EVOLUTION OF EVOLUTION
An Exploratory Work on Genetic Algorithms

by

Deniz Yuret

SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL ENGINEERING AND COMPUTER
SCIENCE IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1992

Signature of Author ____________________
Department of Electrical Engineering and Computer Science
June 1, 1992

Certified by ____________________
Professor Patrick H. Winston
Thesis Supervisor

Accepted by ____________________
Professor Leonard A. Gould
Chairman, Department Committee on Undergraduate Theses

EVOLUTION OF EVOLUTION
An Exploratory Work on Genetic Algorithms

by

Deniz Yuret

Submitted to the Department of Electrical Engineering and Computer Science on June 1, 1992 in partial fulfillment of the requirements for the degree of Bachelor of Science.

Abstract

Genetic algorithms are search procedures that are inspired by natural evolution. This paper in itself is an evolution of ideas on genetic algorithms.

Thesis Supervisor: Professor Patrick H. Winston
Title: Professor of Computer Science
Director, Artificial Intelligence Laboratory
Massachusetts Institute of Technology

Table of Contents

... And there was light

"We can program a computer to solve any problem by trial and error, without knowing how to solve it in advance, provided only that we have a way to recognize when the problem is solved."

– Marvin Minsky

* * *

"The subject of combinatorial optimization consists of a set of problems that are central to the disciplines of computer science and engineering. Research in this area aims at developing efficient techniques for finding minimum or maximum values of a function of very many independent variables. This function, usually called the cost function of objective function, represents a quantitative measure of the *goodness* of some complex system."

– S. Kirkpatrick

* * *

"Any process of exhaustive search can be greatly reduced if we possess some way to detect when *progress* has been made. Then we can trace a path toward a solution, just as a person can climb an unfamiliar hill in the dark – by feeling around, at every step, to find the direction of steepest ascent.

Many easy problems can be solved this way, but for a hard problem, it may be almost as difficult to recognize *progress* as to solve the problem itself. Without a larger overview, the *hill climber* may get stuck forever on some minor peak and never find the mountaintop. There is no foolproof way to avoid this."

– Marvin Minsky

Origin of Species

In this chapter, genetic algorithms, their underlying principles, and their potential uses in optimization problems will be introduced. In the first section, a simple project that gave the author his first inspirations will be discussed. This is the project of perfecting an algorithm of game strategy without expert guidance. Our initial attempt will demonstrate the unfortunate fate of evolution methods that use individual performance as their only selection criteria. Locally optimal solutions and the way they paralyze the progress of evolution will be explained. In the second section Patrick H. Winston's solution of "*devising selection mechanisms that take note of **diversity among individuals**, as well as of individual performance*" will be introduced. Next chapter is devoted to the refinement of such mechanisms. The last chapter describes the final version of the algorithm.

Struggle for Life

It was stated in the abstract that genetic algorithms were search procedures that are inspired by natural evolution. The definition is vague and what is meant by "inspired by natural evolution" should be made clear. Natural evolution is the theory explaining the progress of beings by competition for survival and natural selection. The individuals in the nature tend to form variations either by mating or by mutation. The ones that fit the environment best have the advantage in the struggle for life. The survival of the fittest results in better adaptation to the existing circumstances and the improvement of the traits is thus achieved.

The world of games is built on the spirit of competition, so it provides a natural environment for the use of methods based on evolution. Imagine a world where the only competition for survival proceeds on the game boards. The individuals spend all their lives playing the game against different opponents. At the end of each generation the score table of the grand league is calculated. The survival chance of each individual depends on the total performance. There is some means of reproduction to create new individuals to replace the unfortunate losers. The competition proceeds with the new generation consisting of the survivors and the new offsprings. A progress in skill would be expected through generations, and even perfect players might come to life if we wait for long enough.

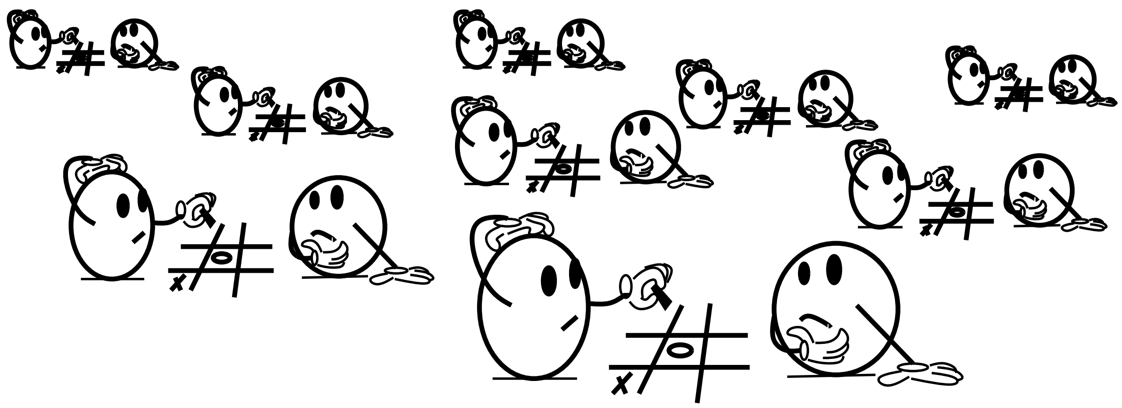

Figure 1: The tic-tac-toe world

The significance of such a method should not be underestimated. In most learning or problem solving methods problem specific information is needed. The progress is achieved under the guidance of a supervisor or a set of carefully selected examples that encapsulate the key information. The evolution procedure described above is independent of the nature and strategy of the problem. So it might be a useful approach when such information is not available. The coordinators of the evolution need only to know the rules and be able to determine the winners. Thus, in principle we are not limited by our own knowledge but only computational power.

This idea was tried on the game tic-tac-toe for simplicity. Various implementations can be imagined for the tic-tac-toe playing individuals. They can be neural nets with the position on the board as input and possible moves as outputs. They can be formulas that compute the goodness of a certain position on the board and the optimum moves can be selected based on this evaluation. The common thing is that the behavior of the individuals depend on a set of coefficients. These coefficients are the "genes" of our individuals. The collection of these genes in each individual make up a chromosome, which is a vector of genes. Thus the whole problem becomes a search for the best chromosome in this gene space. Variation can be achieved by random mutations of particular genes, or by means of mating among individuals using "chromosome crossover". Selection is based on fitness, i.e. the total score received after playing with every other individual.

In my experiments, the individuals started out not even knowing the rules of the game. The ones that made illegal moves i.e. placing a piece on an occupied spot, were punished by low scores. The legal wins and ties were awarded by higher scores. The ultimate goal was to get ties in all the games since that is what perfect tic-tac-toe players will always achieve.

Progress was achieved by the evolution procedure. The population always learnt not to make illegal moves. Although the experiment was tried many times and with several different implementations, the ultimate goal, the perfect tic-tac-toe player was never achieved. Each time the progress would stop with the birth of a player which masters certain lines of play well. When the opponents failed to invent moves outside these lines the result would be convergence of the rest of the population to this long lasting champion. When the variation ceases to exist there is nothing evolution can do...

Tendency to Form Varieties

Tendency to form varieties is stated by Darwin as the fuel of evolution. We can observe rapid progress in the branches of biological hierarchy with a lot of variance. Some of the branches though have stopped a long time ago, and we have species that have remained the same for millions of years. Some forms of crustacea, reptiles, insects and sharks are well known examples. The problem with these species is their almost perfect adaptation to the environment they live in. Almost perfect solutions are fatal for evolution.

The problem in the tic-tac-toe example can be best visualized in the gene space. Our small creatures are actually performing a blind hill climbing to find the global maximum in the space using their respective performances. This would work fine if we had a smooth space with one maximum, but the problem is most problem spaces including this one have many local maxima around the global one. So our creatures do climb a hill and reach to the top, but they have no way of knowing whether they conquered the highest one.

Figure 2: Blind hill climbing is not easy...

Patrick H. Winston offers using diversity as a measure of selection. The selection mechanism has two dimensions. First, the individuals are ranked according to their fitness. Second, each time one gets selected to survive, the rest are ranked according to their diversity from the already selected ones. The selection not only depends on quality only but the combination of these two ranks. This way when a local maxima is conquered, the local champion pushes the rest of the population away from itself. He summarizes this scheme in a principle:

> The **populate-and-conquer principle:** Local maxima should be populated, not avoided, when you are seeking a global maximum.

Being different can be as favorable as being fit for the mechanism of evolution. Discussions on algorithms based on this principle comprise the rest of this thesis.

Natural Selection

In this chapter, building blocks of genetic algorithms will be explained. It will become evident that there is no one way to compose a genetic algorithm, but considerable flexibility of choices in many dimensions. Having stated the dimensions of the programmer's work space we will go through a brief tour of exploration. This brainstorming will uncover some of the intricacies and lead to the final results given in the last chapter.

In the previous chapter we introduced the use of genetic algorithms on a game playing problem. One can think of many other problems from different disciplines. The characteristic of these problems is a complex system whose performance is dependent on many variables. The complexity of the system or the lack of complete information on the performance function prevents us from using ordinary mathematical tools for optimization. Our goal is a search algorithm guided by heuristic principles which will find the optimum solution with maximum efficiency.

The measure of the efficiency is the amount of computation required for the solution. This might depend on the capabilities of our hardware as well as the characteristics of the problem. For example, a parallel architecture would favor procedures that organize search in simultaneous branches, i.e. using a large population. For a game playing problem, performance measure depends on comparisons among individuals, and each comparison requires carrying out a complete game. Since this comprises the bulk of the computation, the search procedure should minimize the number of pairwise comparisons. For an optimization of a complex design problem, calculating the fitness of each individual may require the simulation of the whole design. Thus the total number of individuals in the process should be minimized.

The Problem Space

Throughout the research, the author chose to work in a test environment which maintained the subtleties of real life problems while providing certain specifications. These specifications were needed to remove the problem specific considerations and analyze the algorithm in a petri dish. They also made it possible to perform objective comparisons among algorithms within a controllable environment. Although they are not vital for the theory, a brief explanation now will be helpful to avoid controversies later.

The genes are floating point numbers and the chromosomes are vectors. Each individual, called a "cell", is identified with one chromosome. The number of dimensions is arbitrary, the range of the genes in each dimension is [-1,1].

The fitness is measured by a synthetic function which also has a range of [-1,1]. It is a sinusoid with exponential envelope that has a global maximum at the origin and an adjustable number of local maxima chains surrounding it.

$$\mathbf{fitness}(x_1,\ldots,x_n) = e^{-r^2}\cos(k\pi x_1)\ldots\cos(k\pi x_n)$$

- $x_1,\ldots,x_n$ are the genes of a cell in n dimensional space.
- **r** is the magnitude of the gene vector ($r^2 = x_1^2 + \ldots + x_n^2$)
- **k** is the coefficient that determines the number of local maxima chains.

Figure 1: The fitness function in two dimensions with k = 0, 1 and 2

The Algorithm

Working with the algorithms simulating evolution, one notices a natural split into two distinct procedures. These are what we will call **reproduction** and **selection**. How to create new individuals given the old ones is the problem of reproduction. Seeing the picture as a bunch of cells in the problem space seeking their way to the global maximum, this determines the range of motion of the whole body at each generation. Selection determines the most promising candidates in the population and gives the direction of the motion.

Reproduction

Reproduction is the means to provide varieties. Two ways of reproduction inspired by nature are considered: mutation and mating.

Mutation in nature is a random change in the genes of a cell. Analogously, mutation of a vector will be a random change in one or more of its components. The magnitude and direction of the change, how many genes it is going to involve, and what to do when we get genes outside the [-1,1] range are questions that will be discussed later on.

Mating is the means of creating an offspring out of two individuals. The biological foundation of mating is the chromosome crossover. As a result of the crossover, the offspring gets some of its genes from one parent and the rest from the other. How to select partners, how many genes to get from each parent, and which genes to choose are the degrees of freedom for this method.

The optimum size of the population is still another question. How many offsprings do we want every generation? How many will be selected to survive?

Selection

This is key problem of this thesis. Using fitness alone doesn't work. Using diversity alone will not work. How to combine the two in our decision, is an open problem. If we have a method that favors fitness, we won't be able to get away from our local maxima. If the selection favors diversity, than we will have cells wandering around randomly. The goal is to establish the fine tuning in the balance between fitness and diversity.

The most straightforward way for selection is to sort the population according to fitness and pick the first n cells directly. A more generalized version would calculate the probability of survival for each individual proportional to its fitness. In an arbitrary problem the measure of fitness might have an unexpected distribution so it would be worthwhile to apply a normalization. Or alternatively one might choose to use just ranks instead of the fitness to avoid the magnitude problems. The usual practice is to sort the individuals in a queue. Choose a probability constant p. Select the head of the queue with probability p. If it fails to get selected, remove the head and keep trying with the rest of the queue. This way the probability to select the i^{th} individual will be: $p(1-p)^{i-1}$. This is called the **rank method**.

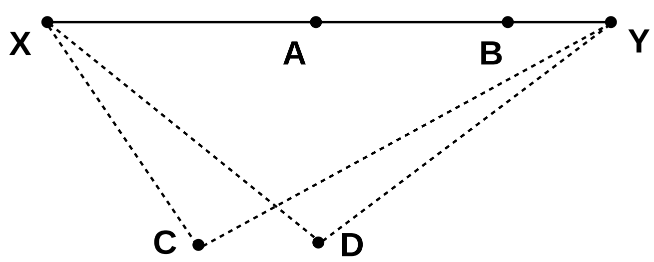

Figure 2: Which one is more diverse with respect to X and Y; A or B? How about C and D?

To include diversity in the selection, some diversity measure should be selected. Given n points in the space, determining which of the $n+1^{st}$ points is **the most diverse**, does not always have a unique solution. For instance, if we have two fixed points at the ends of a line segment and our diversity measure is the sum of distances, point A in the middle of the segment and point B at one end of the segment are equally diverse. However if we choose to use the sum of the inverse squared distances, (or any power of d that is smaller than 1), we will see that point A is more diverse. It is natural to employ the rank method with the diversity measure also. How to combine the two ranks is controversial. The first method we will discuss is to sort the individuals according to the sum of their ranks and use the previously defined *rank method* for selection. This is called the **rank-space method**.

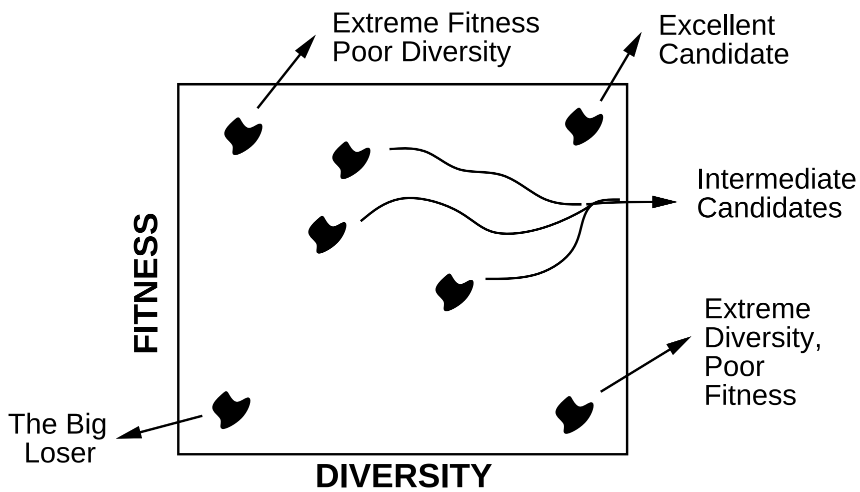

Figure 3: The rank-space

The Search

In this section, the problems and advantages of alternative algorithms will be discussed. The initial rank-space algorithm is based on Chapter 25 of Artificial Intelligence (1992) by Prof. Winston.

The rank-space method

Selecting a candidate using rank method:

- Sort all the individuals and form a queue.
- Select the one at the head of the queue with probability p.
- If it does not get selected, throw it out and repeat the previous step.
- If only one candidate is left, select it.

Combining the ranks using rank-space method:

- Sort the n individuals by quality.
- Sort the n individuals by the sum of their inverse squared distances to already selected candidates.
- Use the rank method, but sort on the sum of the quality rank and the diversity rank, rather than on quality rank.
- Break the ties, if any, in favor of the diverse ones

Number of survivors and the probability to survive, p, are arbitrary and can be specified by the researcher. The selection of inverse squared distances is also arbitrary. The required number of individuals are selected by repetitive applications of the rank-space method and removing the selected ones from the queue.

Reproduction:

- Mutate the n individuals that survived from selection.
- Let all these survivors and the mutants find a partner using the rank-space method, and mate them.

Mutation is done at a single gene which is selected at random. The magnitude is a random number between 0 and some specified maximum radius. The mutations that hit the border are reflected.

Mating procedure simulates crossover. Chromosomes of the parents are taken and a breaking point is selected at random. Both chromosomes get split from the same point and by crossing their pieces two new individuals are created.

Problems and Alternatives

The author's strong faith in the method enabled him to go through a series of painstaking experiments, encounter unexpected problems and invent new solutions. In this section, the major shortcomings of the initial algorithm will be described and solutions will be suggested.

The three major problems are:

- **Saturation Problem:** The number of survivors may fall short of the number of local maxima.
- **Symmetry Problem:** The symmetric positions of local maxima around the global target paralyze the reproduction process.
- **Intermediates Problem:** The cells gathering around the local maxima block the selection of pioneers.

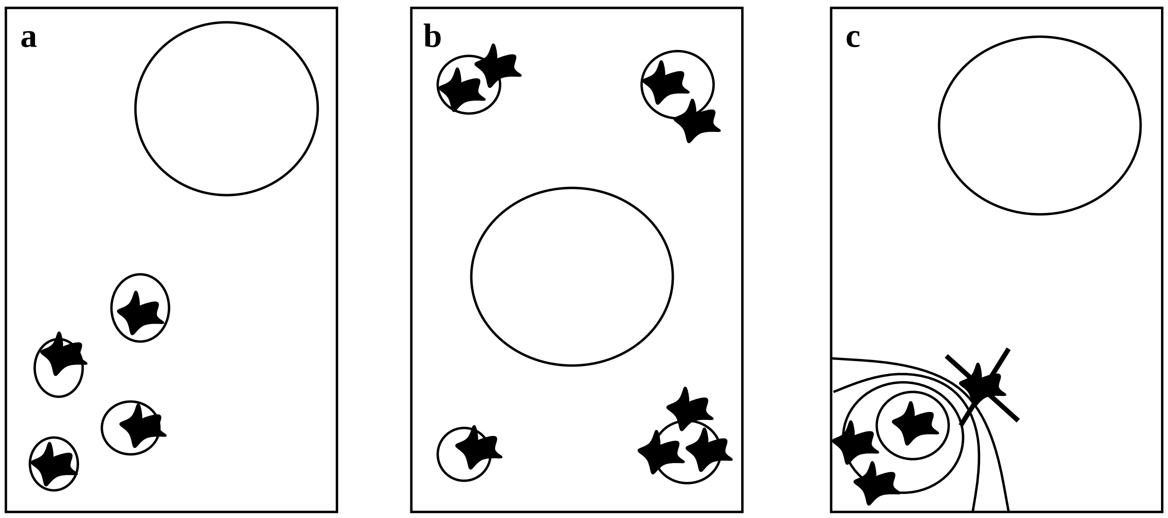

Figure 4: Problems, problems...

The figure illustrates the three problems encountered in different spaces. Small circles represent local maxima, and large circles represent the global maxima. Black blobs represent our cells.

Saturation

The saturation problem is always expected if we use a constant size population and have no information about the smoothness of the space. Remember that the rank-space method only works when the local maxima are populated. At least one member of the population has to stay on the hill and push his friends away. If we have all our members occupied on different hills, there is no one left for exploration.

The key to the solution lies in our **number of survivors** decision. Since we need a generic algorithm we can use with arbitrary problems, a dynamic population would suit best. This way even if we don't know the frequency of local maxima initially, the population can adjust itself and grow in response to new hills encountered.

It would be ideal to have a way of telling that one cell is actually on a local maximum and pin him down there. That way he would get out of the selection and maybe reproduction processes while still performing its repulsion function. All survivor cells get the chance to mutate in each generation. If we keep track of the **age** of the individuals, i.e. number of generations they survive through, the old cells would be the ones that are most likely on local maxima. They have had the chance to mutate in every direction and still failed to produce a better offspring. In the last chapter, the final algorithm will employ these ideas.

Symmetry

The symmetry problem occurs even when the number of the population is adequate. This is due to the dimension dependent implementation of our reproduction algorithms. The cells in Figure 4.b, have no way of creating offsprings close to the center. When they mutate, they cannot get far from their small hills. When they mate, they end up creating offsprings on the small hills again. Since the hills are far apart, they all are almost equally diverse. The ones that try to approach the center have no chance of survival. This problem suggests stronger emphasis on diversity relative to fitness.

Intermediates

This is the most intricate problem of all. Why it happens is not obvious. The author first encountered this problem when the range of mutation was decreased to test the effectiveness of repulsion in the rank-space method. The cells starting from the corner in a two dimensional test space with k=1, would not advance towards the center but instead gather in the corner. A graph would be helpful to visualize the reason.

Figure 5: Symmetric and asymmetric hills

On a symmetric hill the conquerer repulses homogeneously in every direction. So there is always a trade off between height and diversity. Thus after the cell at the top has been selected, all surrounding cells would have equal combined ranks, since the higher they get the closer they are to the leader. Since we break the ties in favor of the diverse ones, the farthest pioneers at the edges would get selected.

However things are not as nice for asymmetric hills. If the gradient on one side of the hill is distinctly greater than the other, the trade off no more exists. The cells that are on the flatter edge of the hill will be higher than the ones on the steep edge. Although they are equal in diversity, the flat side will be favored. This is not wrong, i.e. it is advantageous to take the smoother side of the hill before exploring black holes, but if the smooth side leads nowhere as in the example, we can get stuck in the corner.

Increasing the population would also help in this problem. If we select the second cell at the corner, the following ones are less likely to be selected from that side. Sometimes this problem is encountered in combination with the symmetry problem. When two symmetric hills are conquered, the top ones always get selected, but usually the intermediate cells are favored on explorer ones. This suggests more emphasis on diversity again.

The rank square method

We have seen that some of the problems have been caused by choosing intermediate surrounders instead of real explorers. The rank-space selection mechanism simply adds the two ranks to compare the cells. The indifference curves are straight 45° lines in the rank-space as shown in the first figure. To favor the extremes instead of the intermediates, we need kinked indifference curves as shown in the second figure. This can be achieved by adding the squared scores of the individuals. (Scores are the inverse of the ranks, they show how many individuals you are better than). Basing the selection mechanism on this idea is called the rank-square method.

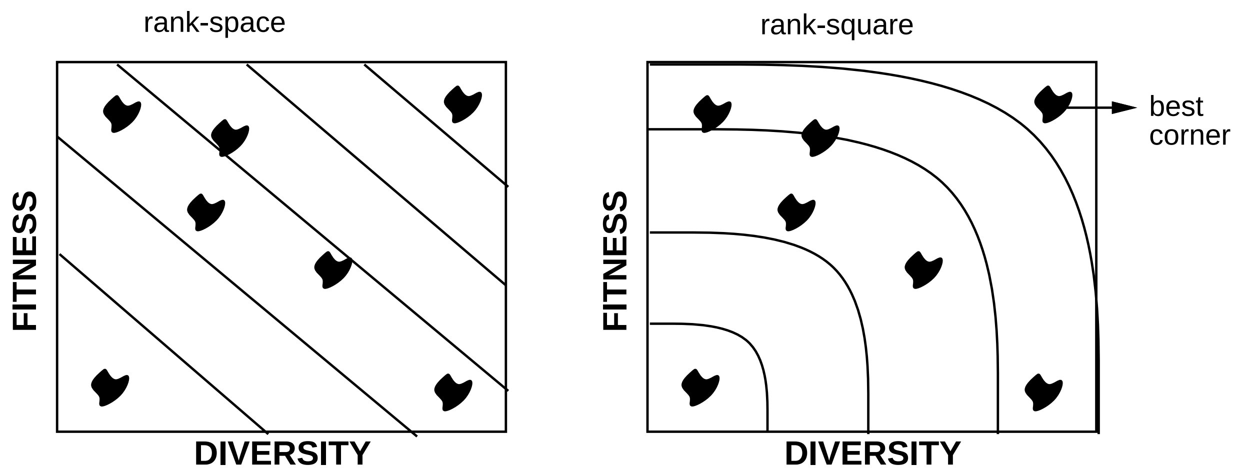

Figure 6: Indifference curves

This attempt gave promising results. It immediately solved the corner problem, and was faster in solving ordinary cases. If kinking the indifference curves helps this much, why not kink them all the way?

The diversity method

Taking this to the extreme was tried. Kinking the indifference curves to the limit means, selecting only the extreme ones i.e. a few from the most diverse ones and a few from the fittest ones. The top few of the fittest ones tend to gather near the leader, so selecting more than one from the fittest list was useless. So the procedure was simplified to picking the fittest one, and selecting the others using only diversity as the criteria. This is called the **diversity method**.

The algorithm based on this method performed impressively, but not as consistent. The speed was magnified due to the lack of the large computation involving combination of the ranks. There was no getting stuck in intermediate positions. The leader would sit on the highest hill encountered so far, and push its offsprings around as diverse as possible. The leadership changed hands when a higher hill was found. In that case the whole population would be exterminated except the new leader and the search would start from the new hill. When all the offsprings reach the farthest available corners, they get stuck, so an extermination and a new start is required.

This method is a great hill climber, but it is blind to the lower hills. It performs well when it finds the global maximum before reaching the corners, but when it misses, a lot of computation is wasted.

Being blind to the lower hills does not seem that bad at the first sight, why should we care about the lower hills when we are looking for the global maximum. Since we are using only diversity now, getting stuck is not a problem on low hills. But this method reveals another important function of populating local maxima: memory. Each cell we leave behind at a low hill is a reminder for the rest that search in that area has been performed. By pushing the offsprings away from itself they encourage search in the unexplored regions. The shortcoming of the diversity method is the loss of this memory function, thus each time it starts a new search it can fail through the exact same path.

Survival of the Fittest

So far we have been wandering around in the design space of the programmer like our small cells. As an advantage, though, we are equipped with brains, not just chromosomes. Using the information gathered so far, it is possible to devise an algorithm which will handle the problems encountered. We have at least reached a local maximum if not the global one.

Problems:

- Problems arising from the insufficiency of population.
- Problems arising from favoring intermediate cells instead of explorers.
- Problems arising from not keeping track of our old path.

Ideas:

- The idea of dynamic population.
- The idea of determining local maxima by age.
- The idea of favoring diversity more relative to fitness.
- The idea of keeping track of old local maxima.

Most of the choices we have made at our initial algorithm were arbitrary. There was no good reason to pick that particular reproduction scheme, number of survivors, probability to survive, picking the sum of inverse squared distances as a measure of diversity etc. It is time to clean up a little bit, and write a new algorithm composed of methods based on reason, considering the problems and ideas that came out of the experiments.

Dynamic Population

To be able to meet the needs for an increasing population while maintaining the required computation in reasonable terms, a complete new structure of population has to be introduced. The new population consists of two groups: the young and the old. The old group starts out empty. The size of the young population is constant and may be dependent on the number of dimensions as well as parallelism in computation. The size of the old population is variable.

Initially the selection among the young is totally based on fitness, since there is no need to worry about diversity before we encounter a local maximum. The reproduction procedure can be chosen arbitrarily. The author sees no reason to choose one way or another since there is always a trade off between speed and size.

To keep things simple a population of one individual and mutation in all the directions will be considered. By the fitness selection, this method is likely to climb up the closest hill with maximum efficiency. To determine arrival on a local maximum, the age of the leader will be monitored. To decrease the chance of a false detection, the range of mutation is decreased with age. Finally at a pre-determined maximum age, when the close mutations provide no further progress, the leader will join the old.

The old never die, never move, and never reproduce. They are computationally easy to handle. They form the **dynamic** part of our population and provide the **memory** for evolution. Their only function is to enter the diversity calculations for the young. After the first old member, the selection of the young is based on diversity as well as fitness. The diversity is calculated only with respect to the old. There is no need to spread the population if we don't have parallel hardware.

Selection

At this point we have a number of young cells with their fitness and diversity ranks at hand. To combine these ranks in the light of the old problems encountered, a totally new approach is suggested.

The valley principle: Select an individual based on fitness if and only if there exists less fit as well as less diverse ones in the population.

The name valley comes from the application of this principle on a landscape with one hill discovered. Assume that there is already an "old" A on top of the hill. If cell is the fittest of the youngsters, but if it is also the closest, it is likely to be on the same hill with the old. As we have seen before, selecting such intermediates kill the chance of real explorers. However if there are cells that are closer to the old than A, then it is likely to be on its way to a new hill, since the lower ones in the middle assure the existence of a valley. Although things aren't quite the same when there are many old cells, the principle still gives valuable results.

Summary

- Maintain two groups of population, young and old. Keep track of the ages of the young group.
- Mutate the young individuals to all possible directions (2n of them in n dimensions) in a range inversely proportional to age.
- Form two queues of the young population using the fitness and diversity ranks. Measure diversity relative to the old cells.
- If the top of the fitness queue is not at the bottom of the diversity queue, select it, otherwise throw it out of both queues.
- Repeat the above process until the desired size of the population is reached.
- Check the ages of remaining survivors. Remove the ones that exceed a certain threshold and place them in the old group. These are the local maxima, and will be the memories of old mistakes for the evolution.

...

The real evolution never has a conclusion...