BioSystems, 22 (1989) 163 – 172
Elsevier Scientific Publishers Ireland Ltd.

Population processing — a powerful class of parallel algorithms

W. Banzhaf

Institut für theoretische Physik und Synergetik der Universität Stuttgart, Pfaffenwaldring 57/IV D-7000 Stuttgart 80 (F.R.G.)

(Received February 10th, 1988)
(Revision received May 16th, 1988)

We present a model for optimization of cost functions by a population of parallel processors and argue that especially diploid recombination of gene strings is a promising recipe for optimization which nature proliferates. Based on a simulated evolutionary search strategy diploidy is introduced as a means for maintaining variability in computational problems with large numbers of local extrema. A differentiation into genotypes and phenotypes is performed. The applied strategy is compared to some traditional algorithms simulating evolution on the basis of two sample cost functions.

Key words: Parallel processing; Optimization; Search strategies; Evolution; Population; Genetic hill-climbing; Diploidy

1. Introduction

Interesting problems which are very difficult for conventional electronic computers can be cast into the form of optimization of some kind of quality or cost function. The problems we consider range from pattern classification tasks such as, for example, face recognition, over optimization in a literal sense, such as the travelling salesman problem, up to general adaptive or intelligent behaviour as used, for instance, to play a game like chess adequately.

The above-mentioned quality function is more or less well-defined, depending on the special nature of the problem under consideration and the granularity of the underlying phase space. In most of the cases, this function possesses many local minima (maxima) which are highly favourable with respect to its neighbourhood, but will complicate search for the global optimum considerably. Depending on the search strategy, it is often possible to devote more computational time to improve the global quality of the solution proliferated by the algorithm.

By looking at nature, we can learn a lot about search strategies in high-dimensional spaces with almost infinitely many local optima. The evolution of life and processes of adaption applied by different species during evolutionary time scales could serve as guidelines for developing reasonable computer algorithms (Darwin, 1859; Dobzhansky et al., 1977). In recent times this point of view has been adopted by many scientists (Bremermann et al., 1965; Cohen, 1973; Rechenberg, 1973; Holland, 1975; DeJong, 1980; Schwefel, 1981; Brady, 1985). From a biological point of view the problem of minimization of a cost function may be seen as that of maximization of a given fitness function.

As Rechenberg (1973) has pointed out, the supposition that evolution has selected the fastest and most secure search strategy (thus giving rise to a meta-evolution of methods during the last 10^9 years) is very reasonable.

0303-2647/89/$03.50
Published and Printed in Ireland

Concerning the quality of solutions nature has found in her feature space, there is no doubt of its sub-global optimality. We term it sub-globally optimal since, by definition, there may exist only one global optimum to a problem. We consider, however, a situation where the quality function is very sensible to changes in the environment. For one species, this environment will contain all other species as well as more slowly varying non-living environmental conditions. Thus we have many subspaces of different dimensions – sometimes partially overlapping – the different optima of which will be called sub-global. Note that this view differs considerably from some opinions stated during the last two decades (Bremermann, 1965; Cohen, 1973), since we assume a very high "adaption velocity" of species and of the evolution process in general.

There exist numerous examples that demonstrate the optimality of solutions nature has found in different areas (Rosen, 1967). The optimal branching structure of vascular trees in the blood system is one prominent example (Cohen, 1954; Kamiya and Togawa, 1972).

The purpose of this contribution is, firstly, to outline some general ideas about missing pieces in the great puzzle of copying nature with respect to her evolutionary plasticity and, secondly, to present primary simulation results concerning at least one critical point in the game: diploid recombination and the differentiation in genotype and phenotype. The emphasis always lies on principles and their application to computing problems.

We begin by outlining the general ideas and then give a brief sketch of the strategies picked up from nature and used in simulations. In the main part we present simulation results with application of different algorithms. Finally we comment on these findings.

2. Some general ideas

We present our ideas as statements which name well-known facts. Simultaneously we want to establish parallels with possible computer applications.

Statement 1: *Nature works with populations of individuals. The phenotypic individuals of a given species are the entity nature uses to search in parallel the space of possible feature combinations. Improvements are accepted and deteriorations are rejected eventually after time delay.*

Transcribed to a parallel processing system: The parallel processors are part of a population of processors which work on the same data each with its given program. Tasks should be partitioned in sufficiently extensive portions so that every processor can work autonomously during a considerable amount of time.

Statement 2: *Nature adapts populations by separating the time scales for the development of individual phenotypes and genotypes. In a first approximation there is only a unidirectional interaction between genotype and phenotype, whereas the reverse interaction is mediated by the gene pool.*

Translated, every processor has its phenotypic part, but as well an underlying structure – the genotype – which evolves rather slowly. The genotypic part determines more or less what is really going on when a processor works with incoming data.

Statement 3: *Recombination is the source of variability in a population as well as a medium to transport important messages, e.g. enormous improvements which should be radiated to all individuals.*

The genotypic part of a processor consists of two portions one of which is dominant, the other recessive (diploid recombination). Every two portions descend from their own "parent processors". These are the processors which send their own genotype to them.

Statement 4: *Randomness is present in the mutations which an individual genotype experiences. Thus noise is used to slightly vary results of recombination.*

In a technical system as well there is an amount of noise which is, however, not suppressed, but sometimes used to accelerate evolutionary changes. If it is not present naturally it is added as white noise of a given variance.

Statement 5: *Selection between individuals assures continuous progress of a population with respect to its adaption to the environment.*

The power of every processor is evaluated and compared to the processor power of the others. On this basis a global decision is made in which one processor's program has to be replaced by a recombination. Note that this process could as well be asynchronous.

Statement 6: *Sub-global optimization is performed by nature on the total population, not on the individual.*

We introduce only one population of processors which means every individual can recombine with everyone else. The variance does not influence fertility rates. Note that this is only a technical agreement. It gives us, however, the result of a single global optimum, since no other species is present. For the sake of clarity, we have only one half female and one half male individuals in our population.

Statement 7: *Improvements are radiated to all individuals in a relatively short time (in number of generations). Thus a fast reaction to changing environmental conditions is possible.*

In principle, almost every processor has a solution which is equal to others in its quality. Distinctions are washed out in a few generations due to a nearly exponential growth in the superior portion.

3. Description of applied strategies

We shall begin our description by distinguishing between simple and advanced evolutionary strategies. In the following, genes are numbers parameterizing the algorithms to be performed by the processors. The simple strategies consist of string-like genotypes

$$x_i^\beta(\mathrm{t}), \quad i = 1, \ldots, Z, \quad \beta = 1, \ldots, N$$

where Z is the number of gene loci and N the number of individuals which could be written as Z components of a high-dimensional vector $\mathbf{x}$. We have N individual processors equipped with N different $\mathbf{x}^\beta$. The numbers in x_i^β which could obey some constraints, such as

$$A \leqslant x_l^\beta \leqslant B \quad \text{or} \quad x_i^\beta \in \mathbb{N}^0 \tag{1}$$

directly specify features of individual β in component i. At the beginning these numbers are chosen randomly from the given set. Based on a quality function $Q(\mathbf{x})$, every processor gets its judgement

$$Q^\beta = Q(\mathbf{x}^\beta)$$

where an ordered set

$$\mathbf{Q}_p = \{Q^{\beta_1}, Q^{\beta_2}, Q^{\beta_3}\ldots\} \tag{2a}$$

with

$$Q^{\beta_1} \geqslant Q^{\beta_2} \geqslant \ldots \geqslant Q^{\beta_N} \tag{2b}$$

of the qualities can be formed. A fixed number D of offsprings emerges from M ($M \leqslant N$) parent processors. Thus a recombination takes place in a bi- or multisexual process. Adding mutations will result in D offsprings

$$\mathbf{y}^\gamma, \quad \gamma = 1, \ldots, D$$

each of which will have its quality $Q(\mathbf{y}^\gamma)$.

For $M = 1$ we have no recombination, only mutations to vary descendants. Now, depending on the particular strategy, the ordered set

$$\mathbf{Q}_d = \{Q^{\gamma_1}, Q^{\gamma_2}, Q^{\gamma_3}\ldots\}$$

$$Q^{\gamma_1} \geqslant Q^{\gamma_2} \geqslant \ldots \geqslant Q^{\gamma_D} \tag{3a}$$

or

$$\mathbf{Q}_{pd} = \{Q^{\beta_1}, Q^{\gamma_1}, Q^{\beta_2}...\}$$

$$Q^{\beta_1} \geqslant Q^{\gamma_1} \geqslant \geqslant Q^{\beta_2}... \qquad (3b)$$

can be taken as a basis to form a new generation of genotypes

$$\mathbf{x}^{\beta}(t + 1), \quad \beta = 1, \ldots, N$$

This is done by selecting the N best individuals, as given by the quality function, either out of D or $D + N$. In the first case, however one needs $D \geqslant N$ whereas the second case does not have any constraint on D. In a real parallel processing system it may well be most efficient to choose $D = N$. Asynchrony can be realized by judging and selecting after some descendants have been generated. In general, selection should then take place under all parents and descendants present at that time.

The advanced evolutionary strategies, on the other hand, differentiate between genotype and phenotype of an individual. The genotype contains M strings, each coming from one of its parents. To avoid communication overhead, diploidy would be an effective choice. Despite the fact that nature uses particular operations such as cross-over and inversion to recombine genetic material, we use a simple random strategy to decide which of the M genetic information at each gene locus is transferred to the descendent as well as to select the dominant one. This simplification does not seem to make a major difference. Thus we have

$$\mathbf{p}^{\beta}(t) = \mathbf{f}\,(\mathbf{x}_a^{\beta}(t)), \quad a = 1, \ldots, M \qquad (4)$$

with the convention of $a = 1$ being dominantly expressed in the phenotype. The corresponding phenotype consequently has features where $\mathbf{f}$ is the mapping from genotype to phenotype. Our aim here does not need any complicated mapping $\mathbf{f}$, so we choose the identity $\mathbf{1}$. We only mention that algorithms with highly non-linear mapping-functions greatly expand the power of population processing (unpublished data). This is, however, beyond the scope of this work.

Recombination was introduced into algorithms simulating evolution more than two decades ago and used by Holland (1975), Brady (1985) and Holland et al. (1986) with considerable success. To our knowledge, however, the present results are the first using diploid recombination by applying the principles of dominance and recessivity of genes found in nature. To distinguish the different strategies using recombination, we term the former intermediate strategies, since its character is haploid. We do not consider them in greater detail here. Note that recombination of recessive genes gives enormous variability in features of phenotypes and is therefore the most important source of variations in a population. A strict selection, on the other hand, will allow only those features to succeed which are superior.

We now describe the other parts of the algorithm. Concerning mutagenicity, we simply apply a constant log-normal partition of mutations (in the diploid case). No rate adaption is used here due to the fact that real progress stems from diploidy rather than from random mutations. In haploid cases, however, we use a mutation rate adaption according to Rechenberg (1973) and Schwefel (1981). The quality function we use does constitute the problem we want to solve using the algorithm. It will be described later.

4. Simulation

We present simulation results concerning two different static cost functions. (They are static with respect to an eventual time dependence which can be considered, too.) We thus look for minima and try to minimize the cost function with simple and advanced evolutionary strategies. Both cost functions are formulated in a 10-dimensional space, the first of which is very harmless and has only one minimum but serves as a demonstration for the following two hypotheses:

(i) In simple evolution strategies where no recombination takes place, mutations are the only source of progress. The velocity in which a system approaches some predefined quality will thus crucially depend on the mutation rate of the system.

(ii) A finite lifetime of individuals which means a selection among descendants excluding parents will for a long time result in a nearly constant average variance of the population. Consequently, such strategies will be candidates for implementation in changing environmental conditions.

Note that the natural time scale we have at hand is the number of generations succeeding one after the other. A generation is defined as an ensemble of individuals, subject to an evaluation of its quality. This definition will include asynchronous updating as well.

Cost function I is defined by

$$Q(\mathbf{p}) = \sum_{i=1}^{10} a_i p_i^2 \tag{5}$$

where a_i are arbitrarily fixed positive values. Here, and in the following, we use p_i for phenotypes and x_i for genotypes. In simple strategies Eqn. (4) is valid as well with M = 1. Minimizing Q analytically results in $\mathbf{p} = \mathbf{0}$. We use simple strategies to solve the problem numerically. The mutation rate follows a log-normal distribution (Rechenberg, 1973; Schwefel, 1981) i.e. the change Δx^β is

$$\Delta x_i^\beta (t) = \sigma_{Ni}\, \sigma^\beta (t) \exp\,[\tilde{\sigma}_N\, \sigma/(10)^{1/2}] \tag{6}$$

where $\left.\begin{matrix}\tilde{\sigma}_{Ni}\\ \tilde{\sigma}_N\end{matrix}\right\} = N(1{,}0)$ is normally distributed.

The updating of the characteristic individual mutation rate σ^β is done by

$$\sigma^\beta (t + 1) = \sigma^\beta (t) \exp\,[\tilde{\sigma}_N\, \sigma/(10)^{1/2}] \tag{7}$$

starting with a fixed σ^β (0). Only the parameter σ will be subject to variations. According to Eqns. (3a,b) we can distinguish between finite life-time (strategy a) of individuals and infinite life-time (strategy b).

Figure 1 gives the course of the cost function for strategies (a) and (b), respectively, for a fixed parameter σ. Qualitatively, strategies (a) differ from (b) in that the average variance over many generations remains nearly constant, whereas in (b) it decays rather rapidly. Figure 2 shows the development of a population of processors in strategy (a). Persisting fluctuations can be realized clearly. Of course, there is no theoretical reason why these fluctuations should persist forever, but a long transient time will proliferate enough variability to cope with changing environmental conditions.

Table I compares convergence velocity defined as number of generations necessary to reach a definite limit in the cost function. It can be seen that the influence of different σ chosen is at least one order of magnitude greater than that of different configurations of the population. This may be one of the rea-

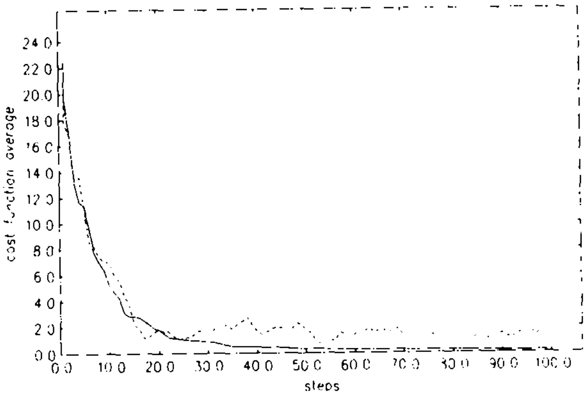

Fig. 1. Cost function I over steps (= number of generations). Average of all exemplars reported. The mutation rate is σ_1 = 0.1. - - - -, haploid (μ,λ)-strategy with μ = N = 5, λ = D = 10. After decaying rather rapidly fluctuations around $Q \approx 2.0$ persist. ———, haploid ($\mu + \lambda$)-strategy with μ = 5, λ = 10. The cost function Q converges, $Q \to 0$.

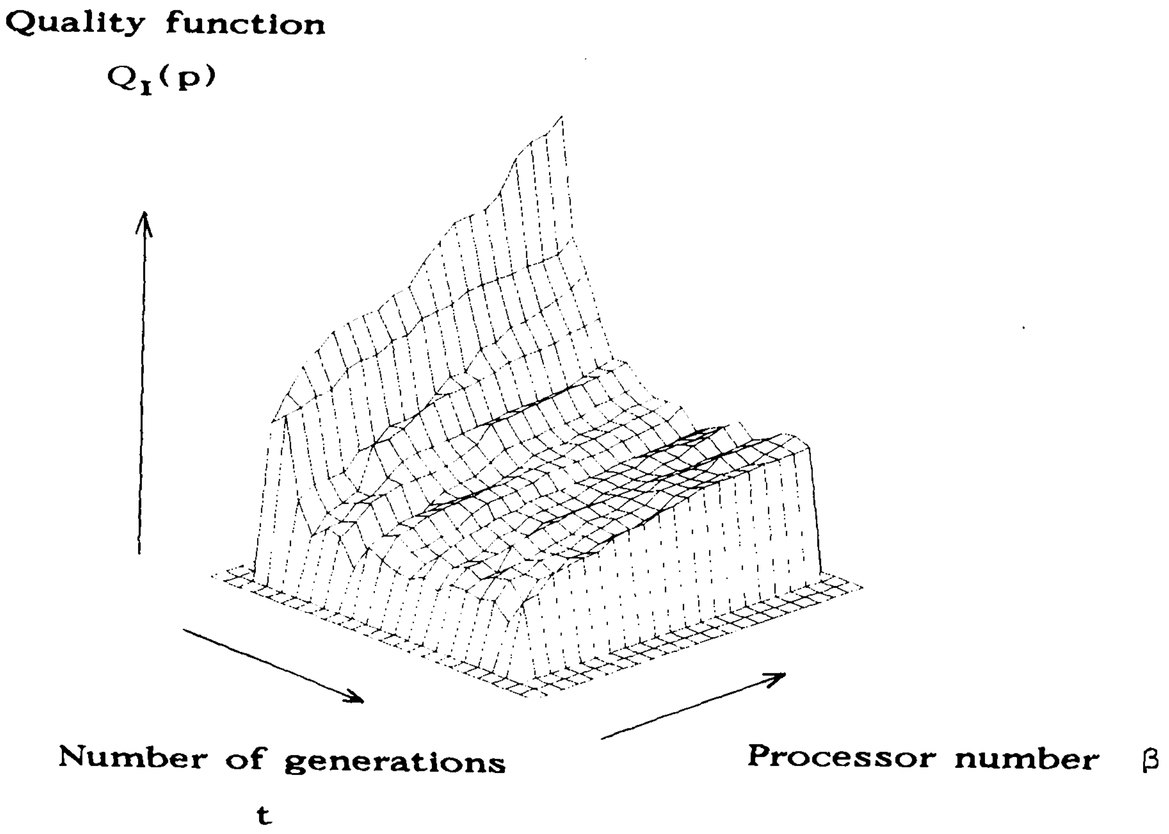

Fig. 2. Cost function *I* of a population of $\mu = 20$ processors. Development over last 40 generations in a (μ,λ)-strategy. Simultaneous fluctuations in the population are clearly visible.

TABLE 1

Comparison of influence of different population configurations μ,λ and variance parameters σ on convergence velocity. Indicated are numbers of generations *t* the population needed to fall short off 1% of cost function *I* at $t = 0$ in haploid (μ,λ) and $(\mu + \lambda)$ (in parentheses) strategies.

Configuration of processor population	Variance parameters			
	$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 1.0$	$\sigma = 10.0$
$\mu = 2$ $\lambda = 12$	>100 (31)	>100 (28)	13 (29)	20 (33)
$\mu = 2$ $\lambda = 20$	>100 (16)	>100 (17)	18 (18)	24 (43)
$\mu = 5$ $\lambda = 30$	>100 (31)	70 (25)	13 (12)	14 (19)
$\mu = 10$ $\lambda = 30$	>100 (19)	56 (27)	11 (11)	7 (21)
$\mu = 20$ $\lambda = 120$	>100 (40)	>100 (27)	12 (11)	9 (8)

sons why in the past evolutionary strategies were felt to be very difficult to handle.

Introduction of diploid recombination as it is done now will change the situation drastically. Two major points are worth mentioning: firstly, recombination supersedes random mutations as the principal source of variations in a population and secondly, we are definitely forced to introduce a distinction between information carried by genes of an individual which will be twofold in the diploid case, and information expressed by an individual which will be single.

Diploid strategies are particularly well suited for problems with many local extrema. To study these effects we introduce a cost function *II*

$$Q(\mathbf{p}) = \sum_{i=1}^{10} a_i p_i^4 + \sum_{i=1}^{10} b_i p_i^2 + \sum_{i=1}^{10} c_i p_i \qquad (8)$$

A special choice of coefficients a_i, b_i, c_i is

$$a_i = 1, \quad b_i = -4, \quad c_i = 1.5 \quad \forall i \tag{9}$$

which will guarantee the presence of 2 local minima in every dimension or 2^{10} minima in 10 dimensions. For the sake of simplicity, all coefficients are chosen equal. These 1024 local minima obey a binomial distribution regarding its quality.

Mutations are still present, but rate adaption (cf. Eqn. (7)) is turned off.

Let us consider a system of 20 processors (10 male and 10 female) with 10 descendants. Diploid strategies will be formally represented from now on by

$$([\mu_1, \mu_2] + ([\lambda_1, \lambda_2]) \tag{10a}$$

(finite lifetime) and

$$([\mu_1, \mu_2], ([\lambda_1, \lambda_2]) \tag{10b}$$

(infinite lifetime). μ_1, μ_2 are numbers of parents (male and female, respectively), λ_1, λ_2 are numbers of male and female descendants. In our case

$$\mu_1 = 10, \quad \mu_2 = 10, \quad \lambda_1 + \lambda_2 = 10 \tag{11}$$

Note that the sex of descendants is not predetermined. Figure 3a is a record of the decrease in cost function *II* of two exemplars. Abrupt transitions are noticeable. This is due to recombinations as the major source of improvements. Different individuals have distinct instants at which progress is made. On the other hand, they follow each other during radiation of improvements. Figure 3b shows the average of the total population together with the best and the worst individual. One can clearly see a decrease in variance between different individuals. From Fig. 3b,c we learn at times when nothing is going on with the best exemplar, i.e. generation steps ∿40 to ∿60, progress is nevertheless occurring: improvements are radiated, variance decreases and the average falls off with undiminished velocity.

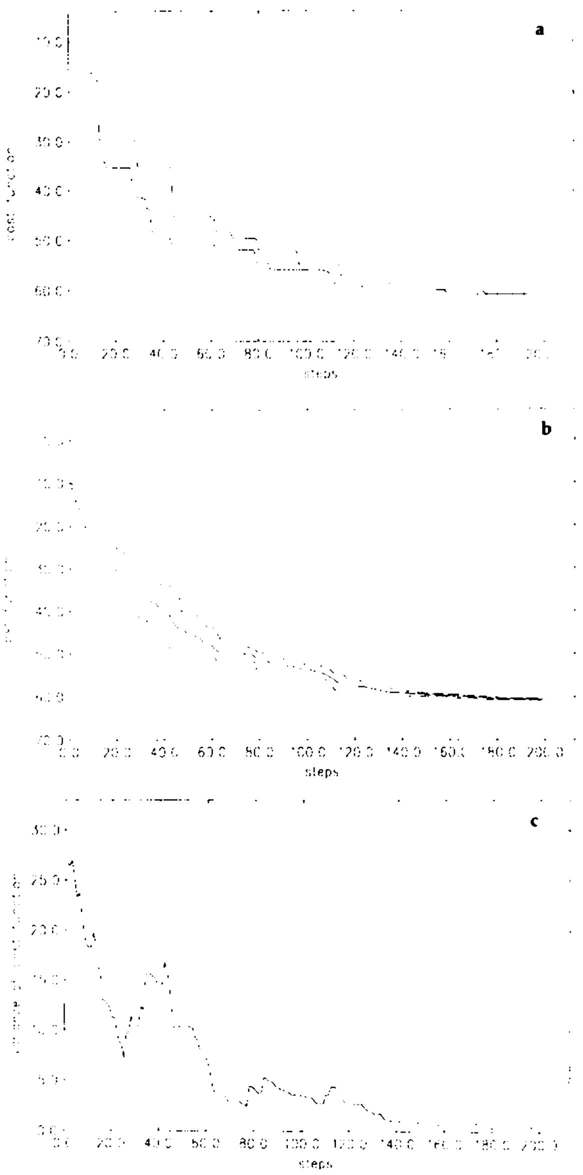

Fig. 3. (a) Course of cost function *II* over number of generations (steps) for two exemplars of the population. Diploid strategy used: ([10,10] + [λ_1, λ_2]), $\lambda_1 + \lambda_2 = 10$. Abrupt leaps are due to recombination. Both exemplars have approached the deepest minimum in around 150 generations. (b) Course of cost function *II* over number of generations (steps) for the best and worst exemplar and for the average of population. After 150 generations virtually all members of the population possess the same quality of its cost functions. (c) Variance of cost function II. Variance is sharply increasing during fast progress of single processors and decreasing during stagnation of the best exemplar.

TABLE 2

Comparison of influence of different population configurations μ_1, μ_2, λ_1, λ_2 and variance parameters σ on convergence velocity. Indicated are numbers of generations t the best exemplar of a population needed to fall short off 10% of cost function *II* at $t = 0$ in diploid ($[\mu_1, \mu_2] + [\lambda_1, \lambda_2]$) strategies.

Configuration of processor population	Variance parameters			
	$\sigma = 0.01$	$\sigma = 0.1$	$\sigma = 1.0$	$\sigma = 10.0$
$\mu_1 = \mu_2 = 2$ $\lambda_1 + \lambda_2 = 2$	333	>500	184	174
$\mu_1 = \mu_2 = 4$ $\lambda_1 + \lambda_2 = 4$	388	193	176	373
$\mu_1 = \mu_2 = 8$ $\lambda_1 + \lambda_2 = 8$	75	91	113	122
$\mu_1 = \mu_2 = 16$ $\lambda_1 + \lambda_2 = 16$	105	80	96	62
$\mu_1 = \mu_2 = 20$ $\lambda_1 + \lambda_2 = 20$	53	87	84	114

Thus a sharp ascent of variance signals an enormous progress of single individuals, whereas a collapse in variance signals the end of development and an equally partitioned quality of solutions.

An important point is the convergence velocity to the best solution, i.e. the deepest minimum. We suppose that it will depend upon the number of individual processors participating at the search, but not on any parameter which regulates mutation. Table 2 gives an overview over 20 searches with different configurations of processors and mutation parameters σ. It is evident that increase in N accelerates the search. Variation of σ, on the other hand, has no stringent consequences.

At the end of this section we present a comparison of tests of different strategies with cost function *II*. We tested strategies such as gradient search, haploid and diploid search applying each strategy to 100 randomly chosen starting states. In Table 3 we give the frequency of appearance of the retrieved minima states over their quality. This clearly demonstrates superiority of diploid strategies.

From the tests one discovers that it is possible to virtually enlarge the population by allowing only one substitution of an individual in every generation. In this way, however, convergence time is lengthened. We emphasize that time consumption considerations need not play an important role here (at least as increases are not exponential) since we are concerned with the question of security of retrieval of global minima which is a greater problem than acceleration of convergence.

TABLE 3

Comparison of different strategies in its ability to discover a global minimum of cost function *II* in 10 dimensions. 100 trials with different starting points in (a) gradient search, (b) haploid search, $\mu = 1$, $\lambda = 10$, (c) haploid search, $\mu = 5$, $\lambda = 10$, (d) diploid search, $\mu_1 = \mu_2 = 10$, $\lambda_1 + \lambda_2 = 20$, (e) diploid search, $\mu_1 = \mu_2 = 10$, $\lambda_1 + \lambda_2 = 40$, (f) diploid search, $\mu_1 = \mu_2 = 20$, $\lambda_1 + \lambda_2 = 20$ converged to the minimum indicated.

Strategy	Number trapped in local minimum of quality											Average quality of solutions
	−61.9	−57.6	−53.4	−49.2	−44.9	−40.7	−36.5	−32.2	−28.0	−23.8	−19.8	
a	–	2	18	36	23	14	5	2	–	–	–	−46.97
b	9	26	32	18	11	4	–	–	–	–	–	−53.06
c	55	24	14	5	2	–	–	–	–	–	–	−58.73
d	83	17	–	–	–	–	–	–	–	–	–	−61.17
e	94	5	1	–	–	–	–	–	–	–	–	−61.60
f	100	–	–	–	–	–	–	–	–	–	–	−61.90
No. of minima	1	10	45	120	210	252	210	120	45	10	1	

Other experiments with a still more hilly cost function (5 minima in every dimension, $5^{10} = 9{,}765{,}625$ minima in 10 dimensions) give evidence for an interesting supposition (Hastings 1986): In a diploid strategy, the number of generations will not grow exponentially if the number of possible solutions does. Indeed, the number of generations required by a typical diploid population with suitable mutation rates is only 500!

5. Conclusions

A model for parallel computing was presented here which has certain similarities to natural evolution. The basic entity was a population of processors which was adapted to a given static problem by means of diploid recombination of "programs" and by mutation. Selection took place under the auspices of a predefined cost or quality function. The programs here were just numbers parameterizing coordinates of a space in which the cost function was well defined. The selection was done deterministically by throwing away all trials which were inferior in a set of fixed size.

It turned out that diploid recombination of genes was a very important ingredient in improving performance of evolutionary algorithms. The traditional approach consisting of mutation of genes as the only source of variability was criticized. It was shown that the security of discovering a global minimum of a relatively complicated cost function was greatly enhanced by introducing diploidy.

On the other hand, mutational effects were not treated adequately. By increasing the population size and introducing a correlation between the quality function and "fecundity" one would reach a more realistic model with respect to biological systems.

We noticed that a distinction between genotype and phenotype of an individual was inevitable and could be seen as a separation of time scales in the evolutionary process. Different adaptive processes are equipped with such a separation of time scales (e.g. Conrad, 1985; Banzhaf, 1987). We took the viewpoint that optimization based on a simulation of natural evolution would lead to powerful and very fast algorithms. This can be seen in contrast to the viewpoint that species are frozen states of evolution or, in other words, caught in a process of development far from being optimal (Cohen, 1973; Holland, 1975).

Evidence for progress in leaps was seen in simulations, the reason of which again was correct recombination of portions of information, each partly superior in some regions. Although results are preliminary in that statistics should be improved, we can notice interesting progress towards a realistic simulation of evolution. The work done can be extended in different directions: one can complicate the mapping function of genotype to phenotype, one can include time-varying cost functions, and one can make selection dependant on other criteria, not necessarily formulated as a cost function. Much remains to be done, since nature will ever provide us with interesting and successful processes to study.

Acknowledgements

I am very grateful to Prof. H. Haken for continuous encouragement and his helpful suggestions as well as to W. Teich and D. Martin for carefully reading the manuscript.

References

Banzhaf, W., 1987, J. Physique 48, 2027.

Brady, R.M., 1985, Nature 317, 804.

Bremermann, H.J., Rogson, M. and Salaff, S., 1965, in: M. Maxfield, A. Callahan, L.J. Fogel (eds.) Biophysics and Cybernetic Systems (Spartan Books, Washington).

Cohen, D., 1973, in: A. Locker (ed.), Biogenesis, Evolution, Homeostasis (Springer-Verlag, Berlin).

Cohn, D.L., 1954, Bull. Math. Biophys. 316, 59.

Conrad, M., 1985, Commun. ACM 28, 464.

Darwin, C., 1859, On the Origin of Species by Means of Natural Selection. (Murray, London) (6th edn., 1972).

DeJong, K., 1980, IEEE Trans. Syst., Man Cybern. 10, 566.

Dobzhansky, T., Ayala, F.J., Stebbins, G.L. and Valentine, J.W., 1977, Evolution, (W.H. Freeman, San Francisco).

Hastings, H.M., 1986, BioSystems 19, 101.

Holland, J.H., 1975, Adaption in Natural and Artificial Systems (University of Michigan Press, Ann Arbor).

Holland, J.H., Holyoak, K.J., Nisbett, R.E. and Thagard, P.R., 1986, Induction, (MIT Press, Cambridge).

Kamiya, A. and Togawa, T., 1972, Bull. Math. Biophys. 34, 431.

Rechenberg, I., 1973, Evolutionsstrategie (Holtzmann Froboog, Stuttgart).

Rosen, R., 1967, Optimality Principles in Biology (Butterworths, London).

Schwefel, H.P., 1981, Numerical Optimization of Computer Models (Wiley, New York).