

# The NQK Model of Fitness Dynamics: Adaptation by Selective Elimination and Random Replacements

Franco di Primio\*

Fraunhofer Institute for Autonomous Intelligent Systems (AIS),  
53754 St. Augustin, Germany

*Random numbers are arranged in a circle. At each time step, the lowest number, and the numbers at its two neighbors, are each replaced by new random numbers. That's all! This step is repeated again and again. What could be simpler than replacing some random numbers with some other random numbers? Who says that complexity cannot be simple? This simple scheme leads to rich behavior beyond what we could imagine. (Per Bak, 1996, p. 138 [1].)*

---

An abstract model of fitness evolution is presented which works on the basis of a form of selection called ranking based elimination (RBE).  $N$  refers to the size of a population and  $Q$  defines the quota of less fit individuals that are replaced by new ones in each generation. The  $K$  parameter makes it possible to take into account dependencies that may lead to the replacement of individuals regardless of their fitness. Unlike the evolution model initially studied by Sneppen *et al.* in [2], our focus here is not on the asymptotic dynamics of critical fitness values, but rather on how the described parameters affect the average fitness dynamics in a short-range perspective, when there is a limit on the length of the evolutionary time range in question (i.e., the number of generations that can be calculated). The main results are that for every maximum number of generations there is a  $Q$  value which yields, on average, a maximum for the average fitness and that even low values of  $K$  (1 and 2) have a disruptive effect on the average fitness.  $Q$  and  $K$  may be used as externally tunable control parameters, but they also can be made to depend on the fitness value of individuals, and thus become internal parameters of the model. An analytical model for the dynamics is given, as well, to complement the experimental results. We also discuss possible applications and argue that, in contrast to  $Q$ ,  $K$  is not scale- and domain-independent.

---

## 1. A fundamental evolutionary algorithm

---

The evolution of natural populations of organisms is an immensely complex, multidimensional, and dynamic process. Genetic algorithms used

---

\*Electronic mail address: franco.diprimio@ais.fraunhofer.de.

in an attempt to imitate evolution are models with a rather low number of dimensions. Their properties are analyzed with respect to a few operators such as various selection schemes, different representations of individuals (binary or not, of fixed or variable length), different syntactic manipulators of such representations (particular forms of mutation and recombination), different degrees of interaction between (more or less spatially distributed) individuals of the same or different populations, and different application domains (e.g., [3] or an introductory textbook like [4]). What we are talking about here is a very simple evolutionary algorithm which is minimal in the sense that it works without specific reference to operators other than selection. The representation of individuals is completely abstracted away and we only focus on their (total) fitness value. In the context of genetic algorithms research one needs to exercise caution in the use of biological concepts (for a criticism of “simple” models of evolution see [5]). Because of the extreme level of abstraction in our model we prefer the term “agent” over “individual” to designate the “objects” of selection. Biologists still disagree about the units of selection: for most of them, it is individual organisms that undergo selection, for some (e.g., R. Dawkins in [6]) it is genes, and for still others it is species or even ecosystems that are selected. Thus, our choice of terminology makes sense, and may even be an advantage because it stresses the potential “universality” of the model allowing consideration of its meaning with respect to different levels or scales of evolution. We have consciously applied the galilean method, eliminating in computer-based gedanken experiments factors that could never be eliminated in reality. Needless to say, we are aware that in our abstract notions of fitness and agents we have eliminated considerably more factors than just “air resistance and friction.”

Our research concerns the dynamics of the average fitness of populations where successive generations evolve in accordance with a form of selection we call ranking based elimination (RBE), which operates on the basis of two parameters,  $Q$  and  $K$ , to be explained later.

Experiments are carried out, using a uniform random number generator as an agent (or fitness) constructor. Choosing a random fitness value, which we assume to be in the interval  $[0, 1]$ , is equivalent to generating a new individual or agent. Indeed, because no other structure or information for agents is given, a population can be simply represented as a vector of numbers. The length of the vector is the number of agents, the number at position  $i$  in the vector is the fitness of agent  $i$  in the population.

In the following we have chosen an incremental presentation of our results for the sake of clarity. We start by discussing the  $Q$  parameter and introduce the  $K$  parameter later.

The first question is: Given a random initial population of  $N$  agents, how does the average fitness of the population evolve from one genera-

tion to the next, as a result of the effects of the following update rule?

The population at time  $t + 1$  is obtained from the population at time  $t$  replacing a certain number of less fit agents (a given part or quota  $Q$  of the population,  $Q \leq N$ ) with new random agents.

(See the appendix for more formal definitions of the RBE procedure given in pseudocode.)

The basic idea is, in other words, to sort a population from worst to best and to replace the first few least fit agents  $Q$  with new randomly generated ones.

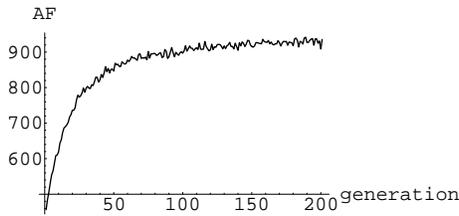
Consider a population consisting of 10 agents. Let its initial representation be the vector: (.45, .65, .89, .79, .93, .24, .56, .22, .08, .67), where the fitness values are rounded to two places for the sake of simplicity. To find the least fit individuals, the vector is sorted in ascending order: (.08, .22, .24, .45, .56, .65, .67, .79, .89, .93). Now, if  $Q$  (the portion of individuals to be replaced) is, for instance, 2 then the first two individuals (with respective fitness values .08 and .22) are replaced by two new agents, that is, two random numbers in  $[0, 1]$ , to get the next population. The update procedure is then repeated.

We carried out systematic simulation experiments with various population sizes, varying  $Q$  values, and different limits to the number of generations. The parameter  $K$ , which makes it possible for us to take into account interdependencies between agents and will be discussed later, is related to the work of S. Kauffman [7] on NK-fitness landscapes. The intention there was to model the complexity of (epistatic) coupling effects between traits or genes ( $N$  being their number) within one organism. The idea to consider only fitness values for the representation of individuals was inspired by the work of Bak in [1]. Having decided not to keep track of the underlying “genetic code,” he thought of agents primarily as species and not as single organisms, and the investigation of what we call parameters  $Q$  and  $K$  was limited to a few specific values and their asymptotic dynamics. We based the name of our model, NQK, on this previous work. In our context, a population consists of agents of any sort, without reference to their specific type (e.g., genes, organisms, species, etc.).

## 2. Effects of the $Q$ parameter

For the sake of greater clarity, we start by examining a more comprehensive, concrete example. We take a population of size  $N = 100$  and choose as the quota  $Q = 6$ . A run of the RBE-procedure for 200 generations is shown in Figure 1.

The  $y$  coordinate is the average fitness expressed as a part of 1000; thus 700 corresponds, for instance, to 0.7. The result of the procedure is a vector equal in length to the number of generations, containing

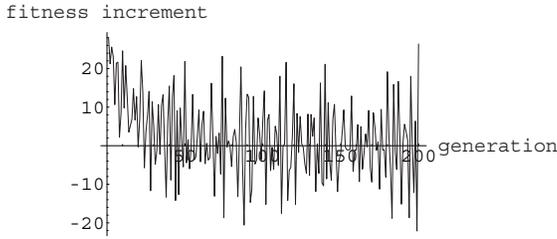


**Figure 1.** Curve of average fitness: population size  $N = 100$  with  $Q = 6$  over 200 generations.

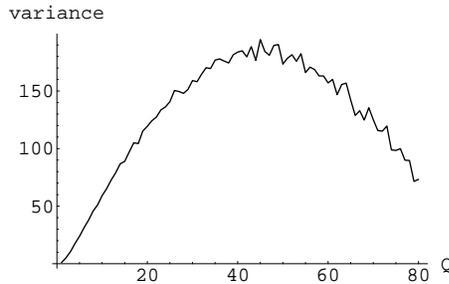
in each position the average fitness reached by the population in that generation. The average of the result vector (i.e., of average fitness values) can be used for rough comparisons between different runs of the procedure. The main focus, however, is on the increments in average fitness from generation to generation. This characterizes our approach. In [1] Bak was primarily interested in the so-called critical or threshold value, that is, the highest level of low fitness reached by the entire population. A procedure is defined to keep track of the lowest fitness value of the population for every generation, and if this happens to be higher than the lowest fitness value of the previous generation, the threshold is set to the new value. While the fitness of an agent can fall below the threshold, the threshold itself can only go up. Bak studied the distribution of “avalanche” sizes and their movement in the population (or ecosystem). An avalanche is defined as a series of fitness changes starting when the fitness of an agent falls below the threshold value. It is over when the lowest fitness agent rises above the (old) threshold. Our intention is to study the behavior of average fitness (not lowest fitness), and how fast certain (averages of) fitness averages are reached.

In the example shown in Figure 1 (notice that the curve looks continuous, but it is not: the dynamic is step-like), average fitness increases rapidly at first (up to generation 50), then slowly, until it finally stabilizes at a mean value of about 0.9. The rate of improvement appears to slow exponentially. Taking a closer look at the relative increments in average fitness from one generation to the next (Figure 2), one sees that they are positive for the first few generations, and that they then tend to fluctuate between positive and negative values.

The fluctuations clearly depend on the fact that the higher the average fitness is, the lower the probability is of increasing it by replacing agents. The variance of the fluctuations also depends on the quota. In the example shown in Figure 1, the variance is approximately 110. If  $Q$  is doubled to 12, the variance becomes approximately 220. The variance (which obviously also depends on the size of the population)



**Figure 2.** Increments of average fitness for the run shown in Figure 1.



**Figure 3.** Variance values for increments of average fitness for  $Q$  from 1 to 80.

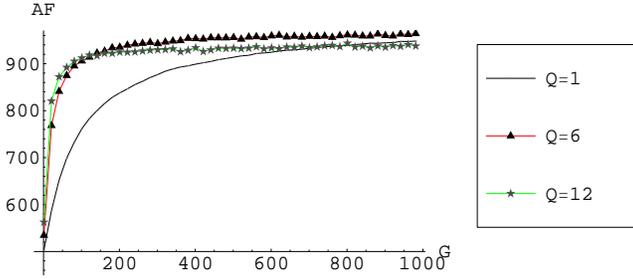
reaches a maximum for values of  $Q$  around 50, then starts decreasing again. Figure 3 shows values of variance (averaged over 100 runs) for population size  $N = 100$ , 200 generations, and  $Q$  values from 1 to 80.

In general, relative increments of average fitness from one generation to the next (starting at generation 0) are positive more often at low values of  $Q$  than at high ones. Otherwise they tend to oscillate between positive and negative values.

The main result can be formulated in qualitative terms as follows. The parameter  $Q$  (when  $K = 0$ , meaning the absence of interdependencies) is the only parameter that determines the dynamics of average fitness.

1. For every maximum number of generations there is a  $Q$  value which yields, on average, a maximum for average fitness.
2. The higher the number of generations, the lower  $Q$  must be in order to get a comparatively high level of average fitness.

Concerning the first point, the two extreme cases are relatively easy to understand. Given, for instance, a population of size  $N = 100$ , its average fitness in generation 0 will be approximately 0.5, given the random makeup of the initial population. If, for each of the next generations, only one agent (the least fit) is replaced by a new random agent ( $Q = 1$ ), the average fitness will at first increase rapidly until



**Figure 4.** Curves of average fitness for  $Q = 1, 6,$  and  $12$  over 1000 generations.

about 50% of the agents have a fitness value below 0.5 and then more slowly. Given enough time, it will reach an arbitrary value near the maximum but not stay there. In the other extreme case, if all 100 agents are replaced in each step ( $Q = 100$ ), the average fitness will, in all probability, stay the same at approximately 0.5.

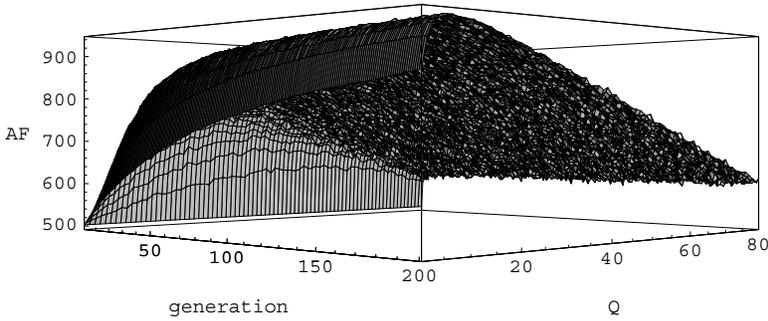
The asymptotic effect of  $Q$  on average fitness can be calculated analytically. When the number of generations can grow without limit, average fitness ( $f_{\max}$ ) will, on average, be:

$$f_{\max} = 1 - \frac{0.5Q}{N}. \quad (1)$$

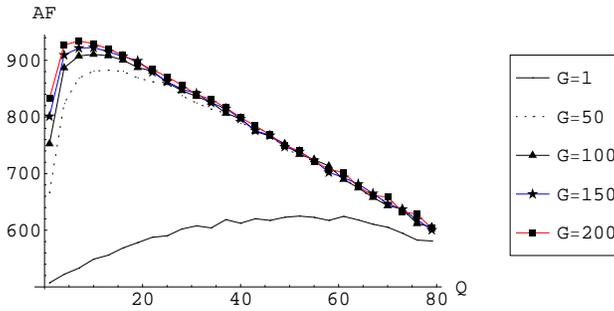
The formula can be derived as follows. In the (very) long run  $N - Q$  agents will reach the maximum fitness value 1. In each generation the remaining  $Q$  will contribute an average quantity of  $0.5Q$  to the total fitness. Average fitness will therefore be  $(1(N - Q) + 0.5Q)/N$ , which can be reduced to equation (1). With this formula, a value of  $Q = 1$  yields the best asymptotic result. A value of  $Q = 0$  means that there is no evolutionary activity.

The interesting cases, however, are the intermediate ones, that is, if it is assumed that the number of generations, or the time for producing them, is limited. For the same population and a maximum of 250 generations, the best average fitness is reached, for instance, if six least fit individuals are replaced in each update step. Thus, the parameter  $Q$  affects the speed at which a certain value for average fitness is reached. To illustrate this, Figure 4 shows the curves of average fitness for population size  $N = 100$ , 1000 generations, and different  $Q$  values of 1, 6, and 12. The data is the average of 20 runs. Clearly, if the time (the number of generations) is limited, for instance to 250, then  $Q = 6$  will give the best result.  $Q = 12$  is better when fewer generations are involved, and  $Q = 1$  is even better than  $Q = 6$ , if more than 8000 generations (not shown in Figure 4) are calculated.

Note that if the population size and the  $Q$  parameter are changed in the same proportion, the average fitness dynamics stay the same. This



**Figure 5.** Three-dimensional plot of the dynamics of average fitness for a population of size  $N = 100$ , for values of  $Q$  from 1 to 80, for generations 0 to 200.

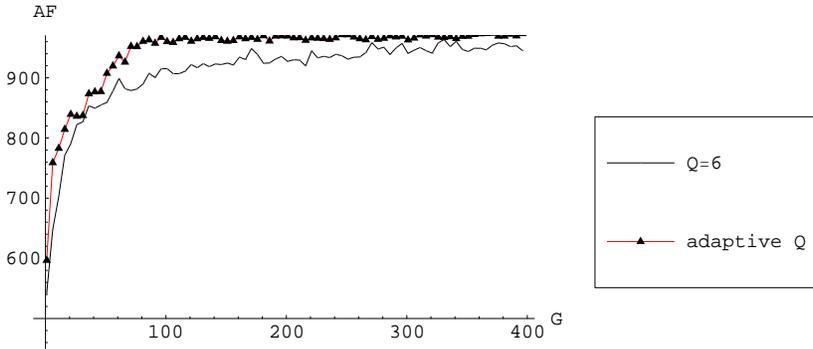


**Figure 6.** Maximum average fitness for  $Q$  from 1 to 80 but only for generations 1, 50, 100, 150, and 200.

obviously means that if  $Q$  is not or cannot be changed, then changing  $N$  can serve as a fitness control parameter.

Figure 5 is a comprehensive three-dimensional plot of the dynamics of average fitness ( $z$ -axis) for a population of 100 agents, for values of  $Q$  from 1 to 80 ( $y$ -axis), for generations 0 to 200 ( $x$ -axis). The average fitness for a given  $Q$  and a given maximum number of generations is averaged over 20 different runs with the same parameters. The data is viewed from the right-hand corner, which allows us to see the difference between the values of average fitness for low generation numbers and different  $Q$  values.

To stress the second point, Figure 6 shows the average fitness for  $Q$  from 1 to 80, but only for generations 1, 50, 100, 150, and 200. The tendency for average fitness to become larger with smaller values of  $Q$ , depending on the generation number, is clear. Equally evident is the tendency of average fitness values to become quite similar for



**Figure 7.** Curves of the average fitness for  $Q = 6$  and  $N = 100$  over 400 generations and for the adaptive  $Q$  starting with 40, and then decreasing by 1 every second generation.

increasing  $Q$  values and generations, which means that high  $Q$  values do not produce significantly different rates of average fitness increase.

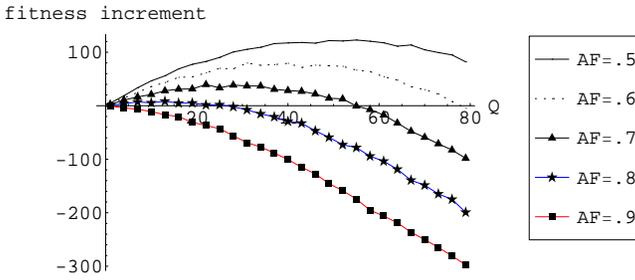
### ■ 2.1 Adapting the $Q$ parameter

Given that different  $Q$  values allow us to obtain different average fitness values at different times, it seems obvious to try to have the parameter change from generation to generation in a manner that will achieve generally better results. Figure 7 shows the curve of an “adaptive” run. Population size  $N = 100$  and the number of generations is 400. The plain curve serves as a reference (with  $Q$  fixed at 6). The other curve (triangles) is obtained by starting with  $Q = 40$  and decreasing  $Q$  by one every second generation, until it remains fixed at 1 in generation 80.

As can be seen, this strategy results in a better performance for every generation. To obtain an optimum adaptation strategy, however, it would be necessary to know what value of  $Q$ —for a given population length and average fitness—yields the best increment of fitness gain. It is possible to derive this information analytically (section 4). Our experiments confirm that there are significant differences. Figure 8 shows for a population of size  $N = 100$  how the increment of fitness varies for values of  $Q$  from 1 to 80 for five different average fitness values (.5, .6, .7, .8, and .9). The increments ( $y$ -axis) are in parts of 1000. The data is the average of 50 runs. Each curve has a maximum that could serve as a parameter value for an adaptive control rule.

### ■ 2.2 Making the $Q$ parameter internal to the model

A changing  $Q$  parameter is not unnatural. The size of biological populations is, for instance, not fixed, and the rate at which unfit organisms



**Figure 8.** Variation of fitness increment (in parts of 1000) for  $Q$  from 1 to 80, when average fitness of a population of size 100 is .5, .6, .7, .8, and .9.

are eliminated is constantly changing. In this sense the  $Q$  parameter in our model is an expression of the probability for agents to be eliminated. The RBE procedure can be changed to incorporate the parameter as an endogenous quantity that is calculated for each generation depending on the elimination probability for every single agent.

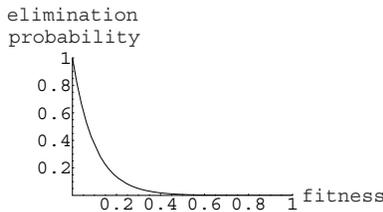
This probability  $P_e$  can be plausibly expressed as an exponential of the fitness value of an agent:

$$P_e(f) = e^{-fc} \tag{2}$$

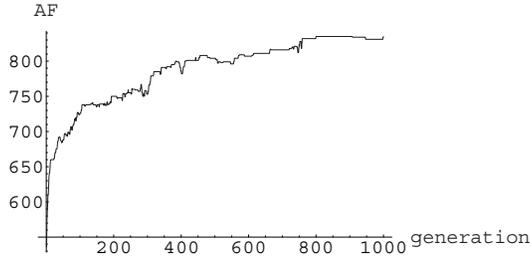
where  $f$  is the fitness value (varying between 0 and 1) and  $c$  is a constant stretching factor allowing us to get a probability near 0 for fitness values near 1. The graph of such a probability function is shown in Figure 9.

The result of an experiment for population size  $N = 100$ , 1000 generations, and an adaptive  $Q$  parameter calculated with  $c = 10$  is shown in Figure 10.

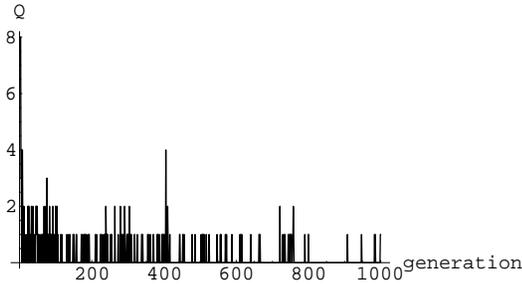
Figure 11 shows the evolution of the  $Q$  parameter over the generations in question. After an initial value of 8 it drops considerably, and after generation 50 it varies between 0, 1, and 2 most of the time, where 0 is obviously the absence of evolutionary activity.



**Figure 9.** Graph of elimination probability based on equation (2) with  $c = 10$ , and  $f$  varying from 0 to 1.



**Figure 10.** Curve of average fitness for population size  $N = 100$ , 1000 generations, and an endogenous adaptive  $\underline{Q}$  based on equation (2) with stretching factor  $c = 10$ .



**Figure 11.** Evolution of  $\underline{Q}$  for the experiment shown in Figure 10.

Other definitions of the elimination probability function are possible. One based on the sigmoid function seems to be better with regard to the average of the fitness averages of all generations. For the sigmoid

$$S(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

a stretching factor is also needed. A fitness value that varies between 0 and 1 must first be mapped into the interval  $[-c, c]$ . The probability is then the complement of the result of the sigmoid  $S$  applied to the mapping result. Given the map function  $M$

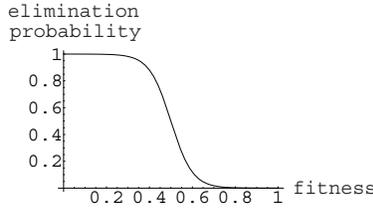
$$M(x, c) = c(2x - 1) \quad (4)$$

the new probability function  $P_s$  is:

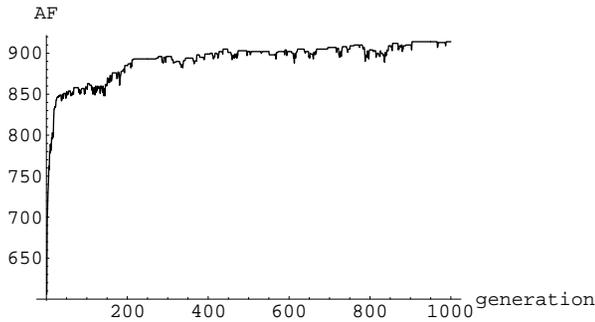
$$P_s(f) = 1 - S(M(f, c)). \quad (5)$$

The graph of a probability function of this kind with  $c = 10$  is shown in Figure 12.

The result of an experiment for population size  $N = 100$ , 1000 generations, and an adaptive  $\underline{Q}$  parameter calculated with  $c = 10$  is



**Figure 12.** Graph of equation (5) with  $c = 10$ , and  $f$  varying from 0 to 1.

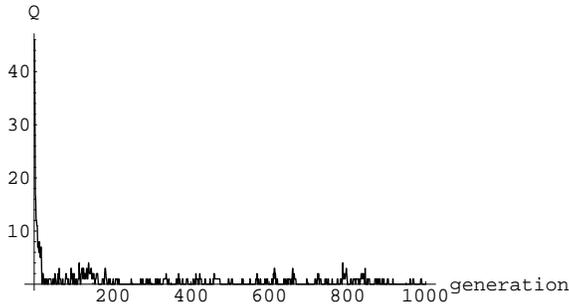


**Figure 13.** Curve of average fitness for population size  $N = 100$ , 1000 generations, and an endogenous adaptive  $Q$  (the sigmoid-based probability function of equation (5)) with stretching factor  $c = 10$ .

shown in Figure 13. The average of the fitness averages is 0.89. With the probability function in Figure 9 it was 0.79.

Figure 14 shows the corresponding evolution of the  $Q$  parameter over the generations. After an initial value of 46 it drops considerably, and after generation 100 it varies between 0, 1, and 2.

Note that unlike the case in common genetic algorithms (GA), where the probability of an individual being copied into the next generation (i.e., not being eliminated) normally depends on the ratio of its fitness to the fitness of the total population (sum of all individual fitness values), in our model there is no need for such a global value. Finally, consider the difference with regard to the generation of new agents. In GA recombination (crossover) plays the major role, and mutation a rather minor one. In our model, mutation (the rate of which depends on elimination probability) is the only innovation factor. However, mutation probability is not the probability of a mutation at a given position in the representation of an individual (e.g., a bit position in a bit string), as in GA. In our model, the whole representation (i.e., every bit position) could be changed, meaning that every agent could be replaced by a completely different one.



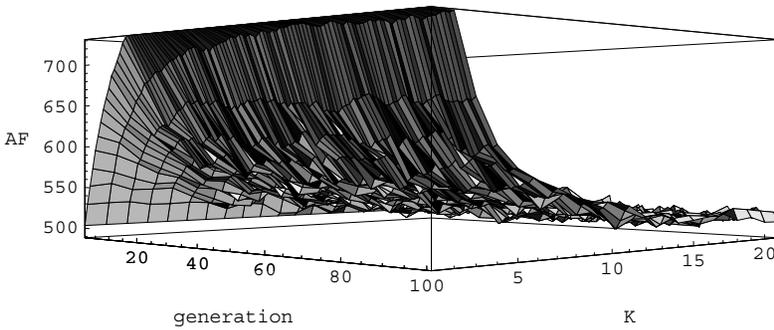
**Figure 14.** Evolution of the endogenous adaptive  $Q$  for the experiment shown in Figure 13.

### 3. Effects of the $K$ parameter

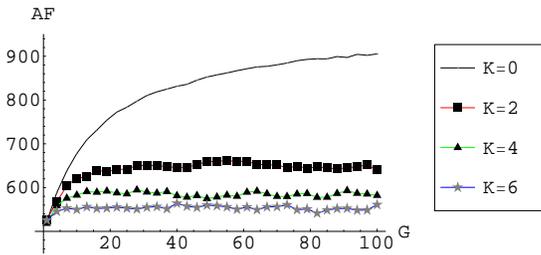
We now introduce the second parameter  $K$  that will allow us to take into account possible dependencies between agents.  $K$  is an integer varying from 0 to  $N - 1$ , with  $N$  being the size of the population, indicating the number of agents an agent is connected to. The idea is that when an agent  $i$  is eliminated,  $K$  other agents, chosen at random and therefore not necessarily among the least  $Q$  fit ones, are also eliminated and replaced by new random agents. This can be interpreted in the sense that the fitness of one agent influences the fitness of other agents, producing a subcritical chain reaction of fitness changes.  $K$  could be different from agent to agent. In the next experiments we choose a unique  $K$  which is valid globally, that is, for all agents, and is to be considered an average length of the chain reaction. The global  $K$  parameter can be, as an average, any real number  $\geq 0$ , provided that the quantity expressing the total number of random replacements, which always equals  $Q \cdot K$ , is rounded off. This can be a sensible choice, for instance, when different values of  $Q$  are used in the same run of the RBE procedure, as is the case when the  $Q$  parameter is made internal to the model (section 2.2). In all the previous examples the parameter  $K$  was tacitly assumed to be 0. Note that in the Kauffman model  $K$  defines the number of genes that affect the fitness contribution of another gene, and thus, as it were, the “inputs” of that gene. In our model the converse applies:  $K$  represents the number of agents an agent is able to affect, that is, its “outputs.”

Figure 15 shows, for a population of size  $N = 100$  and a fixed  $Q$  parameter of 6, how average fitness varies in the course of 100 generations for  $K$  from 0 to 20.

An increase in  $K$  results in a strong decrease in average fitness, as illustrated by the curves shown in Figure 16.



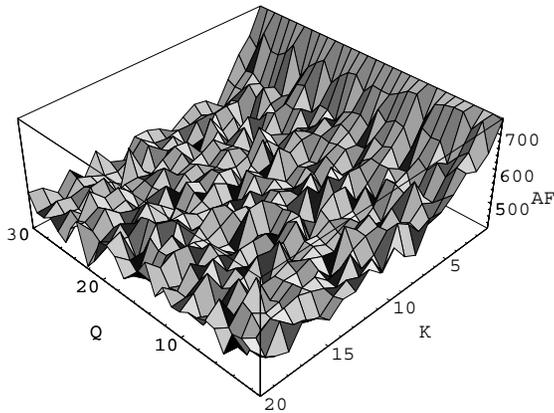
**Figure 15.** Curves of average fitness for population of size  $N = 100$ , 100 generations, fixed  $Q = 6$ , for  $K$  from 0 to 20.



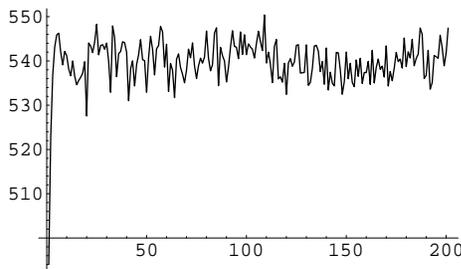
**Figure 16.** Curves of average fitness for  $K = 0$ ,  $K = 2$ ,  $K = 4$ , and  $K = 6$ .  $N = 100$ ,  $Q = 6$ , over 100 generations.

The striking thing is that with respect to  $K = 0$ , even for a value of  $K = 2$  there is a considerable drop of average fitness after a few generations. Figure 17 (viewed from above) shows the average fitness landscape for values of  $Q$  from 1 to 30, and  $K$  from 0 to 19. The population size  $N$  was set at 100. The average fitness given for generation 100 is the result of a single run.

The ruggedness of the average fitness “landscape” for increasing values of  $K$  is evident. Increasing values of  $Q$  are not decisive in this respect, that is, average fitness is low and subject to fluctuations for high values of  $K$ , even if the values of  $Q$  are low. These results are hardly surprising because even a low value of  $K$  (multiplied by the value of  $Q$ ) might result in a high number of replacements for a given population. For instance, if the population size  $N = 100$  and  $Q = 20$ , setting  $K = 5$  may mean that more than 60 individuals will be replaced in each generation randomly, that is, without regard to their fitness level (section 4). A run with parameters of this kind (Figure 18) leads, as expected, to a curve of very low average fitness oscillating around the value of 0.54.



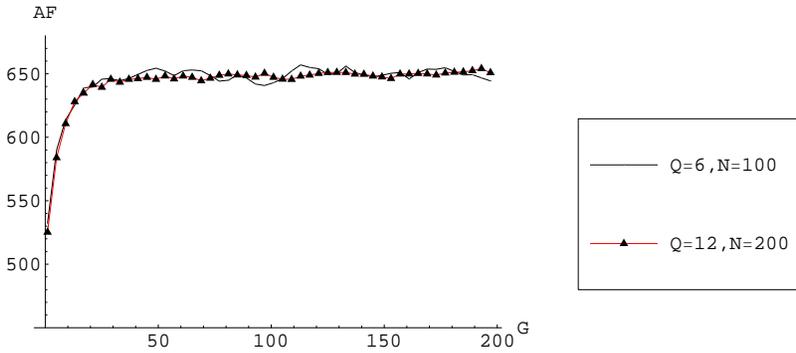
**Figure 17.** Curves of average fitness for values of  $Q$  from 1 to 30, and  $K$  from 0 to 19.



**Figure 18.** Average fitness for  $N = 100$ ,  $Q = 20$ , and  $K = 5$ .

The effect of the  $K$  parameter does not depend on the population size in the following sense. If, for a given  $K$ , the  $Q$  and  $N$  parameters are, for instance, doubled, then the proportion of random replacements per generation ( $KQ/N$ ) does not change and the curve of average fitness remains (approximately) the same. Figure 19 shows the curves for 200 generations where the plain curve plots the average fitness for a population of 100 agents (with  $Q = 6$  and  $K = 2$ ) and the triangles plot the average fitness for a population of 200 agents (with  $Q = 12$  and  $K = 2$ ). The data is the average of 50 runs for each population size, and they almost overlap.

A dependency of  $K$  on  $N$  would be given, if  $N$  varied independently of  $Q$ , but we prefer to take the other point of view, with  $N$  being fixed and  $Q$  variable. Clearly,  $Q/N$ , that is, the ratio of population change at each iteration, is important and not the absolute values of  $Q$  and  $N$ . From our perspective the absolute value of  $N$  determines how many different types of dynamics can be investigated. The larger the  $N$ , the



**Figure 19.** Curves of average fitness for population size  $N = 100$  and  $Q = 6$ ,  $K = 2$ , and population size  $N = 200$  and  $Q = 12$ ,  $K = 2$ .

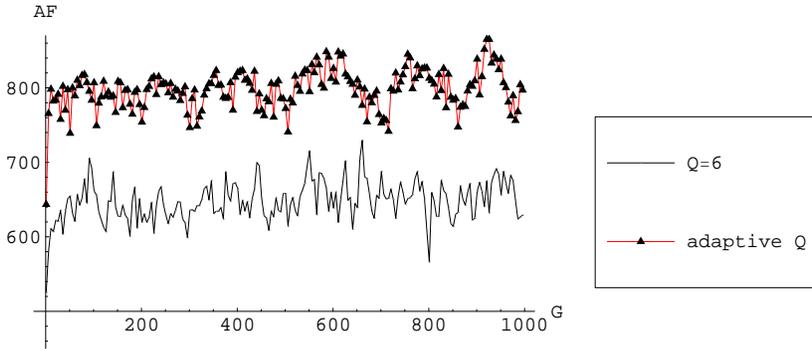
greater the number of  $Q$  values that can be studied.

Note finally that  $K$  can be interpreted as a form of systematic noise parameter, that is, as a regular and not casual form of eliminating agents without regard to their fitness, which would be the case for random influences due to “catastrophes” or other external events. While a major (and hopefully rare) catastrophe may, in individual instances, force down the average fitness of a population,  $K$  does so continuously, depending, as we have shown, to a small extent on  $Q$ .

### ■ 3.1 Adapting the usage of the $K$ parameter

Even a small value of  $K$  has a disruptive effect on the average fitness of a population after just a few generations. Thus, interactions between agents are detrimental in this model. Is there a way to interpret the parameter differently and modify its use in order to gain benefit from these interactions? This is, indeed, possible. Consider the following. If the selection strategy in RBE were, for some reason, inverted, (i.e., in the first  $Q$  the most and not the least fit agents were eliminated), the  $K$  parameter would have a beneficial effect on average fitness.  $K$  would counteract the effects of  $Q$  toward smaller fitness values. This insight suggests also making the use of  $K$  dependent on the elimination probability. The basic idea is to use  $K$  (i.e., to start the chain reaction of agent replacements) only if the most fit agents are eliminated on the basis of the  $Q$  parameter. The  $K$  usage probability can be simply made to be the complement of the elimination probability. This strategy means that the usage of  $K$  is low during the first few generations and may increase with passing generations, as the agents become fitter and fitter. But as more and more generations pass an adaptive  $Q$  also decreases, and so the combined effect  $Q \cdot K$  (the sum of the average length of the chain reactions) is low.

Figure 20 shows a run of the RBE procedure with population size



**Figure 20.** Curves of average fitness for population size  $N = 100$  with fixed  $Q = 6$ ,  $K = 2$ , and adaptive  $Q$  and adaptive usage of  $K = 2$  based on equation (5).

$N = 100$ , 1000 generations with fixed  $Q = 6$  and  $K = 2$  (lower curve, as a reference), and an adaptive run for  $Q$  and the adaptive usage of  $K = 2$ . The probability functions are sigmoid-based (equation (5)) for both  $Q$  and  $K$  usage. Adaptivity clearly yields a better performance.

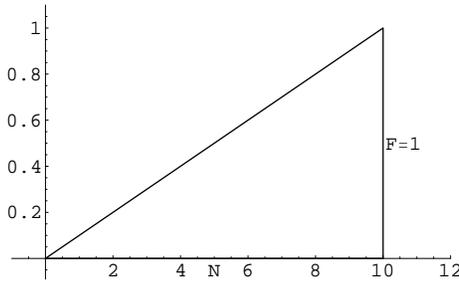
#### 4. Derivation of functions for $Q$ and $K$

In this section we show how to construct functions that for a given  $Q$ ,  $f$  (average fitness),  $K$ , and  $N$ , compute a new average fitness after  $Q$  least fit replacements and  $Q \cdot K$  random replacements.

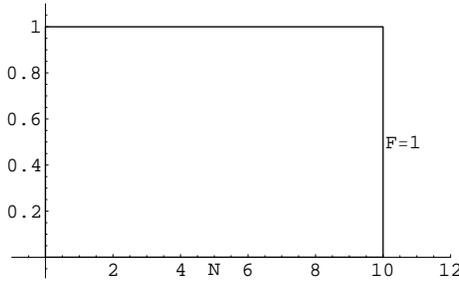
The derivation of the functions is based on the following assumptions. If the population vector is sorted (e.g., in ascending order), the total fitness of a population of size  $N$  will ideally (and on average) obey a regular geometrical distribution. This distribution is right triangular when the value of average fitness  $f = 0.5$  and rectangular for average fitness  $f = 1$ . When  $f$  is between 0.5 and 1 the distribution shows a combination of both. We call this combination quadrilateral or, more specifically, recto-right-triangular because the quadrangle consists of a rectangle and two right triangles. Figures 21, 22, and 23 show three different examples.

The right triangular distribution for average fitness  $f = 0.5$  is the result of sorting (and perhaps adding) vectors of uniformly distributed random numbers. Take a vector of  $n$  random numbers uniformly distributed over  $[0, 1]$ :  $(r_1, r_2, r_3, \dots, r_n)$ ; sort it, for instance, in ascending order. Build another vector of the same form, sort it, and add it to the first one. Repeat this procedure  $m$  times, and then divide each site of the vector by  $m$ . If  $n$  (the length of the vector) and  $m$  (the number of iterations) is big enough, a plot of the resulting vector will be an almost perfect straight line with a slope of  $1/n$  (Figure 24).

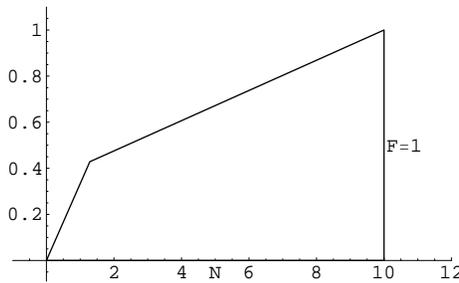
The right triangular distribution is typical at the start of the RBE



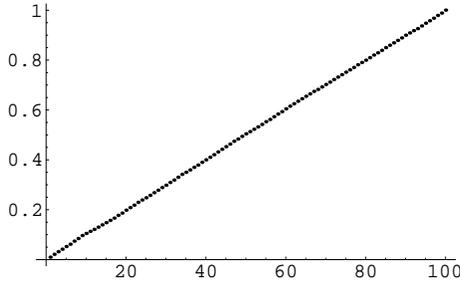
**Figure 21.** Sorted regular right triangular distribution of fitness for a population of size  $N = 10$ , average fitness  $f = 0.5$ , and total fitness of  $N \cdot F/2$ . The value of  $F$ , representing maximum possible fitness, is necessarily 1.



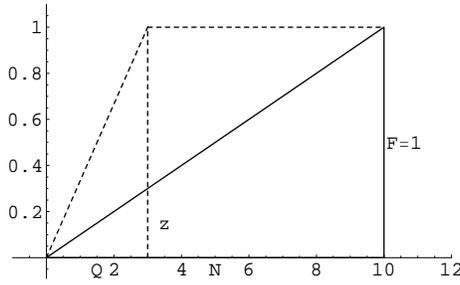
**Figure 22.** Regular rectangular distribution of fitness for a population of size  $N = 10$ , average fitness is 1, and total fitness of  $N \cdot F$ .



**Figure 23.** Sorted regular recto-right-triangular distribution of fitness for a population of size  $N = 10$ ; total fitness for the population  $N \cdot f$  ( $f$  being the average fitness) equals the sum of the area of the quadrangle. In this case average fitness  $f = 0.65$ .



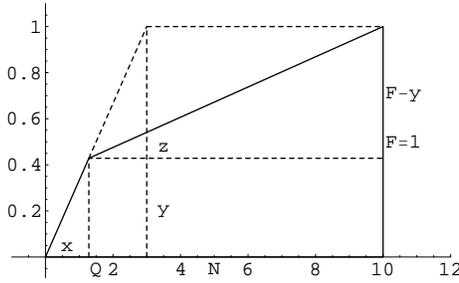
**Figure 24.** Right triangular distribution of 100 sorted averaged random numbers uniformly distributed over  $[0, 1]$  with 100 iterations.



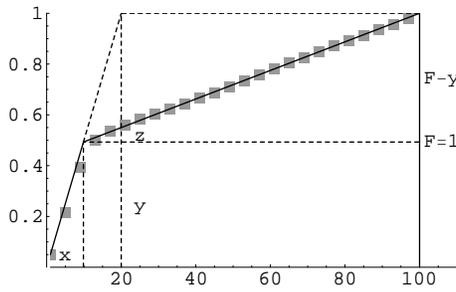
**Figure 25.** Right triangular distribution of fitness for a population of size  $N = 10$ , average fitness is 0.5, total fitness is  $N \cdot F/2$ , and  $Q = 3$ .

procedure. What is then the effect of the first  $Q$  replacements? See Figure 25 where, for the sake of clarity regarding geometrical relationships,  $N = 10$  and  $Q = 2$ . Remember, however, that the average fitness dynamics do not change if  $Q$  and  $N$  are multiplied by the same factor, which means that the ratio of  $Q$  to  $N$  is important and not their absolute values (section 3).

The area of the right triangle is  $NF/2 = Nf$ , where  $f = 0.5$  is the (ideal) average fitness of the population at the outset. Eliminating the least  $Q$  fit agents means subtracting from this area the area of the small right triangle with base  $Q$  and height  $z$ . Replacing the  $Q$  agents with new random ones amounts to adding to this result the quantity  $0.5Q$  (which is the area of the right triangle with base  $Q$  and height  $F = 1$ ). Thus the new area is:  $Nf - Qz/2 + 0.5Q$ . The value of  $z$  is simply calculated from the similarity of the two right triangles with short sides of length  $Q$  and  $z$ , and  $N$  and  $F$ , respectively, stating that  $z : Q = F : N$ . The new average fitness is obtained by dividing the new area by  $N$ . So, this is the average effect of the first  $Q$  replacements. For successive iterations (generations), the sorted distribution will be recto-right-triangular (or



**Figure 26.** Sorted recto-right-triangular distribution of fitness for a population of size  $N = 10$ , average fitness of 0.65, and  $Q = 3$ .



**Figure 27.** Curve (boxes) of the fitness distribution for a population of size  $N = 100$  after four iterations of the RBE procedure with  $Q = 20$ , shown together with the corresponding theoretical recto-right-triangular distribution. The data is the average of 50 different runs. The average fitness is 0.7.

at least this is our assumption). See the example given in Figure 26.

Figure 26 consists of a rectangle with base  $N - x$  and height  $y$ , and two right triangles with short sides of length  $x$  and  $y$ , and  $N - x$  and  $F - y$ . As before, the area of the figure equals  $N \cdot f$ , where  $f$  is the average fitness of the population. We assume that the effect of  $Q$  in each iteration is to increase the value of  $y$  pushing up, as it were, the upper-left edge of the quadrangle along the hypotenuse of the right triangle with short sides of length  $Q$  and  $F$ . This central assumption is confirmed by empirical data. The example given in Figure 27 shows the curve (boxes) of the sorted fitness distribution for a population of size  $N = 100$  after four iterations of the RBE procedure with  $Q = 20$ . The data is the average of 50 different runs.

To determine the effect of  $Q$  in each case we need to know the values of  $x$ ,  $y$ , and  $z$  to be able to calculate the area to be subtracted from the total area of the quadrangle, which, by definition, equals  $N \cdot f$ . Thus we

have (see Figure 26):

$$Nf = \frac{xy}{2} + y(N-x) + \frac{1}{2}(N-x)(F-y). \quad (6)$$

Because of the similarity of the two right triangles with short sides of length  $x$  and  $y$ , and  $Q$  and  $F$ ,  $x$  and  $y$  are further related in the following way:

$$x : y = Q : F. \quad (7)$$

Substituting 1 for  $F$  and solving equations (6) and (7) we get the general formulas for  $x$  and  $y$ :

$$x = \frac{(2f-1)NQ}{N-Q} \quad (8)$$

$$y = \frac{(2f-1)N}{N-Q}. \quad (9)$$

Given  $x$  and  $y$ , the value of  $z$  is simply calculated from the similarity of the two right triangles with short sides of length  $Q-x$  and  $z$ , and  $N-x$  and  $F-y$ :

$$z : (Q-x) = (F-y) : (N-x), \quad (10)$$

then, substituting 1 for  $F$  we get:

$$z = \frac{(Q-x)(1-y)}{N-x}. \quad (11)$$

When  $f = 0.5$ , the sorted distribution is right triangular (see Figure 25),  $x$  and  $y$  are zero, and equation (11) for  $z$  simply reduces to  $z = Q/N$ , the formula that we found previously. Now, to find the new average fitness when the  $Q$  least fit agents have been replaced, the following area (see Figure 26) must be subtracted from the original area:  $S = xy/2 + y(Q-x) + z(Q-x)/2$ . The new area  $A$  is then:

$$A = Nf - S + 0.5Q$$

and the new average fitness is  $A/N$ .

We can now define a function, TEQ (for theoretical effect of  $Q$ ), which takes three arguments,  $Q$ ,  $N$ , and  $f$  (current average fitness), and calculates the new (theoretical) average fitness in one iteration.

```
TEQ(Q, N, f) :=
  x = (2 f - 1) N Q / (N - Q);
  y = (2 f - 1) N / (N - Q);
  z = (Q - x) (1 - y) / (N - x);
  S = x y / 2 + y (Q - x) + z (Q - x) / 2;
  A = N f - S + 0.5 Q;
  RETURN A/N
```

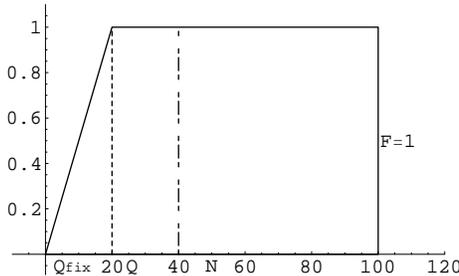
Note that  $Q$  must be less than  $N$  in order to avoid an infinite expression in the calculation of  $x$ . If  $Q = 0$  then  $x$ ,  $z$ , and  $S$  are also

zero, and the value returned by TEQ is simply  $f$ . Note also that the argument  $f$  is assumed to be in  $[0.5, 1]$ , but even if this constraint is made explicit, TEQ is not yet complete. To understand why, note first that when  $x = Q$ , then  $y = 1$ , and the area of the quadrangle represents (as a fixed point of TEQ) the maximum possible (average) fitness that can be reached with the given value of  $Q$  (section 2, equation (1)). Now, what happens when  $x$  is greater than  $Q$ ? To understand this, we have to consider the fixed points of TEQ, which depend on  $Q$  and  $N$ . As explained in section 2, the maximum possible average ( $f_{\max}$ ) fitness equals  $1 - 0.5Q/N$ . In keeping with the equation for  $x$ , the value of  $x$  will always be greater than  $Q$  if  $f$  (the argument of TEQ) is greater than  $f_{\max}$ . Take, for instance,  $Q = 25$ ,  $f = 0.9$ , and  $N = 100$ . The value of  $f_{\max} = 0.875$ , and  $x = 80/3 > 25$ . If the expression for  $f_{\max}$  is substituted for  $f$  in the equation for  $x$ , the equation simplifies to  $x = Q$ , and this means that  $x > Q$  if  $f > f_{\max}$ . The problem with using the value of  $x$  is that it is difficult to understand what has to be done in the case that the test  $x > Q$  is positive. This difficulty can be worked around as follows. Given  $N$  and  $f$  we assume that  $f$  is a fixed point and calculate the corresponding value of  $Q$ . If  $f$  is a fixed point, then  $f = 1 - 0.5Q/N$ , and solving for  $Q$  we get:  $Q = 2(1 - f)N$ . Let us call this  $Q_{\text{fix}}$  (for fixed point value of  $Q$ ) in order to distinguish it from the  $Q$  given as an argument of TEQ. If  $Q > Q_{\text{fix}}$ , then we have, for instance, the situation shown in Figure 28 for the fitness distribution.

The area to be subtracted in this case is  $S = Q_{\text{fix}}/2 + Q - Q_{\text{fix}}$ . The interesting point is now that if equation (12) for the new average fitness is simplified, we get exactly the fixed point corresponding to  $Q$  and  $N$ :

$$f_{\text{new}} = \frac{1}{N} \left( Nf - \left( \frac{Q_{\text{fix}}}{2} + Q - Q_{\text{fix}} \right) + 0.5Q \right). \tag{12}$$

Substituting  $Q_{\text{fix}} = 2(1 - f)N$  into equation (12) for  $f_{\text{new}}$  and simplifying we get  $f_{\text{new}} = 1 - 0.5Q/N$ , which is the same as equation (1). Here



**Figure 28.** Situation for  $Q = 40$  and fixed point value  $Q_{\text{fix}} = 2(1 - f)N$ .  $N = 100$  and  $f = 0.9$ .

is the complete the definition of TEQ.

```

TEQ(Q, N, f) :=
  Qfix = 2 (1-f) N;
  If[Q > Qfix, RETURN 1 - 0.5 Q/N];
  x = (2 f - 1)N Q/(N - Q);
  y = (2 f - 1)N/(N - Q);
  z = (Q - x)(1 - y)/(N - x);
  S = x y/2 + y(Q - x) + z (Q - x)/2;
  A = N f - S + 0.5 Q;
  RETURN A/N

```

The conditional form *IF[test, iftrue]* used in the definition returns *iftrue* if the *test* is true (otherwise *iffalse*, when this is given, as in *IF[test, iftrue, iffalse]*). This is necessary in order to handle the situation when  $Q > Q_{\text{fix}}$ , in which case TEQ returns the fixed point value corresponding to  $Q$  and  $N$ .

Given TEQ, another function can be defined for the calculation of the (theoretical) effect of  $K$ . We call the function TEQK (for theoretical effect of  $Q$  and  $K$ ). It takes four arguments,  $Q$ ,  $K$ ,  $N$ , and  $f$  (current average fitness), and calculates the new (theoretical) average fitness in one iteration.

```

TEQK(Q, K, N, f) :=
  g = TEQ(Q, N, f);
  r = NDR(QK, N);
  S = gr;
  A = Ng - S + 0.5r;
  RETURN A/N

```

TEQK first calculates the new average fitness  $g$  using TEQ. Then  $Q \cdot K$  replacements take place. At this point, the RBE procedure is defined to choose the agents to be replaced at random (Appendix B). Thus there might be overlapping choices. The procedure NDR (for number of different results) is assumed to compute the expected or average number of genuinely different replacements  $r$  given  $Q \cdot K$  and  $N$ . As an example:  $\text{NDR}(40,100)$  equals approximately 33, which means that, for a (population) vector of length 100, the expected number of replacements that take place when  $Q \cdot K = 40$  is about 33.  $\text{NDR}(100,100)$  yields 63 (result used in section 3, Figure 18).

The next steps are simple. The quantity  $S$  to be subtracted from the new area equals  $g \cdot r$  (i.e.,  $r$  times the new average fitness before  $K$  is used), the quantity to be added equals  $0.5 \cdot r$ .

As for NDR, we cannot give an exact analytical definition here (its combinatorial mathematics seems to be formidable), but we claim (without proof) that its value obeys an exponential distribution. If  $r$  is the

number of different results for a given  $Q, K$ , and  $N$ , then (with  $x = Q \cdot K$ ) the following holds:

$$\frac{r}{N} = 1 - e^{-x/N}. \tag{13}$$

Thus NDR can be defined as:

$$\text{NDR}(x, N) := N(1 - \exp(-x/N))$$

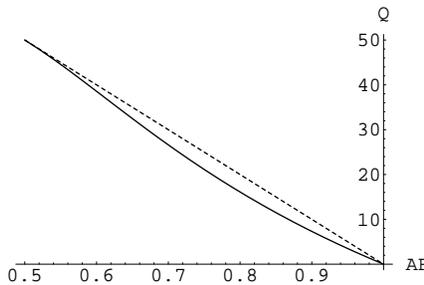
The greater  $N$  is, the better the values returned by NDR in this form approximate the exact values (which we have computed for small  $Q$  and  $K$  values). For values of  $x > 4$ , we have checked the approximation using an iterative stochastic definition of NDR, which is extremely simple to program.

On the basis of these functions, it is now easy to define a function that for a given average fitness  $f$  and population size  $N$  yields the  $Q$  value which is best in the sense that  $\text{TEQ}(Q - 1, N, f) < \text{TEQ}(Q, N, f) > \text{TEQ}(Q + 1, N, f)$ . The function, which we call BestQ, may be based on the first derivative of TEQ, or simply iterate TEQ for values of  $Q$  from 1 to  $N$  to find the maximum. A plot of such a function is shown in Figure 29. The straight dashed line is shown to compare deviations of the curve from linearity.

The definition of BestQ using the partial derivative of TEQ with respect to  $Q$  (leaving aside the use of  $Q_{\text{fix}}$ ) can be given as follows.

The derivative  $t'$  is:

$$t' = \frac{4(f - 1)^2 N^3 + 8(f - 1) N^2 Q + (3 + 8f - 8f^2) N Q^2 - 4f Q^3}{2N^2(N - 2fQ)^2}. \tag{14}$$



**Figure 29.** Plot of the function BestQ( $f, N$ ) for a given  $f$  (average fitness), varying from 0.5 to 1, and a fixed  $N$  (100, in this case) that provides a value of  $Q$  that will give us the best increment of fitness. The straight dashed line is shown for comparison.

Equating  $t'$  to zero and solving for  $Q$  we get the following three solutions for  $Q$ :

$$Q_1 = 2(1 - f)N \quad (15)$$

$$Q_2 = \frac{-3N - \sqrt{9 - 32f + 32f^2N}}{8f} \quad (16)$$

$$Q_3 = \frac{3N - \sqrt{9 - 32f + 32f^2N}}{8f}. \quad (17)$$

To find the right one, we use the additional knowledge that when  $f = 0.5$ ,  $Q$  must be less than  $N$ . Otherwise the new average fitness would be 0.5 and the increment of fitness would be zero. Now, the first formula (which, incidentally, is the equation for the fixed point value of  $Q$ ) and the second formula give  $Q_1 = Q_2 = N$  for  $f = 0.5$ , while the third one yields  $N/2$ .

Here is the complete definition of BestQ.

```
BestQ(f, N) :=
  (3N - sqrt(9 - 32f + 32f^2N)) / 8f
```

The form `sqrt(...)` in the definition stands for  $\sqrt{\dots}$  and  $f^2$  means  $f^2$ . For BestQ to produce a correct result,  $f$  must be  $\geq 0.5$ . If  $f = 1$ , then  $Q = 0$ . Note that  $Q$  does not necessarily have to be an integer.

Note, finally, that the  $K$  parameter is not considered at all in the definition. A consideration of  $K$  is easily possible if BestQ is based on iterating TEQK for different values of  $Q$  from 1 to  $N$ . For the moment there is no simple analytical solution we can propose. From experiments with TEQK it is evident that with increasing  $K$  the value of  $Q$  which yields the best increment of fitness must be increasingly smaller than the value given by BestQ for the corresponding fitness. Consider, for instance, the case that the population size  $N$  is fixed at 100. The values of BestQ for  $f = 0.5$  equals 50, but even for  $K = 1$  this is not the best  $Q$  (the best  $Q$ , calculated iterating TEQK, is 38). For  $K = 2$  and  $K = 3$  the best  $Q$  values are 29 and 23 respectively.

To conclude this section, we want to mention that the RBE procedure is robust with respect to the type of distribution from which fitness values are drawn. We have experimented with runs of RBE working on the basis of normal distributions. Not surprisingly, increase in average fitness is lower if compared with runs of RBE working on the basis of a uniform distribution, but the relative effects of the  $Q$  and  $K$  parameters are comparable. The long-term dynamic shows, however, that for relatively large values of  $Q$ , the greater the evolution time, the more significant the advantage of the normal distribution of fitness. This guarantees the validity of our general qualitative results (section 2)

and is what we understand by robustness. A comprehensive analysis of other types of distributions will be given elsewhere.

## 5. Applications

---

Our results could be important for application problems where an external source of agents (with known distribution) is available and a desired solution cannot be found on the basis of a single agent but only on the basis of the whole population. An optimization could then be carried out simply by replacing “apparently bad” agents (e.g., items in a range of products or people in an organization) with “randomly chosen” new agents, where the choice is not made (and, indeed, cannot be made) on the basis of knowing if the new agents are “really good.” Per Bak somehow recognized this ([1], p. 138) when, on the basis of his model, where only the next two neighbors of an agent in a ring population were considered for dependencies, he suggested that in a business context his selection process would correspond to a manager firing the least efficient worker plus two coworkers and then replacing them with three new people taken in off the street.

With our model, only the right choice of the elimination factor  $Q$  guarantees that we will reach a maximum average quality, depending on the amount of time at our disposal. The lesson to be drawn from the parameter  $K$  is the need to minimize, or possibly eliminate, interdependencies. If this is not possible, then the usage of  $K$  random eliminations should be limited to the case when relatively fit agents are eliminated for specific reasons. While the choice of  $Q$  is independent of scale and domain, this recommendation (certainly not the only possible one) is not.

Consider the following example, but without forgetting that in real life much more is involved than mere statistical averages. Assume it is your task, in some leading capacity, to optimize the average “fitness” or productivity of a group of  $N$  people. Suppose that you cannot vary  $N$ . (A trivial way to increment average fitness is to fire less efficient or “bad” people without replacing them with new ones, the optimum perhaps being attained when you are the only one left. Another trivial approach would be to hire many more than  $N$  new people, and, after a while, to fire most of them and keep the  $N$  best ones.) If you fire bad people, the probability that there will be a chain reaction (i.e., other people leaving out of sympathy with those who were fired) is low. Needless to say, other people should not be fired at random, as Bak suggests. That would be silly. If, however, a “good” member of the group quits or is erroneously fired or is enticed away, the probability for a chain reaction is higher. Good people often exert a positive influence on the fitness of other people. In any case, the conclusion of our model for this particular domain is that you should expect a chain reaction

and compensate for it by hiring new people (choosing them at random if necessary). This interpretation of  $K$  clearly depends on the domain application, but also on the scale or level of evolution. If the agents are “species” it might, for instance, be plausible to view the  $K$  parameter as (almost) independent of “fitness,” as Bak did in the exemplification of his model as an ecosystem with neighboring relationships forming consecutive links in a food chain of prey and predator species.

## Acknowledgments

---

Many thanks to Bernhard Klaassen and Guenter Albers for helpful comments on a previous manuscript version, as well as to Todd Rowland, Managing Editor of *Complex Systems*, and to an anonymous reviewer for a critical reading and valuable suggestions. I also want to thank Thomas Christaller, head of the Fraunhofer Institute for Autonomous Intelligent Systems (AIS), for giving me the opportunity to do this research. Last but not least, I would like to thank Larry Fischer for editing the English text.

## Appendix

Here we provide more formal definitions of the RBE procedure in pseudocode to make it easier for others to check our results. In the list of formal parameters,  $G$  is the number of generations and  $N$  is the size of the population.

### A. Definition of RBE that only uses parameter $Q$

---

RBE( $G, N, Q$ ) :=

1. result = empty vector of length  $G$ ;
2. pop = vector of  $N$  number drawn at random from the uniform distribution;  
 FOR  $i = 1$  to  $G$  DO  
   BEGIN
  - (a) sort pop in ascending order;
  - (b) replace the first  $Q$  positions in the pop vector with random numbers;
  - (c) save average of pop into position  $i$  of result vector;
   END;
3. RETURN result vector

## B. Definition of RBE using parameters Q and K

---

RBE(G,N,Q,K) :=

1. result = empty vector of length G;
2. pop = vector of N number drawn at random from the uniform distribution;  
FOR i = 1 to G DO  
BEGIN
  - (a) sort pop in ascending order;
  - (b) replace the first Q positions in the pop vector with random numbers;
  - (c)  $S = \text{round}(Q \cdot K)$ ; replace S random positions in the pop vector with random numbers;
  - (d) save average of pop into position i of result vector;
 END;
3. RETURN result vector

## C. Definition of RBE based on adaptive Q and adaptive usage of K

---

RBE(G,N,K) :=

1. result = empty vector of length G;
2. pop = vector of N number drawn at random from the uniform distribution;  
FOR i = 1 to G DO  
BEGIN
  - (a)  $C = 0$ ;
  - (b) calculate Q according to the given elimination probability function to mark the positions to change in the pop vector;
  - (c) replace the calculated positions in the pop vector with random numbers, increasing C by 1 according to the complement of the elimination probability of each number to be replaced;
  - (d)  $S = \text{round}(C \cdot K)$ ; replace S random positions in the pop vector with random numbers;
  - (e) save average of pop into position i of result vector;
 END;
3. RETURN result vector

## References

---

- [1] P. Bak, *How Nature Works* (Copernicus Springer-Verlag, New York, 1996).
- [2] K. Sneppen, P. Bak, H. Flyvbjerg, and M. H. Jensen, “Evolution as a Self-organized Critical Phenomenon,” in *Proceedings of the National Academy of Sciences USA*, **92**(May) (1995) 5209–5213.
- [3] D. E. Goldberg and K. Deb, “A Comparative Analysis of Selection Schemes Used in Genetic Algorithms,” in *Foundations of Genetic Algorithms*, edited by G. J. E. Rawlins (Morgan Kaufmann, San Mateo, CA, 1991).
- [4] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, MA, 1996).
- [5] C. R. Shalizi and W. A. Tozler, “A Simple Model of the Evolution of Simple Models of Evolution,” in *ArXiv:adap-org/9910002 v1 15 Oct 1999*, <http://arxiv.org/abs/adap-org/9910002>, 1999.
- [6] R. Dawkins, *The Selfish Gene* (Oxford University Press, Oxford, 1989).
- [7] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution* (Oxford University Press, Oxford, 1993).