 428–433. Morgan-Kaufmann, 1989.

[31] R. E. Marks. Breeding hybrid strategies: Optimal behavior for oligopolist. In H. Schaffer, editor, *3rd Int. Conf. on Genetic Algorithms*, pages 198–207, San Mateo, 1989. Morgan Kaufmann.

[32] H. R. Maturana and F. J. Varela. *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel, Boston, 1980.

[33] J. K. Miller. The coevolution of automata in the repeated prisoner's dilemma. Technical report, Santa Fe Institute, 1989.

[34] H. Mühlenbein. Evolution in time and space - the parallel genetic algorithm. In G. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 316–337. Morgan Kaufmann, San Mateo, 1991.

[35] H. Mühlenbein, M. Gorges-Schleuter, and O. Krämer. Evolution algorithms in combinatorial optimization. *Parallel Computing*, 7:65–88, 1988.

[36] H. Mühlenbein and R. Höns. The factorized distribution algorithm and the minimum relative entropy proncieple. In M. Pelikan, K. Sastry, and E. Cantu-Paz, editors, *Scalable Optimization via Probabilistic Modeling*, pages 11–37. Springer, New York, 2006.

[37] H. Mühlenbein and Th. Mahnig. Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, 31(3):157–192, 2002.

[38] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive Models for the Breeder Genetic Algorithm I. Continuous Parameter Optimization. *Evolutionary Computation*, 1:25–49, 1993.

[39] H. Mühlenbein and D. Schlierkamp-Voosen. The science of breeding and its application to the breeder genetic algorithm. *Evolutionary Computation*, 1:335–360, 1994.

[40] H. Mühlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17:619–632, 1991.

[41] S. Oyama. *Evolutions's Eye*. Duke University Press, Durham, 2000.

[42] W. Padberg and G. Rinaldi. Optimization of a 532-city symmetric traveling saleman problem by branch and cut. *Op. Res. Let.*, 6:1–7, 1987.

[43] D. Parisi and M. Ugolini. Living in enclaves. *Complexity*, 7:21–27, 2002.

[44] A. Rapaport. Modern systems theory – an outlook for coping with change. *General Systems*, XV:15–25, 1970.

[45] D. Schlierkamp-Voosen and H. Mühlenbein. Strategy adaptation by competing subpopulations. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel Problem Solving from Nature*, Lecture Notes in Computer Science 866, pages 199–208. Springer-Verlag, 1994.

[46] D. Schlierkamp-Voosen and H. Mühlenbein. Adaptation of population sizes by competing subpopulations. In *Proceedings IEEE Conference on Evolutionary Computation*, pages 330–335, New York, 1996. IEEE Press.

[47] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, Cambridge, 1982.

[48] J. Maynard Smith and E. Szathmary. *The Major Transitions in Evolution*. W.H. Freeman, Oxford, 1995.

[49] R. Tanese. Distributed genetic algorithm. In H. Schaffer, editor, *3rd Int. Conf. on Genetic Algorithms*, pages 434–440. Morgan-Kaufmann, 1989.

[50] N.L.J. Ulder, E. Pesch, P.J.M. van Laarhoven, H.-J. Bandelt, and E.H.L. Aarts. Improving tsp exchange heuristics by population genetics. In R. Maenner and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature*, pages 109–116. Springer-Verlag, 1991.

[51] H.-M. Voigt and H. Mühlenbein. Gene Pool Recombination and the Utilization of Covariances for the Breeder Genetic Algorithm. In Z. Michalewicz, editor, *Proc. of the 2nd IEEE International Conference on Evolutionary Computation*, pages 172–177. IEEE Press, New York, 1995.

[52] H.-M. Voigt, H. Mühlenbein, and D. Cvetković. Fuzzy recombination for the continuous breeder genetic algorithm. In L. J. Eshelman, editor, *Proc. of the Sixth Int. Conf. on Genetic Algorithms*, pages 104–112. Morgan Kaufmann, San Francisco, 1995.

[53] G. von Laszewski. Ein paralleler genetischer Algorithmus für das Graph Partitionierungsproblem. Master's thesis, Universität Bonn, 1990.

[54] G. von Laszewski. Intelligent structural operators for the k-way graph partitioning problem. In R. K. Belew and L. Booker, editors, *Procedings of the Fourth International Conference on Genetic Algorithms*, pages 45–52, San Mateo, 1991. Morgan Kaufmann.

[55] D. S. Wilson and L. A. Dugatkin. Nepotism vs tit-for-tat, or, why should you be nice to your rotten brother. *Evol. Ecology*, 5:291–299, 1991.

[56] S. Wright. The distribution of gene frequencies in populations. *Proc. Nat. Acad. Sci*, 24:253–259, 1937.

[57] S. Wright. Factor interaction and linkage in evolution. *Proc. Roy. Soc. Lond. B*, 162:80–104, 1965.