

Editorial Board:

K.-P. ADLASSNIG

K. BALKUS

P. BALLONOFF

B. BANATHY

G. BROEKSTRA

E. BUCHBERGER

C. CARLSSON

G. CHROUST

G. DORFFNER

W. GASPARSKI

G. GOLDSCHMIDT

W. HORN

R. HOUGH

G.J. KLIR

O. LADANYI

F. LEYMANN

V. MARIK

G. PASK

F. PICHLER

G. PORENTA

H. PRÄHOFER

L.M. RICCIARDI

J.W. ROZENBLIT

N. SHARKEY

A.M. TJOA

H. TROST

S.A. UMPLEBY

G. WUNSCH

CYBERNETICS AND SYSTEMS RESEARCH '92

Vol. 1

Proceedings of the Eleventh European Meeting on
Cybernetics and Systems Research,
organized by the Austrian Society for Cybernetic Studies,
held at the University of Vienna, Austria, 21 – 24 April 1992

Edited by

ROBERT TRAPPL

*University of Vienna
and Austrian Society for Cybernetic Studies*

$$f(p, B) = f(p, 1) = \frac{1}{1 - t_n p}$$

References

1. S. Benedikt, "Ein Optimalitätskriterium für die Steuerung eines Systems im Falle der Unvollständigen Information", *Journal of Cybernetics* 4, 99-106 (1974).
2. S. Benedikt, "An Analysis of Non-Repetitive Decision Making Under Risk", *MTA SZTAKI*, Budapest, (1980).
3. S. Benedikt, "On making single decisions under risk", *Automatika i Telemehanika* 5, Nauka, Moscow, 111-116 (1983).
4. S. Benedikt, "A criterion for choice of an optimal option in the case of single decisions with risk", *Automatika i Telemehanika* 4, Nauka, Moscow, 163-168 (1985).
5. S. Benedikt, "Decision-making under risk in the case of non-numerous repetition of decisions", *IFAC/IFORS/IMACS Symposium Large Scale Systems*, Berlin, (1989).
6. S. Benedikt, "Decision making under uncertainty with incomplete and not completely reliable information", *Doctoral dissertation* submitted for degree D.Sci. Budapest, (1991).
7. P. C. Fishburn, "Decisions and value theory", *John Wiley and Sons*, New York, (1964).
8. D. Kahneman, and A. Tversky, "Prospect theory: An analysis of decision under risk", *Econometrica* 47, 263-291 (1979).
9. A. Kaufmann, "The Science of Decision-making. An Introduction to Praxeology", *World University Library*. (1968).
10. R. L. Keeney and H. Rajffa, "Decisions with Multiple Objectives: Preferences and Value Trade-off", *John Wiley and Sons*, New York, (1976).
11. O. Lange, "Optimális döntések", Budapest, Hungary (1966).
12. A. Rapoport, "Strategy and Conscience", *Karper and Row*, New York, (1967).
13. A. Tversky, "On the elicitation of preferences; descriptive and prescriptive considerations", In D. Bell (ed), *Conflicting Objectives in Decisions*, A Wiley - Interscience Publication. International Institute for Applied Systems Analysis, 209-221 (1977).

APPLICATION OF THE GENETIC ALGORITHM TO A SIMPLIFIED FORM OF THE PHASE PROBLEM

BYRNE LOVELL and MARTIN ZWICK
Systems Science Ph.D. Program
Portland State University
Portland, OR 97207-751

ABSTRACT

The Genetic Algorithm, a technique for global optimization which simulates evolutionary adaptation, is applied to a simplified form of the "phase problem" in theoretical crystallography. Results are compared with those of a problem-specific algorithm.

1. Introduction

1.1 Crystallographic Background

The central mathematical problem in crystallography, known as the "phase problem", arises from attempts to determine the 3-dimensional structure of molecules from the measured amplitudes of scattered X-rays¹. The structure of a molecule is the set of spatial coordinates of its constituent atoms. The number of atoms, their chemical types, and their bond distances and angles are known; their relative locations are not. Structure solution by X-ray crystallography yields a 3-dimensional electron density function, f , which at high resolution has gaussian-shaped maxima at the atomic locations. To obtain f , one needs both amplitudes and phases of the scattered X-rays. Only the former are measurable; hence the "phase problem".

Mathematical methods are available which, by making use of (a) the measured amplitudes and (b) known relationships between the amplitudes and the phases (derived from a priori knowledge about the properties of f), can deduce the phases for small molecules, e.g., of up to one or two hundred atoms. No such algorithms exist for large molecules (proteins, nucleic acids) with thousands of atoms. Such structures can be solved crystallographically only with greater difficulty, by procedures requiring additional experimental data (multiple isomorphous replacement, anomalous scattering). While small molecules can be solved with high probability in weeks or months of research effort, the solution of large molecules is highly uncertain, and may require years of work.

1.2 Optimization Technique

The genetic algorithm (GA)² is a global optimization technique which simulates certain features of evolutionary adaptation as described by population genetics. The GA generates, typically randomly, an initial population (generation) of "individuals", and evaluates each individual for "fitness". An individual here is a point in the domain of the function to be optimized, and fitness is the value of the function at that point. The GA then derives the next generation first by selecting individuals to serve as "parents" according to their relative fitness, and then by modifying them with one or more "genetic operators", in the hope that some of the "offspring" will be more fit than the parents. Further generations are produced similarly until the process is terminated.

It has been claimed²⁻⁶ that the GA offers unusual powers of optimization in many problems where little a priori information about the search space is available, and where traditional methods of optimization are unsuccessful, e.g., due to problems of discontinuity, high dimensionality, or multimodality. The work described in this paper was done a number of years ago, initially motivated by a desire to find new approaches to the phase problem. Although the results encountered were not encouraging, they are reported here because alternative GA approaches may yet prove promising and because the phase problem provides a challenging problem context in which the properties of the GA can be explored.

2. The Genetic Algorithm

In the GA, an individual, I , is represented as string of genes $\{g_1 \dots g_n\}$, each of which assumes one value from a set of possible values known as alleles, $\{a_1, a_2, \dots, a_j\}$. The alleles are numbers, in the present case 0 or 1. A fitness function, $u(I)$, is defined over the space of possible individuals, with each individual having a number of offspring proportional to $u(I)$. The GA forms the next generation by selecting a parent from the previous generation for reproduction, and by applying genetic operators, such as mutation and/or crossover.

Mutation changes a gene from its current value randomly to any other possible value; here, from 0 to 1 or from 1 to 0. When crossover (recombination) is applied to a parent, a "mate" is randomly selected, and an inter-gene crossover point, x , is randomly chosen from the $n-1$ possible locations. (Actually, a smaller feasible region for crossover is defined which excludes any end segments

for which both parents have identical alleles.) Genes $g_1 - g_x$ of the offspring are copied from the first parent; genes $g_{x+1} - g_n$ from the second. For example, if the first parent is represented as 1010, and the mate as 1101, choosing crossover point $x = 2$ yields offspring 1001. Mutation and crossover rates in the ranges $1/20n - 1/n$ (mutation) and $.60 - .80$ (crossover) have been found to be fairly effective^{2,4,5}.

If each offspring were an exact copy of its parent, eventually the population would consist entirely of copies of the most fit individual from the first generation. The function of the crossover operator is to introduce into the population combinations of the genotypes of two individuals which may be more fit than either parent genotype. As crossover rearranges existing alleles but never creates new gene values, it is dependent upon the existence of variety in the set of alleles in the current population. The function of mutation is to maintain this variety, and ensure that each allele is available to the algorithm. If the reproductive advantage accorded the fitter individuals is too high, then variety is lost too quickly; if it is too low, then information gained from previous generations, embodied in the distribution of the current generation, is underutilized. If the mutation rate is too low, potentially adaptive alleles will be missing from the gene pool; if the mutation rate is too high, again information previously gathered about the search space is underutilized, and the optimization can become merely a random search.

3. The Reduced Phase Problem

We now define the "reduced (simplified) phase problem" (RPP) as follows. Atoms are represented as delta functions of constant height (i.e., of only one chemical type) located on grid points in one dimension. Thus, a structure is simply a binary string, e.g., $I = \{01101001\}$, of known length, n , and with known number of 1's (atoms), m . In functional notation, $I = f(k)$, $k = 1$ to n , where $f(k)$ is the value of the k th bit of the string.

The information in the measured amplitudes of the "target" (true) string consists of the string's Patterson function,

$$P(j) = \sum_{k=1}^n f(k) * f(k+j).$$

$P(j)$ is the number of inter-atomic distances of length j ; distances are calculated modulo n (treating the string as

periodic in n), so $P(n) = P(0) = m$. Each pair of atoms contributes two distances to P , one measured in each direction around the loop, that sum to n . The Patterson of the above example is {12303214}.

The Phase Problem, in this framework, is the problem of finding a string, f , that produces the a given (true) Patterson. The Patterson is invariant to rotation and/or reflection of the string, so, for example, the strings {abcd}, {bcda}, {dcba}, {cbad} are all equivalent. With n rotations and n reflected rotations, there can be as many as $2n$ equivalent (and thus correct) strings. An algorithm which always finds one of these strings would constitute a solution to the Reduced Phase Problem.

For simple cases, the Patterson may not actually define a unique string, even aside from rotations and reflections, i.e., there may exist non-equivalent strings with identical Pattersons; these are called "homometric" solutions. However, the possibility of such solutions in three dimensions with complex structures and real data is dismissed by crystallographers as exceedingly unlikely.

4. Applying the GA to the Reduced Phase Problem

4.1 Procedure

The GA program used for these studies was obtained from Kenneth De Jong of the A.I. Laboratory of the Naval Research Laboratory. The program is about 750 lines of Pascal code, and was run under the Berkeley UNIX operating system (version 4.1) on a VAX 11-780.

We have worked with 32-point strings with 4 to 17 atoms, but mainly with an 11-atom string. Difficulty of solution increases with the number of atoms up to maximum difficulty at half-occupancy. We arbitrarily selected an 11-atom "true" string as our target to exemplify a moderately difficult problem. The space of 32-point strings contains $2^{32} = 4.3 \times 10^9$ points; an 11-atom subset, $C(32,11) = 1.3 \times 10^8$ distinct points. The 32 rotations of an 11-atom string are distinct, and the reflections of these rotations may add up to 32 additional distinct strings, all of which have the same Patterson. We are searching, then, for any one of at most 64 equivalent points in a sub-space of 1.3×10^8 points.

Each run began with a random formation of 100 strings. The individuals in this initial generation were evaluated; expected offspring numbers were assigned to each individual based on relative fitness. Parents were randomly selected

one at a time and subjected probabilistically to the genetic operators (crossover and mutation). When 100 offspring had been created, each was labeled with its fitness value. If the best member of the first generation was more fit than any member of the second, it was added to the second generation as the 101st member. The first generation was then replaced by the second.

As some of the parents selected for reproduction may not have been changed by the genetic operators, the number of new genotypes represented in the second generation depends on the mutation rate (MR) and crossover rate (CR). The probability of an offspring being identical to its parent = $(1-CR) \cdot (1-MR)^n$; the expected number of new genotypes per generation = $100 \cdot (1 - (1-CR) \cdot (1-MR)^n)$, or 81 when MR = .001, CR = .80.

As the GA program runs, it reports the best fitness value yet attained, the number of generations formed, the number of individuals evaluated, and a convergence measure. A four-part convergence measure gives the number of genes at which 80, 85, 90, and 95% of the individuals in the current generation have the same allele. If, for example, 95% of the individuals contain identical alleles at 30 of 32 gene locations, then the population has converged to the extent that further recombination of such similar individuals is unlikely to be of benefit.

The fitness function, $u(I)$, maximized by the algorithm was minus the mean square deviation between the Patterson of the true string and the Patterson of the individual trial solution. The goal was to find the global optimum, i.e., a string which gives exactly the known Patterson. Since the Reduced Phase Problem is already a great simplification of the actual phase problem in mathematical crystallography, there is no way to define, for the RPP, what a "good-enough" solution might be.

To minimize premature convergence, an optional procedure, "radiation", was added to the algorithm to inject variability into the population when convergence exceeded a threshold. This was accomplished by changing the rate of mutation to .3 (or .2) for one generation whenever the number of genes (gridpoints) on which there was 90% agreement within a population exceeded some threshold fraction, typically .50 or .75. This high rate of mutation gave each parent a probability of $1 - (1 - .3)^{32} = .999989$ of alteration during reproduction. As most of the parents resemble the fittest individual, this amounted to selecting 100 new individuals randomly from the region near the current fittest individual. (As noted earlier, this individual is always retained in the population unless one of the new individuals is fitter.)

Two other refinements, procedures "align" and "optimized cutpoint", modified the crossover mechanism. Since the Patterson is indifferent to rotation and/or reflection of the string, some good crossovers may be missed due to the parent strings being non-optimally rotated or reflected relative to each other. Procedure "align" enabled the second parent to be aligned for maximum agreement with the first before the crossover point was selected. Another problem arose from the fact that crossover may not preserve the number of atoms even when both parents have the same number. Procedure "optimized cutpoint" reduced the occurrence of such errors.

4.2 Results

Ten models were run, each with eight different random starting populations (Table 1). The target was an 11-atom string, {10001101 10011000 10000100 00100010}. Because the number of new individuals evaluated in each generation is a function of the crossover and mutation rates, and thus varies considerably, the number of function evaluations, rather than the number of generations, was used to quantify the run time of the algorithm. For the eight starting populations, the minimum, maximum, and median number of individuals evaluated before a zero msd was found are listed in Table 1. The best of the models (9) required 23,551 function evaluations, taking approximately 24 minutes of VAX 11-780 cpu time.

Table 1. GA Runs (ali=align; opt=optimized cutpoint; rad=radiation mutation rate; thr=threshold for rad.; run times given in 1000s of function evaluations; * means run terminated w/o success)

model	Model Parameters							Run Times		
	MR	CR	ali	opt	rad	thr	min	max	median	
1	.010	.90	Y	-	.30	.50	5.9	134.0*	34.0	
2	.010	.80	-	-	-	-	4.8	128.0*	32.1	
3	.010	.80	Y	-	-	-	2.7	128.0*	35.6	
4	.001	.80	Y	-	-	-	3.6	124.0*	102.6	
5	.001	.80	Y	-	.30	.75	3.6	124.0*	40.5	
6	.010	.00	Y	-	-	-	5.3	64.1	30.2	
7	.001	.80	-	Y	-	-	4.5	62.3	34.5	
8	.010	.00	Y	-	.20	.75	5.6	136.0*	66.7	
9	.020	.00	-	-	-	-	2.6	133.5*	23.6	
10	.005	.80	-	Y	-	-	5.5	126.1*	28.3	

5. Comparison of Optimization Methods

5.1 An Alternative Algorithm: Pair Flipping

To assess the effectiveness of the genetic algorithm on the phase problem, we compared its results with those of a problem-specific approach, which implemented a discrete version of an accelerated steepest descent, modified to constrain the number of atoms to the correct value. The algorithm starts with a random string with the correct number of atoms. It then evaluates strings generated by all possible "pair-flips", i.e., changing, at different sites, a 1 to a 0 and a 0 to a 1. (With 11 atoms and 21 spaces, this amounts to 231 evaluations, or about $2 \frac{1}{2}$ typical genetic algorithm generations.) The first string with a smaller Patterson error is adopted; if no such string is encountered, a new random (re)starting point is chosen. Median time to success with this algorithm was about 60 seconds, quite a bit faster than the GA.

5.2 Discussion

The problem-specific "pair-flipping" program significantly surpassed the GA in these tests, and was approximately 1/5 the length of the GA program. In the brief tests of 64-point strings, both techniques performed poorly. The real crystallographic search space dwarfs even this 1-dimensional 64-point space, and commonly requires 3-dimensional arrays with at least 2^{15} points (and usually more). Indeed one can reasonably wonder whether any algorithm exists which can solve the RPP for large n , and the problem may be NP-complete.

For $n = 32$, Table 1 shows that the performances of the various GA models are fairly similar. Most striking is the generally good performance of models without crossover (6, 8 & 9), especially in light of Holland's assertion that crossover is one of the GA's most powerful tools and that mutation plays a relatively minor role². Crossover is advantageous when there are genes or sets of genes whose contribution to fitness is more or less independent of other genes. If crossover does not help in the RPP, perhaps it is because there are no such independently valuable alleles.

Indeed, this is so. Each individual is evaluated by the agreement of its Patterson with the observed Patterson at all points. Each point in the Patterson depends for its value on all points of the string, so an allele or a group of alleles cannot have an intrinsic fitness.

With the crossover mechanism ineffective, the GA performs a directed random search using fitness-weighted reproduction and mutation. The space near the fitter individuals is randomly sampled through mutation; when a new fittest individual is discovered, it comes eventually to be a primary locus of search activity.

In such circumstances, and where information about the search space can be directly used in problem-specific techniques, such as the pair-flipping algorithm described above, such specific techniques can surpass the GA's performance. (The pair-flipping algorithm, but not the GA, can intrinsically guarantee that all candidate strings have the correct number of atoms.) However, this study is only a preliminary assessment of the effectiveness of the genetic algorithm on the Reduced Phase Problem. Refinements to the GA introduced here for the RPP (align, optimized cutpoint) need to be further investigated and other GA approaches may show improved results.

6. Acknowledgements

We thank Dr. Roy Rada of Wayne State University for introducing us (MZ) to the GA, Dr. Kenneth De Jong of the Naval Research Laboratory for providing the program - and encouragement - to do these studies, and the Computer Science Department of P.S.U. for use of the VAX.

7. Bibliography

1. Ladd, M. F. E., & R. A. Palmer, *Structure Determination by X-ray Crystallography*, New York: Plenum Press, 1978
2. Holland, J. H., *Adaptation in Natural and Artificial Systems*, Ann Arbor: U. of Michigan Press, 1975
3. De Jong, K. A., *A Genetic-Based Global Function Optimization Technique*, Dept. of Computer Science, U. of Pittsburgh, Pittsburgh, Technical Report 80-2, 1980
4. Brindle, A., *Genetic Algorithms for Function Optimization*, Ph.D. Dissertation, Dept. of Computing Science, U. of Alberta, Edmonton, Alberta, TR81-2, 1981
5. De Jong, K. A., *Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Ph.D. Dissertation, Dept. of Computer & Communication Sciences, U. of Michigan, Ann Arbor, 1975
6. Bethke, A. D., *Genetic Algorithms as Function Optimizers*, Ph.D. Dissertation, Dept. of Computer and Communication Sciences, U. of Michigan, Ann Arbor, 1980

Computer Aided Process Interpretation

Chairpersons: F. Leymann, Germany, and G. Chroust, Austria