

SANDIA REPORT

SAND91-8740
Unlimited Release
Printed December 1991



SAND91-8740
P002
UNCLASSIFIED

12/91
26P STAC

REFERENCE COPY

C.2

Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?

R. S. Judson, M. E. Colvin, J. C. Meza, A. Huffer and D. Gutierrez

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94551
for the United States Department of Energy
under Contract DE-AC04-76DP00789

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of the contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors or subcontractors.

This report has been reproduced from the best available copy.

Available to DOE and DOE contractors from:

Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge TN 37831

Prices available from (615) 576-8401, FTS 626-8401.

Available to the public from:

National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Rd.
Springfield, VA 22161

SAND91-8740
Unlimited Release
Printed December 1991

Do Intelligent Configuration Search Techniques Outperform Random Search for Large Molecules?

R.S. Judson, M.E. Colvin, J.C. Meza, A. Huffer and D. Gutierrez
Center for Computational Engineering
Sandia National Laboratories
Livermore, California 94551-0969
email: rsjuds@sandia.llnl.gov

Keywords: Configuration search, simulated annealing, genetic algorithms, simplex

Abstract

We compare three “intelligent” configuration search methods against random search on a scalable model problem to measure relative performance over a range of problem sizes. Our model problem is a 2-D polymer composed of atoms connected by rigid rods in which all pairs of atoms interact via Lennard-Jones potentials. The global minimum energy can be calculated analytically. The search methods are simulated annealing (SA), genetic algorithms (GA) and Nelder-Mead simplex. Both GA and SA perform progressively better relative to random search as the problem size increases while simplex performs progressively worse. We find that SA and GA have complimentary strengths which implies that a hybrid GA-SA method would be more efficient than either one alone.

1 Introduction

When it was learned in the middle of this century that all terrestrial life is built from just a few varieties of linear biopolymers, it seemed that the secrets of life itself were close at hand. These early hopes were largely dashed as it became clear that the properties of these molecules are as much dependent on their folded conformation (tertiary structure) as on their linear sequence (primary structure). Despite several decades of research, the tertiary structures of proteins and nucleic acids have proven exceedingly difficult to obtain through either experimental measurement or theoretical prediction. Similarly, the properties of man-made polymers are more dependent on the *in-situ* conformation of the chains than on the chemical properties of the monomeric units. Clearly access to the tertiary structure of polymers is a major bottleneck to the progress of materials science and biochemistry.

Experiments and theoretical arguments suggest that the biologically active tertiary structures are those of low free energy. Hence, the search for the biologically important structures is the search for conformations whose energy are close to the the global minimum on the free energy hypersurface. A successful search method needs to find structures close to the global minimum with high probability. An enormous repertoire of optimization methods have been developed during the past several decades[1]. These include gradient methods such as Newton-Raphson, stochastic methods such as simulated annealing[2, 3, 4], and generalized downhill methods such as Simplex[5]. It is convenient to distinguish methods on the basis of how much information they collect about the potential energy surface during the optimization process. Those that progressively “learn” about the global structure of the search space we call “intelligent”. Using this definition, the lowest-intelligence method is pure random search. Molecular configurations are randomly chosen, the energy is calculated at each point, and the lowest minimum energy conformation is saved. In practice, pure random search is ineffective due to the high hills in the potential surface. Instead gradient descent methods are used to find a local minima in the vicinity of the random starting point.

Most optimization methods are by our definition intermediate-intelligence methods. That is, they probe locally in the search space for minima. The local nature of these methods means that while they are very efficient, they must be started from many points in the hyperspace in order to find the global minimum.

Examples of intelligent optimization methods are the so-called Genetic Algorithms[6].

This is a new class of optimization methods based loosely on Darwinian evolution. A large population of initially random bit-strings, each encoding a point on the hypersurface, are evolved with selection pressure favoring the low energy individuals. This method is intelligent because the population globally samples the search space and implicitly refines the search to successively smaller portions of the space as it evolves. Moreover, the individual members of the population exchange information about the distant points in the hypersurface via the evolutionary crossover operator.

Intelligent optimization methods necessarily involve more overhead than simpler methods. The object of this paper is to evaluate whether these methods are efficient compared to random search on a scalable, well-characterized model of the polymer folding problem. Our model molecule is a 2-D polymer in which nearest neighbors are connected by rigid rods and all other pairs of atoms are connected by Lennard-Jones potentials. The lowest energy states of the polymer are hexagonally packed; however there are many compact, but higher energy configurations.

The methods we compare are simulated annealing[2, 3, 4], genetic algorithms (GA's)[6] and the Nelder-Mead Simplex algorithm[5]. Each method generates initial starting points that are passed on to a downhill gradient routine. Our benchmark is random search in which a gradient method is started from random initial configurations. Our model problem scales in a well defined way so that we can see how the relative performance of the methods change with problem size.

2 Computational Methods

2.1 Model problem

As a model problem, we have looked at a two dimensional polymer consisting of N_a atoms connected by rigid rods of unit length. The rods are free to rotate about the atoms at their ends. The configuration of the chain is completely specified by $(N_a - 2)$ angles Θ . The energy of interaction to be minimized is given by a pairwise additive function of the atom-atom distances R_{ij} . Bonds are allowed to cross which can lead to the formation of complicated, intertwined configurations which we refer to as being knotted. Our energy function uses Lennard-Jones potentials between

each pair of atoms

$$V(\Theta[R]) = \sum_{i,j>i} \left[\left(\frac{\alpha}{R_{ij}} \right)^{12} - 2 \left(\frac{\alpha}{R_{ij}} \right)^6 \right]. \quad (1)$$

For our model problem, we chose the value of α equal to 1, which places the potential minima for a given pair of atoms at unit distance. The global energy minimum structures are hexagonally close packed with unit spacing. However, for our polymer to reach the minimum energy configuration, it must arrange itself in one of a small number of special ways consistent with the constraints imposed by the fixed-length bonds. One of these special configurations for the case of 19 atoms is shown in Figure 1a. We will concentrate on 19, 37 and 61 atoms because these can form maximally compact hexagonal structures of radius 2, 3 and 4.

With this energy function Eq.(1), there are two classes of minima, knotted and unknotted. Figure 1a and 1b show unknotted configurations and Figure 1c shows a knotted configuration. Because it costs energy to pull an atom through a knot, with a barrier of about 7750 energy units, knotted configurations will tend to stay knotted using small step-size gradient methods. Simulated annealing (Section 2.2) can undo the knots by taking steps large enough to cross the barrier into the next minimum. Such barriers do not directly affect the genetic algorithm method (Section 2.3) because it does not search along a one dimensional path. The simplex method (Section 2.4) can also jump over barriers. Since each knot imposes a high energy penalty, most of the methods will quickly find at least a few extended, totally unknotted configuration. These can then relax into low energy compact structures. The second class of minima have the chain unknotted and laid out on a hexagonal grid with each atom having 2-6 unit distance nearest neighbor interactions, but with the configuration not being maximally compact. A sample local minimum structure for the 19 atom case is shown in Figure 1b. To rearrange from this to the most compact configuration requires breaking several non-nearest-neighbor bonds and reforming others in a highly specific way.

We can make estimates of what portion of configuration space is taken up with knotted configurations as a function of number of atoms. To do this, we simply choose a large number of random configurations and count how many have one or more knots. For 19 atoms, the percentage with no knots is 0.058. The percentages for 37 and 61 atoms are 0.0012 and $\simeq 10^{-7}$ respectively. So a first measure of the robustness of an optimization method is how often it can find any unknotted configuration having started from a random one. A second measure is how low in

energy is the distribution of unknotted configurations. A third and final measure is the nearness to the global energy minimum of the lowest energy structure found.

2.2 Simulated Annealing

Simulated annealing (SA) is a widely used method. We largely follow the procedures given by Wilson and co-workers[3, 4]. A molecule starts at some point in configuration space and takes Monte Carlo steps where the value of one angle is changed by a small amount with each step. If the energy goes down, a step is accepted. If the energy increases, the step is accepted with probability $\exp[-\Delta E/kT]$ where ΔE is the change in energy, T is the temperature and k is Boltzman's constant. The temperature starts at a relatively high value and gradually cools down. A large number of Monte Carlo steps is performed at each temperature, typically 200-1000.

Our starting temperature was chosen to be high enough that >80% of steps were initially accepted. The step sizes were chosen from a Gaussian random distribution of mean \bar{S} and standard deviation σ . We use an exponential annealing schedule so that $T_{i+1} = \alpha T_i$ where $\alpha < 1$. If at some temperature, the acceptance ratio drops below 25%, the value of α is halved. At the end of each annealing run, the configuration was further minimized using several steps of a non-linear conjugate gradient (CG) routine[8] which typically provided a large decrease in energy over the final annealing result. We made a number of runs from different initial starting points in configuration space. The parameters values used were $k = 1$; $T_{initial} = 20$; $T_{final} = 1$; number of steps per temperature=250 ($N_a = 19$) or 1000 ($N_a = 37$ and 61); number of temperatures=30 ($N_a = 19$) or 50 ($N_a = 37$ and 61). The value of α was chosen to cool from $T_{initial}$ to T_{final} over the appropriate number of steps. The values of \bar{S} and σ were 0 and 10 degrees.

2.3 Genetic Algorithms

Genetic algorithms[6] (GA) are optimization methods based on several strategies from biological evolution which appear to have been remarkably successful in producing organisms that efficiently exploit their environment. The first such strategy is the use of a breeding population in which individuals who are more "fit" in some sense have a higher chance of producing offspring and passing their genetic infor-

mation onto succeeding generations. The second is the use of crossover in which a child's genetic material is a mixture of his parents'. The third is that of mutation, meaning that genetic material is occasionally corrupted, leading to individuals who may or may not be more fit than they would have been otherwise, but always maintaining a certain level of genetic diversity in the population.

In practice, a typical GA is implemented as follows. An individual is coded for by a "gene" which is a bit string of length N_g that can be uniquely decoded to give a set of physical parameters (angles in the case at hand). A fitness function is defined that can discriminate between individuals. For molecular structure optimization the fitness is the total energy. An initial population of N_{pop} individuals is formed by choosing N_{pop} bit strings at random, and evaluating each individual's fitness. Subsequent generations are formed as follows. All parents are ranked by fitness and the highest fitness individuals are placed directly into the next generation with no change. Next pairs of individuals (including the highest fitness individuals) are selected, with probabilities based on their fitness, and their genes are crossed over to form genes of the remaining individuals in the next generation. Crossover consists of taking some subset of the bits from parent 1 and the complimentary set of bits from parent 2 and combining them to form the gene of child 1. The remaining bits from the two parents are combined to form the gene of child 2. Additionally, during replication there is a small probability of a bit flip or mutation in a gene. This serves primarily to maintain diversity and prevent premature convergence in which a single very fit individual takes over the entire population. To bound the magnitude of the effect of mutations, the binary genes are usually Gray coded. It is important to emphasize that crossover is the key feature that distinguishes the GA as an intelligent optimization method. If crossover is ineffective, GA degenerates into a random walk search. The lowest fitness individuals in each generation may be discarded and replaced by children of more fit individuals. See Figure 2 for an illustration of the crossover and mutation steps. Many variants of the algorithm exist in the literature, but the basic model outlined here is representative. We use a modified version of the Genesis[7] code for the calculations reported here.

As with simulated annealing, we found that the addition of conjugate gradient minimization to the GA to be necessary. Because large regions of configuration space are very high in energy due to one or more pairs of atoms being close to one another (whether or not the configuration is knotted), it is necessary to relax each configuration prescribed by the GA using a few steps of nonlinear conjugate gradient optimization[8]. The energy used for fitness ranking is the relaxed value.

We can also return the angles of the relaxed configuration to the GA. By returning the angles (a “Lamarckian” process because a child can acquire “knowledge” learned by the parent) we speed up the convergence of the method. We used the default parameters supplied by Genesis with the exception of the population size, N_{pop} , which was chosen to be 50.

2.4 Nelder-Mead Simplex Method

Another search algorithm we have examined is the Nelder-Mead simplex method[5, 8]. The method derives its name from the idea of iteratively constructing a simplex (a geometrical figure consisting of $n + 1$ vertices, their interconnecting line segments and faces) in n -dimensional space and searching for a minimum by comparing function values at the vertices of the simplex. The Nelder-Mead simplex method starts by generating a simplex from an initial guess supplied by the user. At the beginning of each iteration, the algorithm ranks the vertices from best to worst in terms of function value. The method then searches for a minimum by looking in directions generated by the algorithm in one of three ways: a reflection step, an expansion step, or a contraction step.

The method first generates a trial point by reflecting the simplex about the worst point. The function is evaluated at the reflected point x_r , and compared against the best point. If the value at the reflected point is lower than the best point, then the reflected point is tentatively accepted. Before accepting the reflected point though, an attempt is made to take a longer step in this direction. The reflected simplex is expanded and the function value at the point, x_e , generated by this step is compared against the reflected point, with the better of the two points replacing the current worst point. The reflected step could however generate a vertex at which the function value is worse than the best point. The algorithm has two options in this case. If the reflected point is better than the next to worst point then x_r is accepted and it replaces the worst point. In the second case, the simplex is contracted and a new iteration begins.

The main advantage of the simplex method is the simplicity with which it can be coded. The main disadvantage is that in practice it only works well for small-dimensional problems, as our experiments verified. Although the simplex method was fairly robust at initially decreasing the function value, the method tended to stall out while it was still far away from the solution. So, as with SA and GA’s, we

only used the simplex method as a way to generate initial guesses for more powerful local minimization routines. A nonlinear conjugate gradient method[8] was used for both GA and SA so we also used it with simplex. However, to see if other gradient methods could do a better job of searching locally, we also had the simplex algorithm drive a truncated Newton method. The particular implementation of the Newton method we used, called BTN (block-truncated Newton), was developed by Nash[9]. The BTN code is a general purpose software package for the solution of unconstrained nonlinear optimization problems. The algorithm employed is a variation of the standard Newton method and differs only in the approach used for solving the Newton equations. To avoid the cost of constructing and factoring the Hessian matrix, the BTN code uses an iterative technique known as the Lanczos method. This method is related to the conjugate gradient method for solving systems of linear equations. Since the convergence rate of the Lanczos method is dependent on the condition number of the linear systems, the Lanczos method is usually used in conjunction with a preconditioner. The BTN package provides several options for preconditioning, all of which are automatically computed from information gathered during the iteration process. Although the preconditioners can be reset, the documentation advises against this option. In our experiments however, we found that the BTN method worked best whenever the preconditioner was reset periodically, and it was not unusual to have a factor of 2 decrease in the number of iterations taken by resetting the preconditioner. We suspect that this is due to the widely different character of the function in the various stages of the optimization process.

The simplex method has two parameters that we varied: the size of the initial simplex and the convergence tolerance. The initial simplex was generated by computing n points from the formula:

$$x_i = x_0 + \lambda e_i, \quad i = 1, \dots, n, \quad (2)$$

where $\lambda = 2.0$, and x_0 is the initial guess. Various values of λ were tried, but this value worked best overall. The simplex method used two rules to determine convergence. The first rule specified a tolerance on the difference between the best vertex and the worst vertex. For all of our test cases this value was set to 10^{-5} . The second tolerance specified a reduction in the initial function value. This value was set to 10^{-10} . In our version of the simplex method, we also specified a total number of iterations (10000) it would take. Whenever the simplex method converged for either of the two cases mentioned above, the method was restarted using the best

vertex as an initial guess. The solution from the simplex method was then used as the initial guess for either the conjugate gradient method or the BTN method. The conjugate gradient method was stopped whenever the initial function value had been decreased by a factor of 10^{-3} . The BTN method was stopped whenever the magnitude of the gradient had been reduced below 10^{-3} .

2.5 Random Search

We performed random search optimization on our model problem as a benchmark against which we could compare the other, “intelligent” methods. For the random search calculations, we chose a large number of random initial configurations from which we went downhill using a non-linear conjugate gradient method[8].

3 Numerical Results

For the simulated annealing (SA), simplex and random calculations, we performed many individual optimizations starting from different randomly chosen initial configurations which produced a distribution of final energies. The same set of initial configurations was used for each method and each method was run until approximately 10^7 total function evaluations were used, including those needed to calculate finite difference gradients. The number of individual minimization runs varied as a function of problem size and method. For SA the numbers are 1000, 200 and 121 for 19, 37 and 61 atoms respectively; for simplex/CG the numbers are 707, 593 and 250; for simplex/BTN they are 170, 91 and 250; and for random search they are 7300, 3502 and 932. For the two GA variants (Lamarckian and non-Lamarckian) we used a population size of 50, initially chosen out of the standard random set of configurations. The GA was then evolved over many generations until about 10^7 total function evaluations were used. The number of generations for Lamarckian GA were 299, 199 and 98 for 19, 37 and 61 atoms respectively. The corresponding numbers for non-Lamarckian GA were 169, 69 and 42.

The lowest energies found for each of the methods are given in Table I, compared against the global minimum. For 19 atoms, all of the methods did quite well, reaching within one energy unit of the global minimum. For 37 atoms, SA, GA and simplex/CG still did fairly well but simplex/BTN did measurably worse.

This is an artifact of the stopping criteria used in those trials — a maximum number of minimization steps was allowed with BTN which often were used before the method found a minimum. For 61 atoms, none of the methods reached the global minimum, although SA and GA-Lamarckian perform better than either of the simplex variants. As expected, for high dimensional problems, even random search outperforms simplex. If the CG routine starts from a random point in a region of high gradient, it will take large steps, jumping over barriers and occasionally finding its way into the low energy region. The simplex method starting from the same point tended to roll downhill but not take very large steps. So it tended to stay in the high energy knotted region, find a point with a low gradient, then defer to the gradient routine. At this stage the gradient routine wouldn't go very far, resulting in a high energy final configuration.

We can look at the distributions of minima that the methods find and compare them against one another rather than against the absolute measure of the global minimum. In all cases, configurations with energy greater than zero are knotted and less than zero are unknotted. Notice that the energies greater than zero are plotted on a log scale and in particular that there are always some configurations with energies $> 10^{13}$. These are configurations that are so badly knotted up that the local gradient routine (either CG or BTN) was unable to go anywhere. In Figures 3, 4 and 5, we show the integrated probability distributions as a function of energy for 19, 37 and 61 atoms respectively. For instance, Figure 3 shows that 50% of the final SA values (heavy solid line) had energies less than -40; 85% had energies less than -20; and 100% had energies less than +100.

For 19 atoms, the low energy distributions were similar for most of the methods, but simplex left many more knotted configurations than either SA or GA. All of the intelligent search schemes produced better low energy distributions than did random search. The non-Lamarckian GA showed the best performance, leaving no knotted configurations. Lamarckian GA actually produced fewer of the lowest energy configurations than any of the other methods. For 37 atoms, SA did very well at finding a low energy distribution of configurations, but like simplex, it left quite a few configurations in high energy knotted states. Lamarckian GA had the most unknotted configurations but they were higher in energy on average than those produced by SA. Recall from Table I, though that the lowest single energy was found by Lamarckian GA. All methods produced better low energy distributions than random search, but random search actually found more unknotted configurations overall than did either simplex or SA, for reasons alluded to above. SA and

simplex often find high energy, low gradient points which the gradient methods are not able to get out of. The results are much the same for 61 atoms — SA produced a good distribution of low energy configurations, but left many ($> 80\%$) in high energy, knotted states. Lamarkian GA was much better at unknotting, but was less good at getting unknotted configurations into low energy states. However, it again produced the lowest energy individual. Simplex does poorly (and worse than random search) on all counts — it finds few unknotted configurations and virtually no low energy states. Of the local methods tested with simplex, BTN gave a better distribution than CG. If the maximum allowed number of iterations for BTN was increased it would probably also find lower absolute minima than the simplex/CG hybrid.

4 Discussion

Points on our model energy surface can be classified into one of three regions: (1) high energy, knotted, (2) low energy, unknotted but not minimized and (3) low energy, unknotted and minimized. Each method can be characterized by how efficiently it makes the transition from region 1 to 2 and how well it searches regions 2 and 3 to find low energy points in 3. For small problems, SA, GA and simplex were all efficient at making the jump from region 1 to 2; they efficiently find “reasonable” regions of configurations space. For larger problems, SA and simplex fall increasingly far behind GA in getting out of the high energy region. Additionally, the Lamarkian version of GA becomes increasingly more efficient than the non-Lamarkian version. A GA method is expected to behave well in this situation due to the communication between individuals in the population embodied in the crossover operator. Once one or two individuals find regions significantly lower in energy than the population average, the rest of the population gets dragged along. An encouraging point is that GA continues to be efficient at the largest problem size we tested. GA bears some formal similarities to the simplex algorithm which is known to fail for high dimensional problems. Hence there was some concern that GA would also fail for high dimensional problems, which it clearly does not.

The next requirement is that methods efficiently search the low energy regions 2 and 3 to find optimal points in 3. From Figures 3, 4 and 5 and Table I, we see that SA and GA have complimentary strengths here — SA tends to find a

narrow, low energy distribution of configurations, while GA produces a population having a broader distribution of energies, with individuals both lower and higher in energy than SA. Except for the smallest problem, simplex is outperformed by SA and GA. Its major disadvantage is an inability to go uphill to get over barriers and into lower energy wells. SA explicitly allows movement uphill and GA allows it implicitly by not ruling out the production of children with energies higher than their parents’.

We can make a few broad conclusions that should be applicable to more realistic molecules. For small molecules, simulated annealing followed by a few steps of conjugate gradient remains the easiest route to low energy configurations. Of course for such small molecules, exhaustive search is feasible and clearly warranted in some cases as nicely shown by Saunders and co-workers[10]. For larger problems, when a small number of good (i.e. physically motivated) starting configurations are not readily available, GA is the best method for quickly producing a population of relatively low energy structures. However GA is not very good at fine-tuning structures once the population is reasonably homogeneous in energy. So after a fairly small number of generations, the best structures produced by the GA might be passed on to a more localized search method, probably SA. We are starting to use this approach for a set of real molecules; the results of that work will be reported at a later date. One goal will be to continue to push up the number of degrees of freedom and see exactly how large a molecule we can reasonably expect to treat with these methods.

Finally, we should mention that a number of comparisons between GA and SA have been reported which yield ambiguous results about which method is superior[6, 11]. Many of these studies have concentrated on a single test problem or a suite of problems of similar size. Clearly the relative performance of the methods can be strongly problem dependent. What we see here is that there is also a large size-dependent term in the relative efficiency for the two methods which implies that it is dangerous to extrapolate from performance on small problems to that for larger ones.

Acknowledgements

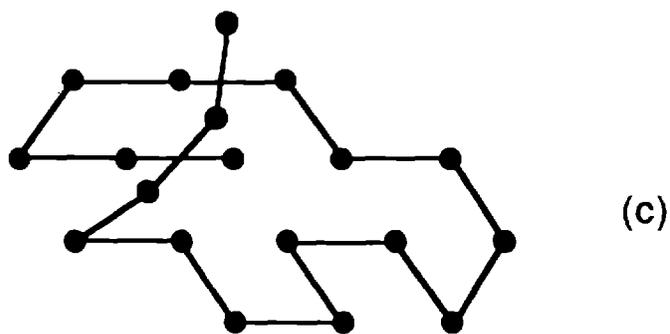
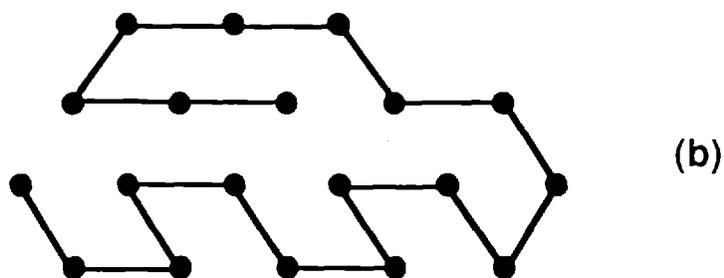
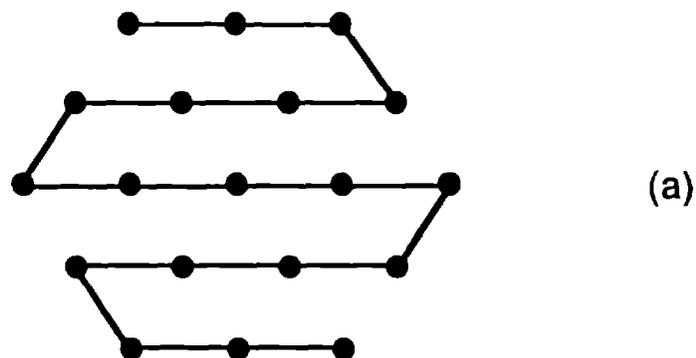
The authors acknowledge support by the Department of Energy under contract DE-ACO4-76DP00789. A.H. acknowledges additional support from the TRAC program administered by Associated Western Universities.

References

- [1] A.R. Leach. *Reviews in Computational Chemistry II*, p. 1. Eds. K.B. Lipkowitz and D.B. Boyd (VCH Publishers, New York, 1991).
- [2] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. *Science*, **220**, 671 (1983); S. Kirkpatrick. *J.Stat.Phys.* , **34**, 975 (1984).
- [3] S.R. Wilson, W. Cui, J.W. Moskowitz and K.E. Schmidt. *Tert.Letts.*, **29**, 4373 (1985).
- [4] S.R. Wilson and W. Cui. *Bicpolymers* , **29**, 225 (1990).
- [5] J.A. Nelder and R. Mead. *Computer Journal*, **7**, 305 (1965).
- [6] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, (Addison-Wesley, 1989); L. Davis. *Handbook of Genetic Algorithms*, (Van Nostrand Reinhold, 1991); *Proc. of Third Inter. Conference on Genetic Algorithms*, J.D. Schaffer, ed. (Morgan Kaufman, 1989).
- [7] J.J. Grefenstette and N.N. Schraudolph. *GENESIS 1.2.ucsd*. (Available via anonymous ftp from the Artificial Life Archive server, iuvax.cs.indiana.edu [129.79.254.192] in the directory pub/aiife/software/unix/GAucsd).
- [8] W.A. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling. *Numerical Recipes in C*, (Cambridge UP, New York, 1988).
- [9] S.G. Nash and A. Scfer. Tech Report 64, Center for Computational Statistics, George Mason University, Fairfax, VA 22030 (submitted to ACM TOMS) (1990).
- [10] M. Saunders. *J.Comput.Chem* **10**, 203 (1989).
- [11] L. Ingber and B. Rosen. *genetic Algorithms and very fast simulated re-annealing: A comparison*, available via email from ingber@umiacs.umd.edu.

Method	Best energy		
	19 atoms	37 atoms	61 atoms
Simulated Annealing	-44.2	-94.8	-164.4
GA (Lamarckian)	-44.3	-97.3	-166.6
GA (non-Lamarckian)	-44.3	-95.1	-156.9
Simplex/CG	-45.3	-96.1	-134.9
Simplex/BTN	-44.3	-88.2	-129.3
Random	-44.3	-95.0	-143.2
Exact	-45.3	-98.3	-171.5

Figure 1. Examples of two compact, unknotted configurations for the 19 atom polymer and a single knotted configuration. For all unknotted local minima, the atoms lie on a hexagonal grid. (a) One of the global minimum structures with the maximum number of unit distance interactions. Note that there are a large number of configurations with this same energy but with different paths followed by the bonds. (b) A higher energy, local minimum configuration. (c) A knotted configuration.



Crossover

Crossover point

Parent 1:	0 0 0 0 0		0 0 0
Parent 2:	1 1 1 1 1		1 1 1
Child 1:	0 0 0 0 0		1 1 1
Child 2:	1 1 1 1 1		0 0 0

Mutation

0 0 0 0 0 0 0 0 \longrightarrow 0 0 0 0 0 1 0 0

Figure 2. An illustration of the genetic algorithm crossover and mutation operators. In the top panel, the bit strings for two parents are shown along with a randomly chosen crossover point. The gene for child 1 is formed by taking the bits to the left of the crossover point from parent 1 and those to the right of the crossover point from parent 2. Child 2's gene is made up of the complementary pieces of the parents' genes. In the bottom panel, an example of a single point mutation to a gene is shown. In this illustration, the genes contain only 8 bits, but in the computations described, the gene uses 10 bits for each angle. For example, in the 19 atom case, there are 17 angles and the gene contains 170 bits.

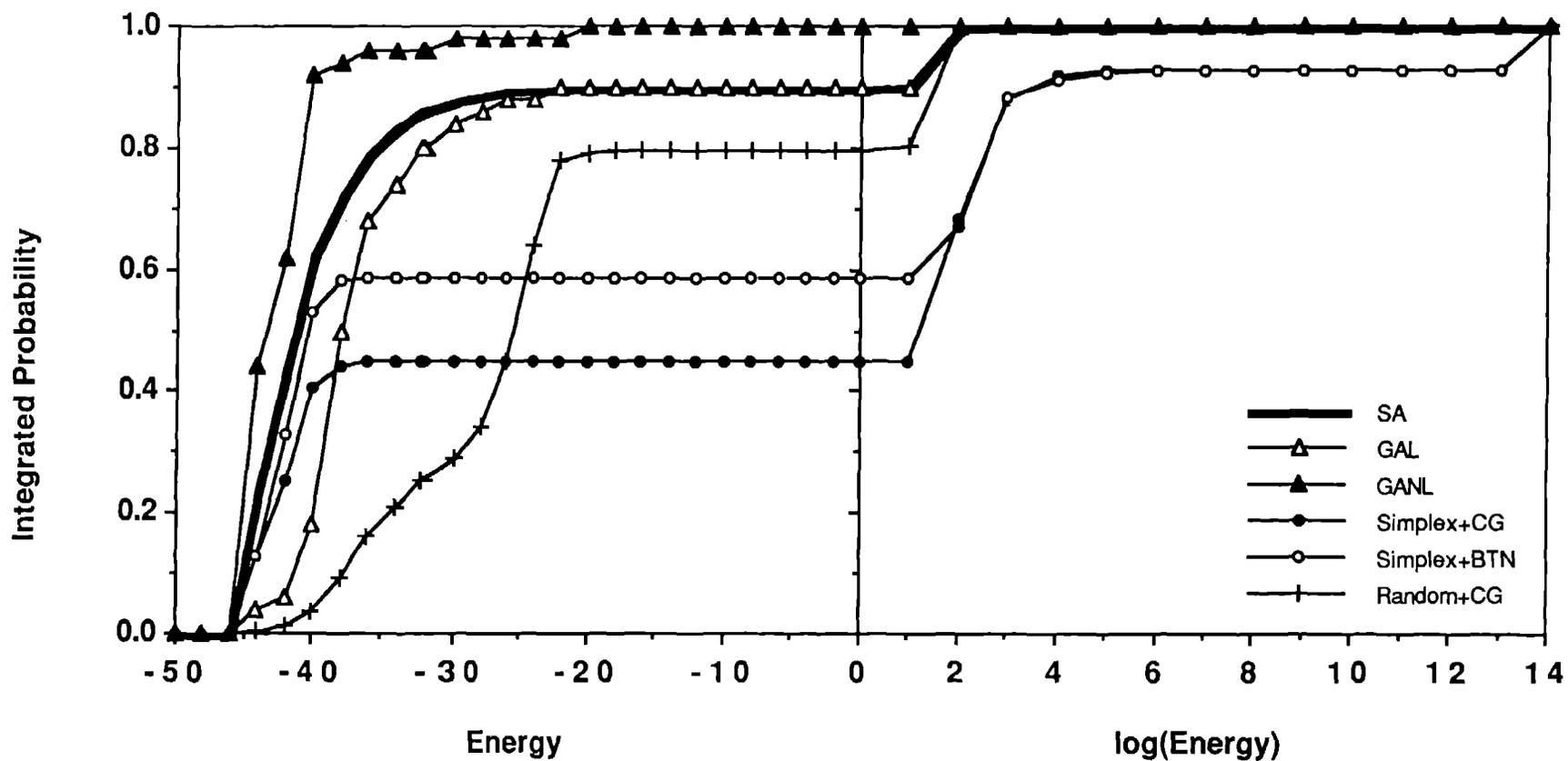


Figure 3. Integrated distribution of energies found for the 19 atom case. All unknotted configurations have energy less than zero and all knotted configurations have energies greater than zero. Note that the energies greater than zero are plotted on a log scale. The abbreviations in the key are SA (simulated annealing), GAL (Lamarckian GA), GANL (non-Lamarckian GA), simplex+CG (simplex followed by non-linear conjugate gradient), simplex+BTN (simplex followed by block-truncated Newton) and random+CG (non-linear conjugate gradient starting from randomly chosen initial points).

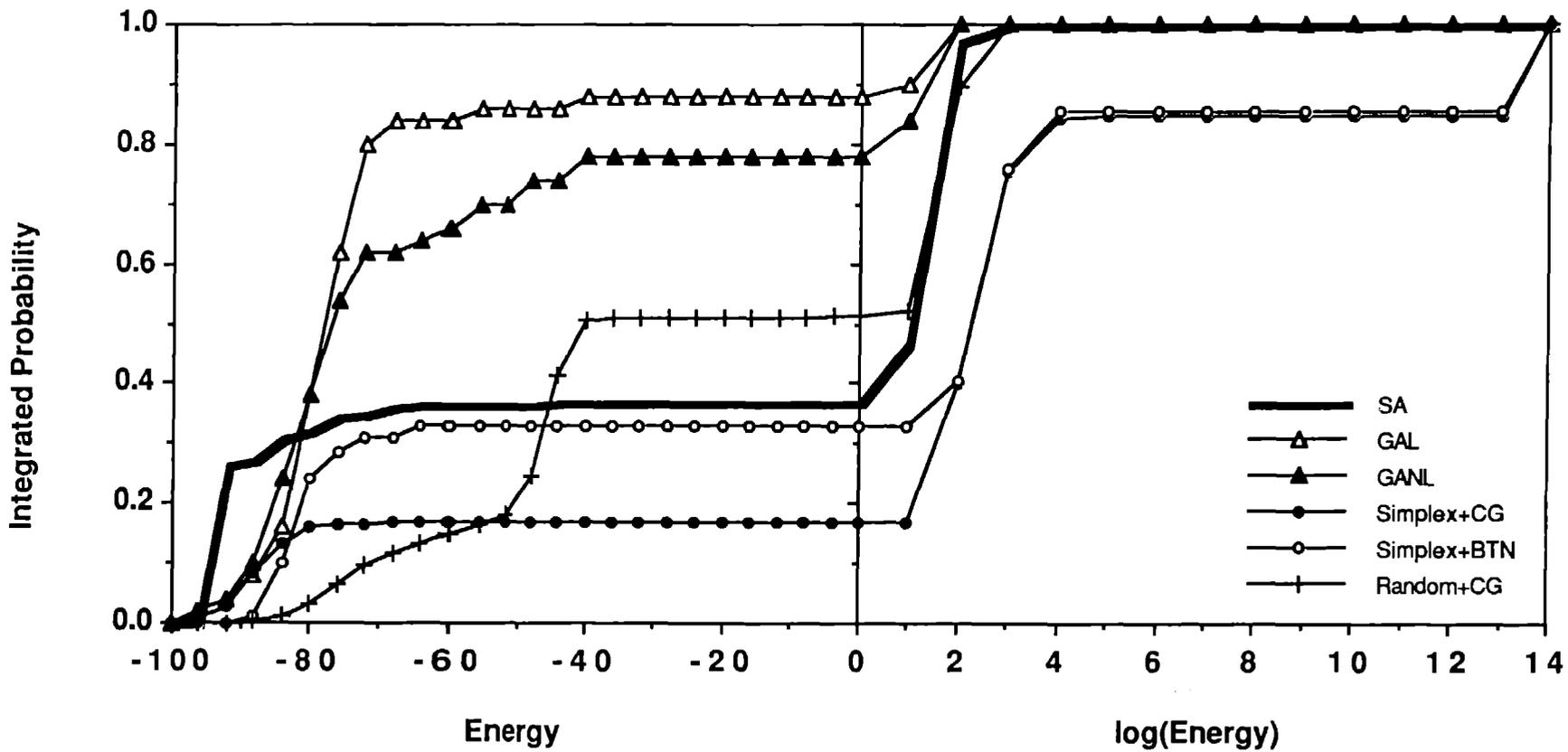


Figure 4. Same as Figure 3 except for 37 atoms.

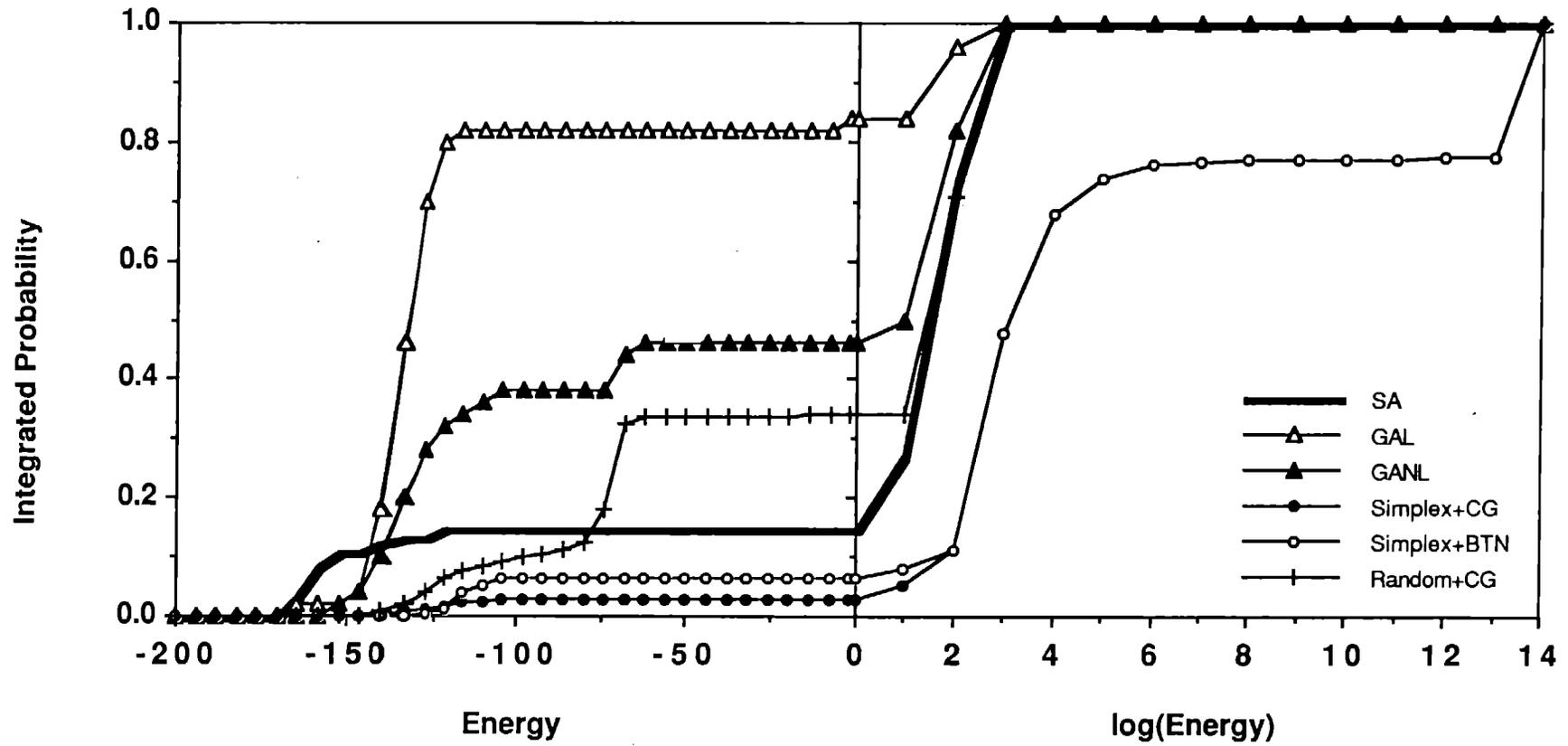


Figure 5. Same as Figure 3 except for 61 atoms.

UNLIMITED RELEASE
INITIAL DISTRIBUTION

Richard Gillian
Department of Chemistry
University of California, San Diego
9500 Gilman Drive
La Jolla, CA 92037
(619)534-0290

Roberta Susnow
Princeton University
Department of Chemistry
Princeton, NJ 08544

Prof. Herschel Rabitz
Princeton University
Department of Chemistry
Princeton, NJ 08544

Dr. John Grefenstette
Navy Center for Applied Research In Artificial Intellegence
Code 5514
Naval Research Laboratory
Washington D.C. 20375-5000

Dr. Adi Treasurywala
Sterling Drug Inc.
81 Columbia Turnpike
Rensselaer, NY 12144

Dr. Edward P. Jaeger
Sterling Drug Inc.
81 Columbia Turnpike
Rensselaer, NY 12144

Dr. Arup Ghose
Sterling Drug Inc.
81 Columbia Turnpike
Rensselaer, NY 12144

Dr. Paul Kowalczyk
Sterling Drug Inc.
81 Columbia Turnpike
Rensselaer, NY 12144

Dr. John Wendoloski
Sterling Drug Inc.
9 Great Valley Parkway
Malvern, PA 19355

Dr. Li Hou
Sterling Drug Inc.
9 Great Valley Parkway
Malvern, PA 19355

Dr. Ray Saleme
Sterling Drug Inc.
9 Great Valley Parkway
Malvern, PA 19355

Dr. Tim Mattson
Scientific Computing Associates
246 Church Street
suite 307
New Haven, CT 06510

Donald Frederick
111 West Pennsylvania Ave.
San Diego, CA 92103

David Levine
Building 203, MCS, C-245
Argonne National Laboratory
9700 South Cass Avenue
Argonne, Illinois 60439

Ann Huffer
1686 Beasley Drive
Deland Florida 32720

Dr Paul Weiner
Amber Systems Inc.
28 Tower Street
Somerville, MA 02143

Shankar Subramanian
4061 Beckman Institute
405 N. Matthews Ave.
Urbana, IL 61801

Tom Ngo
Department of Chemistry
Harvard University
Cambridge, MA 02138

Ben Halpern
5866 Rainbow Drive
San Jose, CA

Bill Cannon
Department of Chemistry
University of Houston
Houston, TX 77204

Richard Brewer
Physics Department
590 Commonwealth Drive
Boston University
Boston, MA 02215

Djamal Bouzida
Physics Department
590 Commonwealth Drive
Boston University
Boston, MA 02215

Jeff Blaney
Chiron Corp.
4560 Horton St.
Emeryville, CA 94608

8000 J.C. Crawford

Attn: 5300 E.E. Ives
8300 P.L. Mattern
8400 R.C. Wayne
8500 P.E. Brewer
1900 D.L. Crawford

8200 R.J. Detry

Attn: 8240 C.W. Robinson
8270 R.C. Dougherty

8210 W.D. Wilson

8210 J.M. Harris

8210 M. Pendley

8211 J.C. Meza (50)

8211 M. Colvin (15)

8211 D.S. McGarrah

8211 C. Tong

8211 C. Janssen

8211 R.S. Judson (10)

8211 Daryl Gutierrez

8353 C.F. Melius

8354 M. Koszykowski

8354 W. A. Glauser

8535 Publications Div./ Central Technical Files, 8524-2

3141 Technical Library Processes Division (3)

8524-2 Central Technical Files (3)