GLOBAL OPTIMIZATION STUDIES ON THE 1-D PHASE PROBLEM

Martin Zwick, Byrne Lovell, and Jim Marsh

Systems Science Ph.D. Program, Portland State University, Portland, OR 97207

The Genetic Algorithm (GA) and Simulated Annealing (SA), two techniques for global optimization, were applied to a *reduced* (simplified) form of the *phase problem* (RPP) in computational crystallography. Results were compared with those of "enhanced pair flipping" (EPF), a more elaborate problem-specific algorithm incorporating local and global searches. Not surprisingly, EPF did better than the GA or SA approaches, but the existence of GA and SA techniques more advanced than those used in this study suggest that these techniques still hold promise for phase problem applications. The RPP is, furthermore, an excellent test problem for such global optimization methods.

INDEX TERMS: *phase problem, computational crystallography, global optimization, Genetic Algorithm, simulated annealing*

1. CRYSTALLOGRAPHIC BACKGROUND

1.1 *The Phase Problem*

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 intensities, $|F(h,k,l)|^2$, of scattered X-rays [Ladd & Palmer, 1978]. 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. f is the Fourier Transform of the amplitudes and phases of the scattered X-rays.

$$f(x,y,z) = \sum\sum\sum |F(h,k,l)|\, e^{i(u(h,k,l)-2p(hx+ky+lz))} \quad (1)$$

Only amplitudes, $|F(h,k,l)|$, are measured; the phases, $\phi(h,k,l)$, are not observable; hence the "phase problem".

Mathematical methods are available which make 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) to 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 are solved crystallographically 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, the solution of large molecules is highly uncertain, and may require years of work.

1.2 *The "Reduced" Phase Problem*

We now define a "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., S = {01101001}, of known length, n, and with known number of 1's (atoms), m. f(k) is the value of the kth bit of the string.

The measured amplitudes can be used to generate, by a phaseless Fourier Transform, the self-convolution, called the Patterson function, of the string,

$$P(j) = \sum_{1}^{n} |F(h)|^2 e^{-2\pi ihj/n} = \sum_{1}^{n} f(k)\, f(k+j) \quad (2)$$

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 Reduced Phase Problem is the "inverse problem," i.e., the problem of finding a string that produces the given 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 RPP.

For example, tests of the genetic algorithm utilize a 32-point string having 11 atoms. 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 an one of at most 64 equivalent strings in a sub-space of 1.3×10^8 strings. 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 and are here regarded as acceptable solutions. The possibility of such solutions in three dimensions with complex structures and real data is dismissed by crystallographers as exceedingly unlikely.

1.3 *A Problem of Global Optimization*

The function to be minimized is the mean square Patterson error,

$$E = (1/n) \sum (P_o(j) - P(j))^2 \quad (3)$$

where P_o is the "observed" Patterson, derived from measured amplitudes and corresponding to the "true" f (call it f_o) and P is the calculated Patterson derived from some proposed f. E may equivalently be defined as the mean square difference in x-ray intensities.

The "objective error" for any string is

$$E_f = (1/n) \sum (f_o(j) - f(j))^2 \qquad (4)$$

The phase problem arises because E and E_f are only partially correlated, i.e., because decreases in E do not invariably correspond to decreases in E_f. This non-colinearity makes the phase problem a problem of nonlinear global optimization, to which no general solution exists. This fact is sometimes insufficiently stressed in the crystallographic literature, which abounds in local solution methods, which occasionally are offered as possible solutions to the phase problem despite their strictly local optimizing capability.

The goal of the present study was to find an algorithm which always (or often) yields the global optimum, i.e., the string which gives exactly the known Patterson and thus E = 0. In real crystallographic applications, an algorithm which reliably obtained a solution near the global optimum would be valuable, since local methods (using *a priori* knowledge about f not exploited by the algorithm) might allow successful refinement of a nearly correct solution. However, near correctness cannot here be accepted since the Reduced Phase Problem is already an extreme idealization of the actual mathematical (and practical) problem. There is no way to define, for the RPP, what a "good-enough" solution might be, and no additional constraints to exploit.

The reduced phase problem is of interest for two reasons: (1) it offers a model system for testing phasing methods which might then be applied to macromolecular crystallography; (2) it is a simply defined yet difficult to solve problem on which optimization algorithms can be evaluated, i.e., it is an interesting alternative to the traveling salesman problem as a "paradigmatic" computational problem.

The work described in this paper (and partially reported earlier in [Lovell & Zwick, 1992]) was approximately ten years ago -- as is revealed later in the citation of both hardware and software used, but it remains relevant today: a general solution of the phase problem is not significantly nearer now than it was then, and the RPP is not yet in use as a simple test system either for phase problem explorations or global optimization studies. This investigation was originally motivated by a desire to find new approaches to the phase problem. Although the results obtained did not suggest that the phase problem would be solvable by these means, still, as applications of the Genetic Algorithm and of Simulated Annealing, the efforts reported here are rather preliminary. Better results may be achievable by more sophisticated GA and SA techniques. At the very least, the phase problem provides a problem context in which the properties of the Genetic Algorithm and Simulated Annealing can be explored.

2. THE GENETIC ALGORITHM

2.1 *Basics*

The genetic algorithm (GA) [Holland, 1975; Goldberg, 1989] is a global optimization technique which simulates certain features of evolutionary adaptation as described by population genetics. It has been claimed [Holland, 1975; DeJong, 1980; Brindle, 1981; DeJong, 1975; Bethke, 1980] 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 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.

An individual, I, is represented as string of genes $\{g_1...g_n\}$, each of which assumes one value from a set of possible values known as alleles, $\{a_1,a_2,...,a_j\}$. 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 recombination (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 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 recombination rates in the ranges 1/20n - 1/n (mutation) and .60 - .80 (recombination) have been found to be fairly effective [Holland, 1975; Brindle, 1981; DeJong, 1975].

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 recombination 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 under-utilized, and the optimization can become merely a random search.

2.2 *Application to the Reduced Phase Problem*

The GA program used for these studies was obtained in 1983 from Kenneth De Jong of the A.I. Laboratory of the Naval Research Laboratory. (Obviously, this was a very "primitive" version of a GA program, compared to GA software available today.) 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 worked with 32-point strings with 4 to 17 atoms, but mainly with an 11-atom string. Difficulty of solution increases with the number or 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.

Fitness was defined as -E as given by Eqn-(3). 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. 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 101^{st} 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 (M) and recombination rate (R). The probability of an offspring being identical to its parent = $(1-R)*(1-M)^n$; the expected number of new genotypes per generation = $100*(1 - (1-R)*(1-M)^n)$, or 81 when M = .001, R = .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.

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 recombination 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. (A similar diffficulty arising from circularity in problem representation was encountered in a different context by Belew [1989].) 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.

3. SIMULATED ANNEALING

Simulated Annealing (SA) [Metropolis, Rosenbluth, Teller, & Teller, 1953; Kirkpatrick, Gelatt, & Vecchi, 1983] is another global optimization method which has been widely used in a variety of problem areas. A closely related method, "Molecular Dynamics" [Brunger, 1989] has been applied to crystallographic problems, and SA has also been applied directly to the refinement of phases [Sheldrick, 1989]. SA basically modifies steepest descent by accepting steps which produce higher error, according to a Boltzmann-like probability function, which depends upon a pseudo-temperature defined for the problem. This temperature is set initially at high values and is then lowered according to some annealing schedule. At high temperatures, most steps are accepted and the procedure roams the state space of the problem; as the temperature is lowered, steps which reduce the error are increasingly favored.

In the present study, the procedure used was the following: A change in the solution string, from f to f', produces some change in the error function, from E to E'. Change is accepted with a probability,

$$p = 1/(1 + e^{(E'-E)/T}) \qquad (4)$$

where T is the temperature. At high T, whether E'-E is positive or negative, the second term in the denominator approaches 1, and $p \rightarrow 0.5$; thus at high temperatures, accepting or rejecting a change is equally probable. At low T, if E'>E, the 2nd term in the denominator is large, and $p \rightarrow 0$; if E'<E, the 2nd term is small and $p \rightarrow 1$. Thus at low temperature, steps which increase Patterson error are always rejected, while steps which decrease this error are always accepted.

All steps evaluated involved the movement of an atom to some previously unoccupied grid point, that is, the procedure preserves the correct number of atoms.

The temperature, T, was progressively lowered each step by some cooling rate, C:

$$T' = C\,T \qquad (5)$$

In this study, cooling rates were extremely slow, from .99992 to .999. Runs were terminated either at success or at a cutoff temperature.

4. PROBLEM-SPECIFIC APPROACHES

4.1 *Simple 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 a descent procedure, 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 evaluate 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 2½ 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. The pair-flipping program was approximately 1/5 the length of the GA program.

Note that while the constraint of the known number of atoms is incorporated into pair flipping, and is also a feature of the SA approach, this constraint is not built into the GA representation, although the GA addresses the constraint in the optimized cut-point procedure.

4.2 *Enhanced Pair Flipping*

At the time that the Simulated Annealing studies were done, an improved problem-specific approach was being explored which enhanced simple pair flipping by three means:

> (1) it tried double pair flips after single pair flips have brought about convergence to a local minimum, i.e., it changed two 0's to 1's and two 1's to 0's in an attempt to reduce E;
> (2) it allowed the refinement procedure to continue beyond convergence, i.e., to accept uphill steps (this generates the possibility of limit cycles and thus necessitates the specification of some maximum number of cycles); and
> (3) it employed multiple random starting points.

These features were explicit attempts to get around the local minimum problem. Using double pair flips allowed partial exploration of portions of state space immediately beyond the error "barrier" which defined the current local minimum. The acceptance of uphill steps is an explicit incorporation of an SA-like modification of steepest descent. The use of multiple starting points is a standard approach to the local minimum problem.

When single pair flips are considered, all possibilities are examined and the best was selected (this can be done in $O(n \log_2 n)$ as opposed to $O(n^2)$ calculations). This is not possible for double pair flips, which instead were generated and prioritized from a limited list of the best single pair flips.

5. RESULTS WITH THE GENETIC ALGORITHM

Ten models were run, for n=32, 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 mutation and recombination 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 E 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, 11 atoms on 32 grid points; (M = mutation rate; R = recombination rate; 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 Parameters						Run Times		
model	M	R	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

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 [1975] that crossover is one of the GA's most powerful tools and that mutation plays a relatively minor role. 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. Perhaps 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 it is hard to see how an allele or a small group of alleles (a "schemata") can have a significant degree of intrinsic fitness. This issue needs to be explored theoretically.

If the crossover mechanism is 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.

There is an alternative explanation for the failure of recombination to enhance performance. Perhaps effective recombination requires both that strings be aligned optimally and the constraint of the number of atoms be obeyed. It was an oversight of this study that procedures "align" and "optimized cut-point" were not simultaneously tested in any model; this will be done in future work.

In tests of 64-point strings, the GA performed poorly.

6. RESULTS WITH SIMULATED ANNEALING

Results with Simulated Annealing are summarized in Table-2. For 32 grid points and 16 atoms, near 100% solution rate can be attained if very slow cooling rates are used. Small differences in the cooling rate have a dramatic effect on solution success. For 64 grid points and 16 atoms, success is very significantly reduced, and with 32 atoms, success is essentially minimal. This supports the assertion made earlier that half-occupancy maximizes problem difficulty for any n value.

Table-2. Simulated Annealing Runs. C = cooling rate; n = number of grid points; m = number of atoms. % solution is listed for 100 cases at different cooling rates for 3 different (n,m) combinations.

	% Solution		
	n: 32	64	64
C	m: 16	16	32
.99	54		
.992	53		
.994	59		
.995	73		
.998	86		
.999	84	11	0
.9992	90	10	1
.9994	91	16	1
.9996	94	19	1
.9998	99	13	1
.9999	94		
.99992		15	2

A small anomaly can be noted in the table for n=32: while % Solution increases nearly monotonically as the cooling rate is lowered, the transition from C=.9998 to .9999 reduces success rate from 99% to 94%. The reason for this is that the .9999 run utilized a somewhat higher cutoff temperature than the value for all other runs. This, is a second illustration of the sensitivity of SA performance to SA parameters.

The eleven n=32 runs took a total of six hours of CPU time on an IBM-4381. The seven (n,m) = (64,16) and the seven (n,m) = (64,32) runs together took a total CPU time of about six days. Clearly, on this machine, without a more powerful algorithm, the (64,32) case represents about the limiting size of problem which can be addressed. With current machines, clearly larger sizes can be tackled, but the computational difficulty of this problem is most likely exponential in character, so enormous gains in soluble problem sizes are still unlikely.

7. RESULTS WITH PROBLEM-SPECIFIC APPROACHES

Runs using the pair flipping algorithm on the same VAX 11-780 used for the GA calculations exhibited a median time to solution of about 60 seconds. This is notably superior to the best of the GA runs (model 9) which had a median time to solution of 23.6 minutes.

However, in tests of 64-point strings, simple pair flipping, like the GA, was generally unsuccessful.

In contrast, enhanced pair flipping (EPF) showed improved performance, as summarized in Table-3. These runs were done roughly in parallel with the Simulated Annealing study, and should thus be compared with Table-2.

Table-3. Enhanced Pair Flipping Runs. n = number of grid points; m = number of atoms; nc = maximum number of cycles; ns = number of random starting points. % Soln = number of correct solutions out of 100 cases; Time = CPU time on a Gould-9006 (minutes or hours).

	Run parameters					
run #	n	m	nc	ns	% Soln	Time
1	32	16	8	1	52	4.0 m
2	32	16	32	1	80	11.1 m
3	32	16	32	5	97	22.4 m
4	64	16	64	1	25	3.3 h
5	64	16	64	6	52	15.6 h
6	64	32	64	6	20	20.8 h

It is apparent that EPF achieved results superior to those obtained in the Simulated Annealing study. This is evident in both the % Solution figures and in the longer times requred for SA runs. However, in fairness to the SA efforts, it must be pointed out that the use of double pair flips adds power to the problem specific approach which is not available in the simple pair flipping used by SA. Cycling beyond convergence also represented a direct borrowing, by this problem-specific package, of an essential feature and advantage of Simulated Annealing. Also, it must be acknowledged that the magnitudes of the two efforts were not comparable: the SA calculations were done over several summer months with a small program, while the enhanced pair flipping approach was embedded in a much larger program system developed over a number of years. Finally it should be noted that enhanced pair flipping was notably superior also to an RPP implementation of "density modification," a widely used phase refinement method in macromolecular crystallography [Podjarny, Bhat, & Zwick, 1987].

8. DISCUSSION

Comparing the GA and simple pair flipping on n=32 cases reveal the superiority of the latter. This illustrates the fact that where information about the search space can be directly used in problem-specific techniques, such techniques can surpass the GA's performance. The pair-flipping

algorithm (and the SA approach which also utilized it), but not the GA, intrinsically guarantees that all strings have the correct number of atoms.

Both the Genetic Algorithm and simple pair flipping performed poorly on n=64 cases. Simulated annealing and enhanced pair flipping did better, but the computational requirements for success in such runs already indicate that the limit of these techniques is being approached for machines of the class used. The real crystallographic search space dwarfs 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.

The effectiveness of enhanced pair flipping suggests that in this type of problem, both local and global optimization capability is required. Global optimization capability is obviously needed because the error function is not unimodal, and all structure refinement methods will converge to some local minimum. Good local optimizing capability is also required, since in the RPP, as here defined, an optimization procedure must actually reach the global minimum of E=0, not merely come close to it.

Pair flipping achieves a stronger local minimization when extended by consideration of double flips, while cycling beyond convergence and multiple starting points adds global capability. The Genetic Algorithm might have performed better if it was supplemented with a local minimization capability (in effect implementing non-Darwinian evolution): why, after all, should strings not be at least locally optimal before they are recombined? The desirability of supplementing the GA with a local minimization procedure has also been argued by Belew et all [1989], who used neural nets for this purpose. Similarly, the Simulated Annealing approach might have been enhanced with a stronger local capability, e.g., by consideration of double flips.

Other modifications of the Genetic Algorithm might improve its performance. Perhaps most simply, mutations could be defined as pair flips rather than single 1 to 0 or 0 to 1 changes. As noted earlier, procedures "optimized cut point" and "align" should be simultaneously applied. Other genetic operations might be introduced and other modes of selection utilized. Different representations of f might be used in place of the simple binary string. For example, the array of distances from one atom to the next could be taken as the state variables. Or, the RPP could be defined as a permutation problem, where one starts with a string with the correct number of atoms all, say, at the left-most bit positions, and then permutes 0 and 1 locations. The use of Walsh functions to define f offers yet another representation approach. An optimal representation would intrinsically reduce the search space to single members of each rotation/inversion equivalence class which satisfy the known constraints (number of atoms), yet preserve the syntactic validity of the results of all genetic operations. Altogether aside from its possible value for crystallography, a study investigating the relative merits of these different representations would no doubt shed interesting light on the Genetic Algorithm itself.

The use of coevolution to facilitate optimization might be attempted by using a weighted form of fitness function,

$$E = (1/n) \sum w(j) (P_o(j) - P(j))^2 \qquad (3)'$$

While the string population would evolve to maximize fitness, the population of weighting vectors would evolve to minimize string fitness. Hillis [1991] has found that such coevolution of solutions and problems can enhance optimization performance. Weighting the fitness function so that near-neighbor interactions were more important than far-neighbor interactions might make the problem, in the words of Simon [1962], at least "partially decomposable," and the "schema theorem" of Holland [1975] more applicable. As it stands now, the RPP is a totally non-decomposable problem, so that perhaps it is not surprising that crossover is so ineffective. Indeed, decomposability could be "tuned," and its impact of the efficacy of the GA systematically studied.

There are also a variety of means (other than the "radiation" procedure reported here) by which premature convergence might be prevented; e.g., one would could segregate the string population into subpopulations with limited inter-group breeding. One might even attempt, e.g., via the Genetic Programming approach of Koza [1991], to evolve not string solutions to particular Patterson functions, but RPP-solving algorithms which would then be generally applicable. Similarly, there are no doubt numerous ways by which more powerful forms of Simulated Annealing might be applied to the RPP. AT the least, the SA procedure could itself utilize also double pair flips. It is known that the efficacy of SA depends sensitively on the precise annealing schedule, particularly in the neighborhood of "phase transitions," and methods exist to identify the critical temperature [Basu & Frazer, 1990].

The relative merits of GA and SA optimization is under active investigation and debate. Ingber and Rosen [1992] have claimed that "very fast simulated reannealing" (VFSR) is orders of magnitude more efficient than the GA. Also, it has been argued that hybrid GA-SA methods might be more efficient than either method alone [Judson et al, 1991].

The Reduced Phase Problem is a hard, yet conceptually simple problem, ideal for exploring the capabilities and limitations of GA and SA global optimization techniques. Perhaps it should be added to the ensemble of "standard" problems on which these and similar methods are routinely tested. If new insights into the phase problem are gained or improved solution algorithms are found, these could lead to important contributions to macromolecular crystallography. If new insights are achieved into particular global optimization methods, these methods might perhaps be further improved and their domain of applicability might be better understood.

ACKNOWLEDGEMENTS

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

REFERENCES

Basu A. & Frazer, L. N. [1990]. "Rapid Determination of the Critical Temperature in Simulated Annealing Inversion." *Science*, (249), pp. 1409-1412.

Belew, R. K., McInerney, J., & Schraudolph, N. N. [1989]. "Evolving Networks: Using the Genetic Algorithm with Connectionist Learning." *Artificial Life II*, edited by Langton, C. G., Taylor, C. Farmer, J. D., and Rasmussen, S. Addison-Wesley, pp. 511-547.

Bethke, A. D. [1980]. *Genetic Algorithms as Function Optimizers*, Ph.D. Dissertation, Dept. of Computer and Communication Sciences. U. of Michigan, Ann Arbor.

Brindle, A. [1981]. *Genetic Algorithms for Function Optimization*, Ph.D. Dissertation, Dept. of Computing Science. U. of Alberta, Edmonton, Alberta, TR81-2.

Brunger, A. T., Karplus, M., & Petsko, G. A. [1989]. *Acta Cryst.* (A45), pp. 50-61.

De Jong, K. A. [1980]. *A Genetic-Based Global Function Optimization Technique*, Dept. of Computer Science. U. of Pittsburgh, Pittsburgh, Technical Report 80-2.

De Jong, K. A. [1975]. *Analysis of the Behavior of a Class of Genetic Adaptive Systems*, Ph.D. Dissertation, Dept. of Computer & Communication Sciences. U. of Michigan, Ann Arbor.

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

Hillis, W. D. [1991]. "Co-Evolving Parasites Improve Simulated Evolution as an Optimization Procedure." Langton et al, eds. *Artificial Life II* (see above, Belew reference), pp. 313-324.

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

Ingber, L. & Rosen, B. [1992]. "Genetic Algorithms and Very Fast Simulated Reannealing: A Comparison." preprint from *Mathematical and Computer Modeling*.

Judson, R. S., Colvin, M. E., Meza, J. C., Huffer, A. & Gutierrez [1991]. "Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?" *Sandia Report* SAND91-8740.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. [1983]. *Science* (220), pp. 671-680.

Koza, J. R. [1991]. "Genetic Evolution and Co-Evolution of Computer Programs." Langton et all, eds., *Artificial Life II* (see above, Belew reference), pp. 603-629.

Ladd, M. F. E., & R. A. Palmer [1978]. *Structure Determination by X-ray Crystallography*. Plenum Press, New York.

Lovell, B. & Zwick, M. [1992]. "Application of the Genetic Algorithm to a Simplified Form of the Phase Problem." R. Trappl, ed., *Cybernetics and Systems Research '92* (Proceedings of the Eleventh European Meeting on Cybernetics & Systems Research). World Scientific, New Jersey, pp. 261-268.

Metropolis, N., Rosenbluth, M., Teller, A., and Teller, E. [1953]. *J. Chem. Phys.*, (21), p. 1087.

Podjarny, A. D., Bhat, T. N., & Zwick, M. [1987]."Improving Crystallographic Macromolecular Images: the Real Space Approach." *Ann. Rev. Biophys. Biophys. Chem.*, (16), pp. 351-373.

Sheldrick, G. M. [1989]. "Phase Annealing Direct Methods for Larger Structures," *Acta Cryst.* preprint.

Simon, H. [1962]. "The Architecture of Complexity." *Proc. American Philosophical Society*, (106), pp. 467-482.

Martin Zwick is currently Professor of Systems Science at Portland State University, Portland, Oregon. He received his Ph.D. in Biophysics from MIT in 1968, did postdoctoral work in the Department of Biochemistry of Stanford University, and was Assistant Professor in the Department of Biophysics & Theoretical Biology at the University of Chicago. His research in this period was in mathematical crystallography and macromolecular structure. In the 1970's his interests shifted to systems theory, methodology, and philosophy, and in 1976 he took his present position in the Systems Science Ph.D. Program at PSU. During the years 1984-1989, he was Director of the Program.

His current research is primarily in three areas: (1) information- and set-theoretic modeling (synchronic and time-series analysis of nominal or nominalized data); (2) "artificial life" (evolutionary simulations, genetic algorithm optimization, chaotic & nonchaotic dynamics in cellular automata); (3) systems philosophy (the metaphysics of "problems"). He also continues research in mathematical crystallography using systems methodologies.

Byrne Lovell is a risk and uncertainty analyst in the Financial Services Group at the Bonneville Power Administration, Portland, Oregon, where he helps plan tactical and strategic methods for assessing, reducing, mitigating, or compensating for financial risks in Bonneville's wholesale electric power business.

He received his B. A. in Mathematics from Pomona College (1974), an M. S. in Counseling from the University of Oregon (1980), and will complete his Ph.D. dissertation, titled, "A Taxonomy of Types of Uncertainty," in the Systems Science Ph.D. Program at Portland State University in 1995. He is studying how different types of uncertainty pose different problems for rational decision-making, and as a result are amenable to different forms of amelioration.