

Spring 1998

The structural optimization of atomic and molecular microclusters using a genetic algorithm in real-valued space-fixed coordinates

John Arthur Niese

University of New Hampshire, Durham

Follow this and additional works at: <https://scholars.unh.edu/dissertation>

Recommended Citation

Niese, John Arthur, "The structural optimization of atomic and molecular microclusters using a genetic algorithm in real-valued space-fixed coordinates" (1998). *Doctoral Dissertations*. 2030.
<https://scholars.unh.edu/dissertation/2030>

This Dissertation is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

**THE STRUCTURAL OPTIMIZATION OF ATOMIC
AND MOLECULAR MICROCLUSTERS USING A
GENETIC ALGORITHM IN REAL-VALUED
SPACE-FIXED COORDINATES.**

BY

John A. Niese

BS Chemistry, Washburn University, 1991

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Chemistry

May, 1998

UMI Number: 9831963

UMI Microform 9831963
Copyright 1998, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

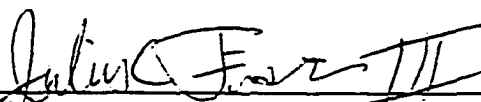
This dissertation has been examined and approved.



Dissertation director, Howard R. Mayne
Professor of Chemistry



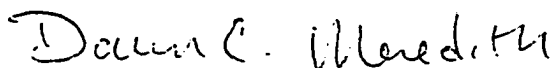
N. Dennis Chasteen
Professor of Chemistry



Julius C. Fister, III
Assistant Professor of Chemistry



Olof Echt
Associate Professor of Physics



Dawn C. Meredith
Associate Professor of Physics

4/17/48

Date

DEDICATION

This work is lovingly dedicated to the most important people in my life:

My wife, daughter, sister, and parents.

Their sacrifices have made this accomplishment possible.

ACKNOWLEDGEMENTS

I wish, first and foremost, to acknowledge and to thank my advisor, Professor Howard R. Mayne. His guidance and patience have enabled me to complete this degree. I consider myself privileged to have been introduced to the world of research by Professor Mayne. He treated me as a peer from the beginning of our relationship and I count him today as one of my dearest friends.

I also want to acknowledge and to thank my parents, John E. and Elaine C. Niese, and my sister, Ann L. Kaeding, for helping me navigate our shared trials and tribulations. You never stopped loving, supporting, and encouraging me, even when I was unable to encourage myself. I am forever grateful.

To my wife, Janelle A. Niese, and my daughter, Elizabeth H. Niese, I wish to express my deepest thanks and acknowledge the many sacrifices you both have made over the past years. I could not have completed this work without the support, optimism, and love you bring to my life.

I wish to acknowledge the financial, professional, and personal, support provided to me by the Department of Chemistry Faculty and Staff. I have greatly enjoyed my time as a Graduate student in your Department. Furthermore, I am indebted to each of you for sharing your skills and knowledge. I hope I may contribute as much to my students as you have to me.

I have developed many close friendships while here. In particular, I wish to acknowledge Jackson Barrett and John Beauregard, both senior “Maniacs” when I arrived in Professor Mayne’s research group, who started my coding career on the old VAX machine and helped me to avoid many of the pitfalls of graduate school. We will ski together again. Thanks for everything.

Finally, I wish to acknowledge and thank Jamie Clothier, Dan and Darlene Hill, Siobhan Milde, Todd Grimmer, and Ron White for their support and companionship which continues to this day. Each of you contributed greatly to this work, particularly by reminding me of what is truly important in this life.

TABLE OF CONTENTS

DEDICATION	<i>iii</i>
ACKNOWLEDGEMENTS	<i>iv</i>
LIST OF TABLES	<i>vii</i>
LIST OF FIGURES	<i>viii</i>
ABSTRACT	<i>ix</i>

CHAPTER	PAGE
INTRODUCTION	1
<i>I.</i> LENNARD-JONES ATOMIC CLUSTERS	9
<i>II.</i> SILICON CLUSTERS	31
<i>III.</i> THE GA OPERATORS FOR ATOMIC AND MOLECULAR CLUSTERS	44
<i>IV.</i> BENZENE, NAPHTHALENE, AND ANTHRACENE	71
<i>V.</i> CONCLUDING REMARKS	88
BIBLIOGRAPHY	91
APPENDICES	95

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
1.	Times needed to minimize $(LJ)_n$ by the SFMGA.	24
2.	Best fit parameters to data in Figures 2-4.	26
3.	Statistics for unseeded runs of $(Si)_n$.	39
4.	Statistics for seeded runs of $(Si)_n$.	41
5.	Number of times GM located for $(LJ)_{13}$.	59
6.	Performance of operators for $(LJ)_{13}$.	60
7.	Summary of operator's performance, $(LJ)_{13}$.	61
8.	Number of times GM located for $(LJ)_{19}$.	62
9.	Performance of operators for $(LJ)_{19}$.	63
10.	Summary of operator's performance, $(LJ)_{19}$.	64
11.	Fitness function varied.	65
12.	Summary of performance for $(H_2O)_n$, $n=2-8$.	66
13.	Number of times GM located for $(H_2O)_8$.	67
14.	Cartesian coordinates for DB benzene dimer.	78
15.	Exp-6-1 potential parameters.	80
16.	Aromatic hydrocarbon potential energies.	81

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.	CPU time -vs- cluster size.	27
2.	CPU time -vs- cluster Size, unseeded.	28
3.	CPU time -vs- cluster Size, seeded clusters.	29
4.	CPU time -vs- cluster Size, overall.	30
5.	Structures for the (Si) _n cluster, n = 3-10.	42
6.	Fittest structure as a function of generation.	43
7.	Fitness -vs- potential.	68
8.	Geometry of (H ₂ O) _n structures, n = 2-8.	69
9.	Geometry of (H ₂ O) _n structures, n = 9-13.	70
10.	DB benzene dimer geometry.	79
11.	Geometries for naphthalene clusters.	83
12.	Geometries for anthracene clusters.	84
13.	The average best SFMGA potential for (Ben) _n .	85
14.	The average best SFMGA potential for hydrocarbons.	86
15.	The average best SFMGA potential for (Ben) ₆ .	87

ABSTRACT

THE STRUCTURAL OPTIMIZATION OF ATOMIC AND MOLECULAR MICROCLUSTERS USING A GENETIC ALGORITHM IN REAL-VALUED SPACE-FIXED COORDINATES.

by

John A. Niese

University of New Hampshire, May, 1998

This dissertation documents the development and application of the space-fixed modified genetic algorithm, SFMGA. The SFMGA is shown to be both portable and fast for the structural optimization of Lennard-Jones, silicon, water, benzene, naphthalene, and anthracene microclusters.

We introduce the SFMGA and apply it to LJ atomic clusters. CPU times needed to obtain the global minimum are compared with similar methods. We then investigate a complicated potential representing silicon atoms. The results show that SFMGA is applicable to non-pairwise additive potentials.

We demonstrate the use of SFMGA for clusters where the monomers are molecules. Water clusters are optimized and the relative performance of the genetic operators, for both LJ and H₂O clusters, is explored. Finally, we investigate benzene, naphthalene, and anthracene clusters. In these clusters the size and potential surface complexity can be varied independently.

INTRODUCTION

We have developed a successful structural optimization routine for atomic and molecular clusters, the Space-Fixed Modified Genetic Algorithm, or SFMGA. The SFMGA routine and several of its applications are presented in this dissertation. We begin by discussing optimization in general.

It is usually obvious how to plan a day's route to complete a given set of errands in the shortest amount of time. Less obvious, however, is how to route a city's buses to minimize the miles driven by the buses, within the constraints of the schedule. Similarly, computer chip manufacturers want to place as many circuits as possible upon a given computer chip, but how to do so is not simple or obvious.

Optimization procedures or routines seek to find the "best" solution to a problem. The problem must be well defined and any candidate solutions must be able to be objectively ranked. The appropriate ranking for the city bus problem might be total distance traveled per day - the less miles the better - while comparing circuits per area would allow one to rank different chip layouts. There are surely several satisfactory bus routes, meaning the routes have similar

“total distance traveled” values, and many chip designs with similar densities. Any useful optimization routine must be able to find the best, or global, solution despite the presence of many similarly-ranked local solutions.

In chemistry, the closely-spaced solutions may be the minima in a potential energy surface. An optimization routine must discern between the tremendous number of minima which are present in most potential energy surfaces. Furthermore, there are high barriers to be overcome that exist between many of the minima. These features make the structural optimization of chemical systems extremely challenging. This work develops and implements a structural optimization routine for atomic and molecular clusters. The appropriate objective function is the cluster potential energy.

Clusters or microclusters are small gas phase aggregates of atoms or molecules bound by strong covalent forces or weaker van der Waals forces. Their small size gives clusters a common characteristic: They all have high surface particle to total particle ratios. The ground state structure of an atomic or molecular microcluster is often the geometry which has the lowest potential energy. Even when this is not the case, for systems containing heavy atoms, the lowest-lying local potential energy minima will be important in determining the thermodynamics at low temperatures. The search for the global minimum in a function which possesses many local minima is common to many areas of science

and technology. Well known examples in the chemical literature, besides the minimization of cluster potential energies, include conformational analysis of organic molecules¹⁻³ and native structures of proteins.⁴

There is a large variety of techniques available for obtaining global minima. Some which have been applied to cluster geometries are briefly reviewed here. One technique is the DC/GOP approach, which uses the properties of convex functions in a strategy similar to that used in linear programming.⁵ A more physically-motivated approach⁶ is to use crystal lattice structures as a starting point, and then relax the finite cluster of interest. Unfortunately, there is no guarantee that the bulk structure and cluster structure are closely related.

Various methods based on the diffusion equation have been proposed.⁷ The idea here is, roughly speaking, to begin with a broad distribution of points on a potential function, then run the diffusion equations backward in time until the global minimum is located. These methods can equivalently be viewed as a continuous removal of a smoothing function applied to the potential, with the goal of mapping the minimum of a broad featureless potential adiabatically onto the true global minimum as the smoothing is removed. Related methods include solving the Schrödinger equation in imaginary time together with a gradual reduction of the size of Planck's constant. The ($i\tau \rightarrow \infty$, $\hbar \rightarrow 0$) limit is a delta

function in the deepest minimum. This method has been referred to as “quantum annealing”,⁸⁻¹⁰ and is related to quantum (or diffusion) Monte Carlo.¹¹ Some of these and other related methods have been reviewed in an elegant pedagogical article.¹² They have all performed well on relatively small atomic clusters, but extension to larger or to molecular systems is not straightforward.

A rather different strategy for global minimization is to use stochastic methods. These methods are less likely than deterministic derivative-driven approaches to become trapped in local minima. However, there are no guarantees that they will converge to the global minimum in a finite number of steps. Among the techniques already used in this field are simulated annealing,¹³⁻¹⁵ J-walking,¹⁶ and pivot methods.¹⁷ The last two methods are related to simulated annealing. In simulated annealing, the potential is sampled by a Metropolis Monte Carlo¹⁸ walker at a fixed temperature for a long enough time that it can equilibrate. The temperature is then lowered, and the system re-equilibrated. In the limit of zero temperature and infinite time, the system should have reached the global minimum. The practical difficulty is to find a cooling schedule that is sufficiently slow to ensure convergence in finite computer time.

Another stochastic method which has been the subject of intense activity in recent years is that of genetic algorithms (GA).¹⁹⁻²¹ The genetic algorithm approach is based on concepts of Darwinian evolution. In this approach a population of candidate geometries (individuals) is maintained. Each individual (geometry) in the population is formed by “encoding” the physical coordinates of the problem into a numerical string – the “genotype” – which is then manipulated by “genetic operators”. Each individual is assigned a fitness based on that individual’s potential energy: The lower the potential energy, the fitter the individual. The average fitness of the population is changed by allowing the more fit individuals to exchange genetic materials with each other. This is usually done by selecting two parents based on their fitness, and allowing them to “mate”. The offspring produced contain some genetic material from both parents. The offspring are then included in the gene pool. If, on average, the offspring are more fit than the parents, then the quality of the gene pool will increase. The goal is to achieve the fittest possible individual. It is hoped that this individual is the geometry of the global minimum energy. The (traditional) GA approach for problems other than microcluster minimization has been reviewed.²²

In so-called “traditional” GA (TGA) the coordinates of each individual are coded into binary (base-2) numbers. Genetic information is exchanged between individuals primarily through a one-point crossover (see Appendix A) between

two individuals, with an occasional mutation of one bit into its complement. However, several GA approaches for locating global energy minima in atomic and molecular clusters have used real-valued coding of the coordinates. We use some form of space-fixed (base-10) coordinates for each of the monomers in the cluster. For instance, for three atoms, the physically meaningful variables are the three atom-atom distances; the simplest space-fixed (SF) coordinates are the nine Cartesian components of the three atoms. However, consider the ten-atom case, which has 30 SF coordinates as opposed to 45 internal coordinates. The size of the problem increases linearly with the number of particles in SF coordinates, whereas the number of internal coordinates grows quadratically.

Initially, the use of SF coordinates was rather controversial. One of the strongest theorems in the GA literature – the “schema theorem” or the “building block hypothesis”¹⁹ – predicts rapid improvement of the fitness under favorable conditions. However, the theorem holds only if the positions of the coded bits, the schemata, are meaningfully related to the physical context of the problem. While this is the case for internal coordinates, it clearly does not hold for SF coordinates, where an infinite set of Cartesian coordinates can map onto a single Euclidean distance. However, we will see that this “counterintuitive” approach is able to cope with considerably larger systems than those which have used binary coding in “meaningful” coordinates.

The disadvantages of internal coordinates are evident in an early paper by Hartke²³, where he obtained the global minimum for $(\text{Si})_4$ (which has a complicated, non-pairwise additive potential) using TGA at the expense of a very restrictive choice of internal coordinates. He was able to extend this TGA treatment to larger clusters,²⁴ but only by using the $(\text{Si})_{n-1}$ structure to “seed” the $(\text{Si})_n$. Our study of this silicon system, in real-coded space-fixed coordinates, is presented in Chapter II.

Many of the GA studies which have been conducted to date have used Lennard-Jones clusters as a test case.²⁵⁻³⁰ A review of these provides further insights into the various approaches currently used in the field. A landmark study was that of Zeiri,²⁹ who introduced real coding of the coordinates to the field of cluster minimization. Also innovative was his proposal of an array of genetic operations to transfer information between these genotypes. He used an array of crossovers, arithmetic and geometric averaging, and inversion on his real strings. While crossover and inversion have binary analogs, the averaging operators are a novelty not available for binary-encoded genotypes. Zeiri's calculations were actually on $\text{H}_2(\text{LJ})_n$. He found the GA to be at least as successful as simulated annealing³¹ in locating the global energy minimum for n as large as twelve.

We present here the development and application of a real-valued space-fixed modified genetic algorithm, SFMGA. Chapter I introduces the SFMGA and applies it to LJ atomic clusters. CPU times needed to obtain the global minimum are compared with similar methods. Chapter II applies SFMGA to a complicated potential representing silicon atoms. The results show that SFMGA is applicable to non-pairwise additive potentials. Chapter III introduces the use of SFMGA for clusters where the monomers are molecules. Water clusters are optimized and the relative performance of the genetic operators, for both LJ and H₂O clusters, is explored. In Chapter IV we investigate benzene, naphthalene, and anthracene clusters. In these clusters the size and potential surface complexity can be varied independently. Chapter V contains concluding remarks.

CHAPTER 1

LENNARD-JONES ATOMIC CLUSTERS

There has been some discussion on the best choice of coding to use in Genetic Algorithm optimizations. Earlier GA-based approaches to cluster geometry optimization have used the “standard GA” to search the geometry space. In the standard GA, variables are coded as binary (base-2) bit strings, and the operations are carried out on these strings. According to Goldberg^{19,32} the “alphabet” should be of as low a “cardinality” as possible. That is, the number of possible characters used to convey genetic information should be as small as possible; clearly the binary system fulfills this requirement best.

However, as the number of variables in a problem becomes large, the cost associated with the low cardinality of the binary alphabet may become prohibitive³². For instance, suppose the problem depends on N real variables. Roughly speaking, for a worst case scenario, an exhaustive search through these variables requires on the order of N^2 operations. On the other hand, if each of these reals is translated into a binary number (say eight bits), there are now $(8N)$ digits to deal with, and the exhaustive search is now through a space of $(8N)^2$.

Unless the search is much more efficient in the binary space, the penalty for binary coding for large N may be significant.

In considering structures such as clusters, there is another important consideration. That is, the choice must be made as to how best to represent the geometries. For instance, in an early paper, Hartke²³ used the "standard GA" to find the minimum geometry of Si₄. He used a coordinate system which was carefully chosen to be as separable as possible. (Here "separable" means, roughly, the ability of a coordinate to be approximately minimized independent of other coordinates.) In the language of GA, such coordinates describe isolated building blocks, which can be either well adapted or not well adapted, and are therefore relatively clearly related to the fitness. In fact, Hartke states²³ "Straightforwardly taking the Cartesian or internal coordinates ... does not work", and suggests that the coordinates must represent "small building blocks". This is in accord with the "principle of meaningful building blocks"¹⁹ central to GA theory.

However, as a multivariable problem increases in size, and the solution's dependence on the variables becomes increasingly complex, it becomes much more difficult to separate the variables. This, of course, is why normal modes are introduced into the discussion of vibrational motions of molecules; the normal modes are suitable combinations of interatomic distances which are

separable for low-amplitude excursions from the global minimum.

Unfortunately for the problem at hand, the calculation of such coordinates presupposes knowledge of the global minimum.

A further difficulty is that a separable representation especially chosen to be appropriate to any particular n -atom cluster is not easily generalized to other cluster sizes. Furthermore, even in the case of Si_4 mentioned above²³ the chosen coordinates spanned a restricted search space. It is clear from a later paper by the same author that selection of such candidate coordinate systems for larger clusters is problematic²⁴.

Zeiri, on the other hand, employs the real (base-10) Cartesian Space-Fixed (SF) coordinates for as many as fourteen atoms as the individuals in his nontraditional GA-based scheme²⁹. In this work, the structure of $\text{H}_2(\text{LJ})_n$ clusters was obtained without using the local minimization approach proposed by Gregurick et al.³⁰

We find that straightforwardly using SF coordinates is tempting. In an atomic cluster containing n atoms, for instance, there are $\frac{n}{2}(n-1)$ interatomic distances needed to describe the geometry. By comparison, there are always $3n$ space-fixed coordinates. Thus, for $n > 7$, fewer coordinates are needed to describe the cluster in SF coordinates than in internal coordinates. If such additional

coordinates as dihedral angles are needed, the number of internal coordinates increases, whereas such coordinates can always be simply obtained from the SF data. For the sake of evaluating the performance of the algorithm, the number of coordinates required in the SF system is $O(n)$, whereas that required in the internal coordinates is $O(n^2)$.

Clearly, then, for larger systems, the search space is smaller ($O(n^2)$) for SF coordinates than it is for internal coordinates ($O(n^4)$). However, the question now becomes that of the efficiency of the search procedure in a space where individual points in that space (e.g. the z coordinate of the i^{th} atom) are not directly related to the potential energy function.

Zeiri²⁹ finds his results are at worst competitive with those obtained using simulated annealing. Furthermore, use of the Cartesian coordinates provided portability between cluster sizes and required no restriction of the search space. However, the representation in SF coordinates is plainly contrary to the spirit of Goldberg's "building block hypothesis"; none of the coordinates stands alone as a meaningful building block¹⁹. Furthermore, many of the variables are interchangeable by symmetry. Can this representation be used efficiently with a Genetic Algorithm approach? While Zeiri has enjoyed success with it, there is no direct timing comparison with other GA approaches available for the system he has chosen. The purpose of this chapter is to systematically explore the viability

of using the SF coordinates, and to compare with benchmark calculations³⁰ on $(\text{LJ})_n$ clusters using more traditional coordinates.

In their paper, Gregurick, Alexander and Hartke³⁰ proposed a global geometry optimization technique using a modified Genetic Algorithm approach for clusters. They refer to their technique as a deterministic/stochastic genetic algorithm (DS-GA). In this technique, the stochastic part is a traditional GA, with the manipulations being carried out on binary-coded internal coordinates (atom-atom distances). The deterministic aspect of their method is the inclusion of a coarse gradient descent calculation when assigning the relative fitness of each geometry. However, they did not use the resulting geometry of the gradient descent as an individual in the population. Tests of this technique show it is vastly more efficient than searches without this local minimization.³⁰ They report geometries for clusters of up to $n = 29$ LJ atoms, and find that their computer time scales as $O(n^{4.5})$.

We investigate here the feasibility of using the SF (base-10 coded) coordinates in a GA-inspired optimization technique, which incorporates a local minimization of the candidate solutions with the subsequent use of these minimized geometries during breeding, not merely for assigning fitness.

For the cluster potential energy, we use a pairwise-additive Lennard-Jones potential:

$$V(r) = 4\epsilon \sum_{i=1}^{n-1} \sum_{j>i}^n \left(\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right)$$

Where r_{ij} is the distance between any two atoms. The values of ϵ and σ used³³ are 0.0123 eV and 3.36 Å, respectively. Cluster energies are often reported in reduced energy units, $\bar{V} = \frac{V}{\epsilon}$. A potential of this form has a dimer equilibrium distance $r_e = (2)^{1/6} \sigma$.

Each individual, X_i , in the population to be evolved consists of the real (base-10) SF Cartesian coordinates of each of the n atoms in the cluster: $X = (x_1, y_1, \dots, z_n)$. We choose the number of individuals in the population to be typically 10 or 20. Initially, the coordinates are randomly chosen within a box of size L^3 in the first octant. We take $x_1 = L\zeta$, etc., where ζ is a freshly-generated random number between 0 and 1. We have used $L = \sqrt[3]{6n} \sigma$, where n is the number of atoms, for this work. A conjugate gradient minimization³⁴, (*Numerical Recipes* contains a useful CG routine) is performed every generation on each individual to place each structure in the vicinity of the "nearest" minimum. The conjugate gradient procedure is halted if any $r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2 \geq L^2$. Our choice of L^2 for the termination of a conjugate gradient minimization follows that of Gregurick et al.³⁰ We, as they, utilize this value to

prevent dissociation of a cluster during the local minimization. However, we also use L , as described above, when we randomly generate the individuals of a zeroth generation. It is unclear how Gregurick, et al.³⁰ generate their zeroth generation.

Given an individual (that is, a geometry) X_i , we can calculate the potential energy of the cluster for that geometry, $V_i = V(X_i)$. Given the set $\{V_i : i=1, N\}$ we can assign the fitness, f_i , of the i^{th} individual. We use the convention that the f_i are normalized to unity. An intermediate quantity, F_i , is evaluated by taking a function of V_i .

$$F_i = (V_{\max} - V_i) / (V_{\max} - V_{\min}) \quad i = 1, N$$

This form of F_i is known as the “range” fitness function. The quantities, V_{\max} and V_{\min} are $\max\{V_i\}$ and $\min\{V_i\}$ respectively. The values of f_i are then found by normalization:

$$f_i = \frac{F_i}{\sum_{i=1}^N F_i}$$

The next generation (the “children”) is formed from the current generation (the “parents”) as follows. First the best 20% (that is, those with the highest fitness) of the individuals in the current generation are passed intact to the next generation. (This is known as “elitism”¹⁹.) The remainder of the population in the next generation is obtained by use of genetic operators on the current generation. These are: 1. inversion; 2. geometric mean; 3. arithmetic

mean; 4. n-point crossover; 5. 2-point crossover. They are partially described elsewhere by Zeiri²⁹. We give the complete details in the Appendices. Of these operators, number 1 transforms one individual into a different individual; numbers 2 and 3 use two parents to “breed” one child; numbers 4 and 5 use two parents to produce two children. Consequently, if operators 4 or 5 are chosen to produce the last child, they actually produce two children: the last plus one extra. In this case, to ensure that we have N individuals in the population every generation, we choose to have 1 less individual carried over as elitist.

All operators are given the same weighting, $w_\alpha = 0.2$, for $\alpha = 1$ through 5. Following standard Monte Carlo practices¹⁸, a random number on [0,1] is generated, and used to determine the operator to be selected. The requisite parents (one or two depending on the operator) are then selected weighted by their fitness using fresh random numbers.

A typical run contains 10 or 20 individuals in a population. A run was terminated when either the (presumed^{6,30}) global minimum was found or the potential energy of the bestfit structure did not change for 5 generations.

We also implemented a seeding procedure³⁵. For an n-atom cluster, one atom is added to a globally minimized ($n-1$) cluster. This is carried out in the following manner. The Cartesian coordinates of a minimized ($n-1$) cluster were

previously saved in a data file after having its center of mass translated to the origin. As an $(n-1)$ cluster is read into our algorithm the distance of each atom from the origin is calculated. This determines the distance from the origin, r_{max} , of the furthest atom. The n^{th} atom is then randomly placed upon the sphere centered at the origin with $radius = r_{max} + r_e$. Furthermore, the seed $(n-1)$ structure is randomly rotated about its center of mass by generating 3 random Euler angles. The random Euler rotation generally prevents any of the seeded $(n-1)$ coordinates in each individual from being identical, since the same $(n-1)$ minimized structure is used each time. The n -atom cluster then undergoes a local minimization after which its center of mass is placed at $(x, y, z) = (\frac{L}{2}, \frac{L}{2}, \frac{L}{2})$. This process is applied, with fresh random numbers, to each individual as the zeroth population is created. All coordinates were allowed to freely evolve during the initial local minimization and in any subsequent generations.

RESULTS AND DISCUSSION

The results for the minimization of $(LJ)_n$ clusters, $n=[4-29]$ without seeding and $n=[5-55]$ with seeding, are shown in Figure 1. We plot CPU time as a function of cluster size, n . Each result is the best successful run of at most 20 independent runs. In all cases, the global minimum was found to ± 0.001 reduced energy units⁶, usually within the first 5 runs. For the unseeded runs, we show

the best results for populations of 10 and 20 individuals. There was no systematic difference in convergence times for the two population sizes. We report the times for each population size in Table 1. Populations larger than 20 were attempted, but did not improve the times and are not reported here. We find that for larger clusters, seeding results in faster convergence.

To implement the seeding method we started with a nucleus of $n=4$ and built the cluster in increments of one each time, as explained previously. If seeding is used, the upper limit of cluster size which can be minimized using this technique has yet to be found. The CPU times, for both seeded and unseeded runs, (on a DEC 2100-500) are given in Table 1 and shown in Figure 1. It can be seen from the data presented that our implementation of the GA is able to find the minimum for a relatively large cluster "from scratch" (unseeded) in a reasonable time.

One of the purposes of this chapter is to compare the results using SF coordinates without binary coding with the results of Gregurick et al.³⁰ Since our calculations were carried out on a faster machine³⁶ than theirs, we have, for the purposes of comparison, multiplied our cpu times by a factor of 5.0 in all succeeding Figures. (We realize that comparison of CPU times is not straightforward. However, the operations involved in this work and the Gregurick et al. paper – potential evaluations, and their derivatives– seem to be

reasonably similar to those in the benchmark timing routines.³⁶) The CPU times for cluster minimization in which no seeding was employed are compared in Figure 2. The SF variant of the modified GA performs faster for all cluster sizes reported. Furthermore, Gregurick et al. report no converged results for $n \geq 20$.

In Figure 3, we compare the times needed to minimize a seeded cluster. The time reported is the CPU time to obtain the optimal energy for cluster size n given a minimized cluster of size $(n-1)$. It can be seen that the binary-coded approach (DS-GA) is comparable to the present work at low cluster size. However, as the cluster size (and the size of the search space) increases, the SF application becomes increasingly preferable.

One way of comparing numerical algorithms is to compare how they scale with the size of the problem. In order to obtain this measure, one plots $\log(t)$ against $\log(n)$ and obtains the best straight line fit. For the data in Figures 2 and 3, these scalings are given in Table 2. In both cases – seeded and unseeded – the SF approach fares better. Only if we compare the data at very low n values is the DS-GA performance comparable.

One interesting point to notice from Table 2 is that the seeding technique increases in efficiency as the second solvation shell is closed. The first solvent

shell closes at $n=13$, the second at $n=55$. Presumably this is due to the low number of available second shell sites for the added atom.

Perhaps a more reasonable measure of the performance of the seeding technique is to measure the cumulative time needed to minimize a cluster, $(LJ)_n$.

For our runs, we define this time as:

$$t_n^{cum} = t_4 + \sum_{i=5}^n t_i$$

where t_i is the time needed to minimize the i^{th} cluster starting from the $(i-1)$ structure if $i > 4$, or from scratch if no seeding was used, as in the case of $n=4$.

Gregurick et al.³⁰ report an overall scaling for their method. It appears that they used a similar definition of their cumulative time. We are able to reproduce their reported scaling law if we further define:

$$t_n^{cum} = t_{20} + \sum_{i=21}^n t_i$$

for their data. Results of t^{cum} as a function of n are given in Figure 4. The raw timings are given in Table 1.

Clearly the SF version of the DS-GA presented here is at worst comparable to, and usually superior to the DS-GA of Gregurick et al.³⁰ This is a rather surprising result in light of Goldberg's discussions¹⁹ of the greater efficiency of GA operations using binary-coded variables, and the "building block hypothesis".

It is unclear whether our enhanced efficiency is due to the base-10 coding itself or to the more complex operations applicable to base-10 variables. In an attempt to compare our procedure with the more traditional GA approaches, we have carried out comparison calculations on several cluster sizes. The most important operator in the "traditional" GA is the one-point crossover^{19,21}. We have carried out runs using our base-10 coding together with this single operator. We report here only the results for $n=19$, which we find to be typical. Using an initial population of ten individuals and only the one-point crossover operator, the global minimum was located only twice in a batch of one hundred independent runs. This compares poorly to the usual location of the global minimum at least once in ten runs, as we report here. When the population size was increased to one hundred individuals the probability of locating the global minimum increased to approximately 30%, but the CPU time of the best run also increased. It appears that the efficiency of our technique is related both to the choice of the real Space-Fixed variables, and to the use of appropriate operators to search the variable space.

We mention here some of the caveats concerning the seeding technique. For certain potential parameters, even such simple clusters as $(LJ)_n$ can undergo phase changes³⁷. In such circumstances, genetic information obtained for clusters of phase α may actually be detrimental for clusters of phase β . In

addition, there may be several families of morphology (particularly in bonded structures) in which there is little similarity between structures X_n and X_{n+1} , even for small n . We have observed this in silicon clusters³⁸ and report our findings in Chapter II.

CONCLUSIONS

We have presented calculations of global potential energy minimizations for Lennard-Jones clusters using a modified Genetic Algorithm approach. We have used the philosophy of the DS-GA of Gregurick et al.³⁰. However, we allowed each geometry created in the search to be immediately quenched to a local minimum instead of simply using the numerical result when assigning the structure's fitness. The technique presented here restricted the individuals in the populations to be the geometries of local minima. Thus, the search became a search through a finite (albeit large) number of individuals, rather than over an infinite set of possibilities. In further contrast to the approach of Gregurick et al., we used the atomic Space Fixed Cartesian coordinates directly as our genetic material. Since the Cartesians scale as only ($O(n^2)$), while internal coordinates

scale as $O(n^4)$, we realized an immediate reduction in the search space without introducing a (possibly³⁸) detrimental truncation of the space. This required the use of nontraditional genetic operators, which we adapted from the work of Zeiri.²⁹

We find the SF Cartesian version of the DS-GA with real coding is comparable to the DS-GA using internal coordinates with binary coding at low n . However, at high n , the SF version is superior. It is capable of minimizing clusters up to $n=29$ without any seeding. Using seeding, minimized clusters of $n=55$ were readily attainable. It was found that the CPU time required scaled as $O(n^{3.3})$.

Table 1. Times needed to minimize $(LJ)_n$ by the SF modified GA

The times are CPU times on a DEC 2100-500. Given are times in seconds for unseeded ($n=[4-29]$) and seeded approaches ($n=[5-55]$). The data for the seeded method are the times needed to minimize a cluster of size n starting with a minimized cluster of size $(n-1)$. Results are the best of ten or, at most, twenty independent runs. Figures in parentheses are the number of generations required for convergence. A zero implies convergence upon conjugate gradient minimization of the zeroth generation.

n	10 per pop	20 per pop	10/pop; seeded
4	0.049 (1)	0.103 (0)	NA
5	0.052 (0)	0.157 (1)	0.102 (1)
6	0.162 (4)	0.342 (3)	0.284 (2)
7	0.073 (1)	0.267 (1)	0.374 (2)
8	0.201 (2)	0.366 (2)	0.262 (1)
9	0.219 (1)	0.494 (2)	0.375 (1)
10	0.478 (1)	0.541 (1)	0.390 (1)
11	1.158 (1)	1.176 (1)	0.488 (1)
12	1.982 (1)	0.980 (1)	0.589 (1)
13	1.899 (2)	3.274 (2)	0.743 (1)
14	1.453 (2)	3.874 (2)	0.765 (1)
15	1.864 (2)	6.763 (2)	1.076 (1)
16	2.736 (2)	7.180 (2)	1.189 (1)
17	11.511 (3)	8.196 (1)	1.355 (1)
18	12.673 (5)	17.518 (4)	3.220 (4)
19	3.387 (2)	15.356 (3)	1.745 (1)
20	11.445 (2)	15.276 (2)	1.968 (1)
21	17.323 (4)	64.211 (6)	2.190 (1)

22	33.170 (3)	42.924 (2)	2.469 (1)
23	16.151 (3)	46.166 (4)	2.919 (1)
24	36.998 (4)	83.541 (5)	3.453 (1)
25	68.748 (6)	56.805 (4)	4.327 (1)
26	171.275 (10)	325.439 (16)	3.617 (1)
27	177.368 (9)	151.053 (5)	5.587 (5)
28	116.783 (20)	457.961 (29)	5.869 (1)
29	118.968 (16)	275.080 (15)	17.345 (1)
30			18.472 (5)
31			6.488 (5)
32			17.658 (5)
33			6.747 (1)
34			8.563 (1)
35			6.538 (1)
36			11.998 (5)
37			14.687 (1)
38			15.160 (1)
39			8.149 (3)
40			10.858 (2)
41			10.684 (4)
42			12.394 (3)
43			12.137 (2)
44			12.523 (1)
45			13.967 (1)
46			12.963 (1)
47			13.368 (1)
48			15.630 (1)
49			15.819 (1)
50			21.733 (1)
51			15.804 (1)
52			16.692 (1)
53			16.493 (1)
54			18.611 (1)
55			18.320 (1)

Table 2. Best fit parameters to data in Figures 2-4

It is assumed the data can be cast in the form $t \propto n^\gamma$. The value of γ is obtained using an unweighted linear least squares fit³⁹ to $\log(t)$ vs $\log(n)$. The integers in brackets [nmin,nmax] denote the range of cluster size over which the fit was taken. See text for explanation of the terms unseeded, seeded, and cumulative.

	DS-GA	This work
Unseeded	3.9 [4-20]	3.6 [4-20]
		4.4 [4-29]
Seeded	7.5 [21-29]	4.9 [21-29]
		3.3 [5-55]
		3.6 [17-41]
		2.2 [42-55]
Cumulative	4.5 [4-29]	3.2 [4-29]
		3.3 [4-55]

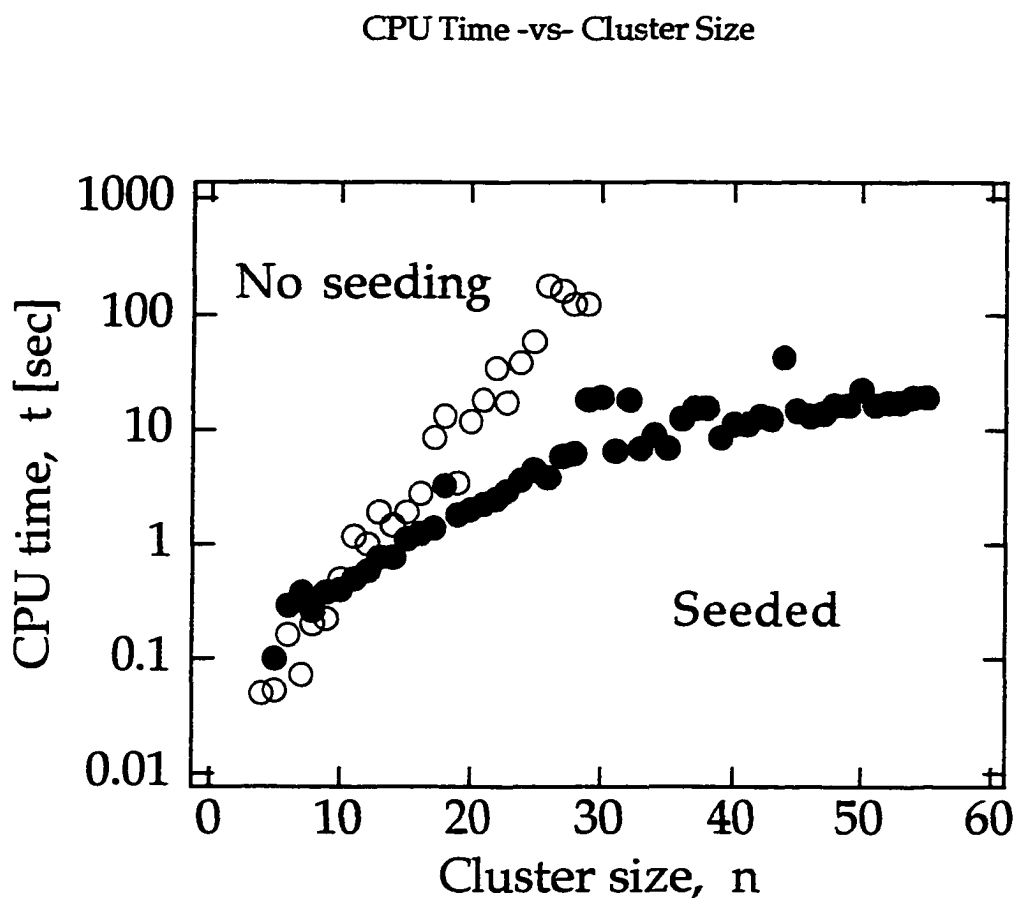


Figure 1. Plot of CPU time for global minimization of $(L)_n$ cluster as a function of cluster size. Open circles are unseeded calculations (see text); filled circles use the seeding technique described. Each time shown is the best successful result of ten independent runs. Note that the ordinate scale is logarithmic.

CPU time -vs- Cluster Size, unseeded

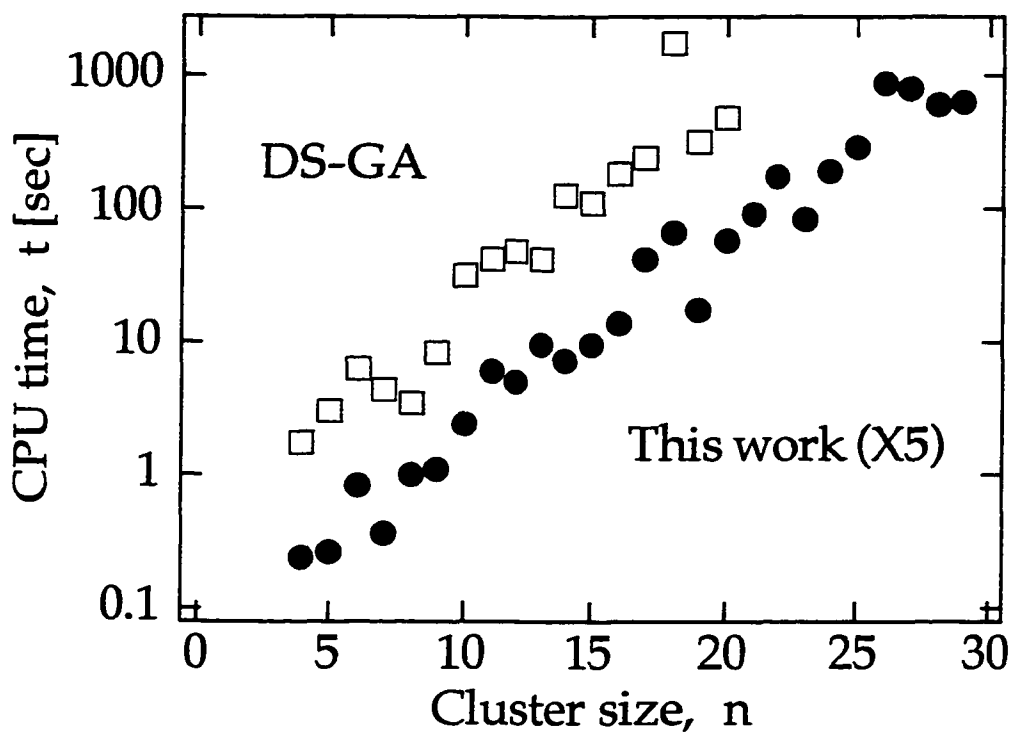


Figure 2. Plot of CPU time for unseeded clusters, using the DS-GA compared with present method. Open squares are the results of Gregurick et al.; filled circles are present method multiplied by a factor of 5.0 (see text). Note that the ordinate scale is logarithmic.

CPU time -vs- Cluster Size, seeded clusters

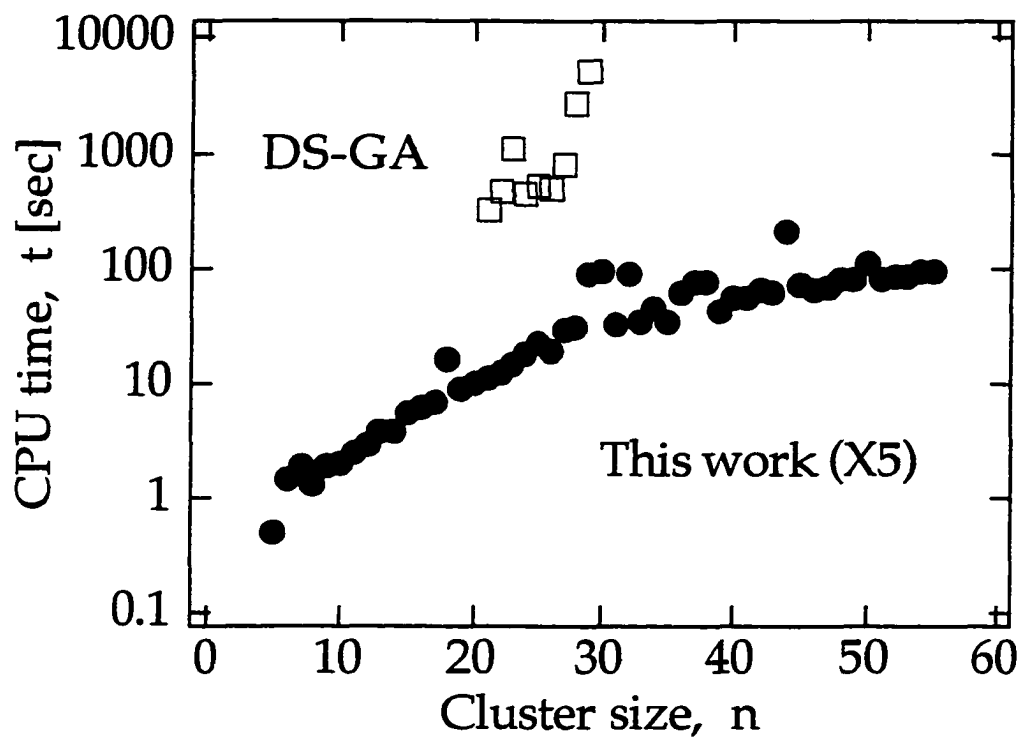


Figure 3. Plot of CPU time for seeded clusters, using the DS-GA compared with present method. Symbols are as in figure 2. Note that the ordinate scale is logarithmic.

CPU time -vs- Cluster Size, overall

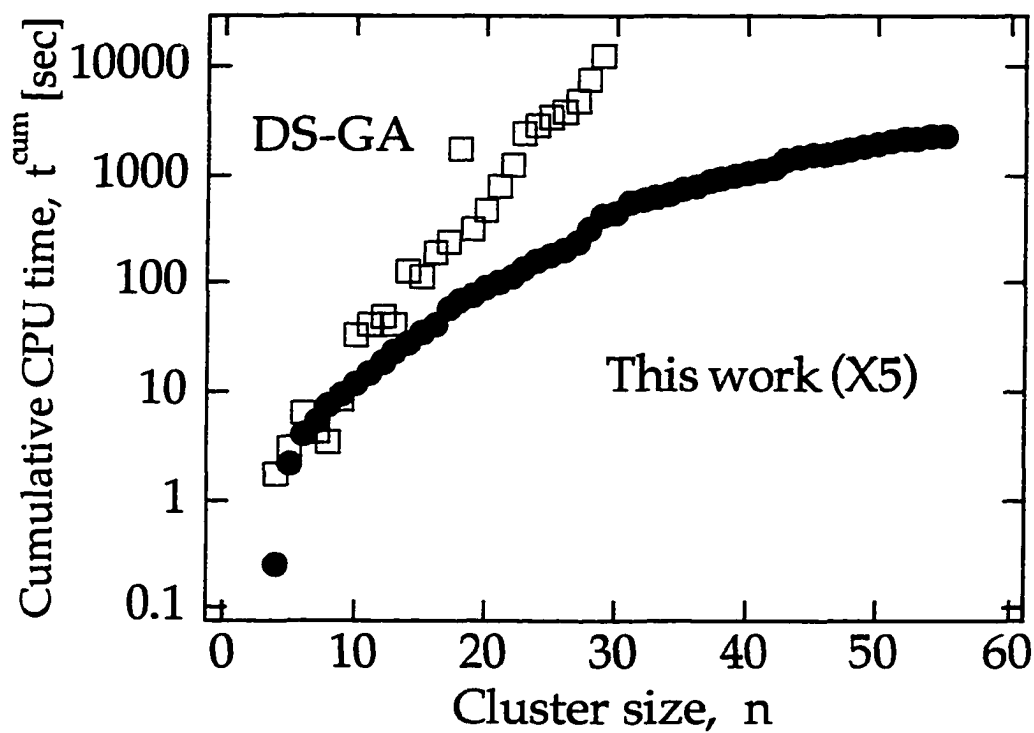


Figure 4. Plot of cumulative CPU time (see text) for minimization of clusters using the results of Gregurick et. al. compared with present method. Symbols are as in figure 2. Note that the ordinate scale is logarithmic.

CHAPTER II

SILICON CLUSTERS

Our main goal here is to test our approach with a more complicated potential. Additionally we investigate the approach of obtaining larger structures via a growth technique. In the simplest version of this method considerable effort is expended in finding the geometry of the smaller clusters. Typically, one atom is then added and the new structure found by minimizing the potential of this "seed+atom" hybrid. The progression to larger clusters is effected by repetition of this procedure.

The growth sequence of such simple structures as Lennard-Jones atomic clusters is known^{28,35,40}. Each additional atom usually finds a site on the outside of the seed cluster at which its coordination number can be maximized. However, the growth sequence for clusters governed by more complicated potential energy functions is less clearly understood. In particular, there may be abrupt changes in morphology between clusters of size n and those of size $(n+1)$. These changes have been reported for the silicon cluster potential employed here⁴¹.

Clearly, though, there are dangers in such approaches. A minimization technique should be able to explore the configuration space without restriction, if at all feasible. Seeding may "tempt" the routine to prematurely converge to a minimum similar to the seed structure. Thus, if the geometry of the global minimum of a potential is not known in advance, presumptions about the structure may lead to spurious results. Furthermore, searches which are guided by some symmetry restriction or in some coordinates which possess lower dimensionality than the true problem may not sample the full configuration space satisfactorily.

Each individual, X_i , in the population to be evolved consists of the SF Cartesian coordinates of each of the n atoms in the cluster: $X = (x_1, y_1, \dots, z_n)$. We choose the number of individuals, N , in the population (10 in this work). We have used $L = \sqrt[3]{3n} r_e$, where r_e is the dimer equilibrium distance, for this work. A conjugate gradient minimization is performed every generation on each individual to place each structure in the vicinity of the nearest minimum. The conjugate gradient procedure is halted if any interatom distance $\geq 1.2L$. We utilize this value to prevent dissociation of a cluster during the local minimization. The procedure used was that described in Chapter I, excepting the differences noted here.

These operators are: 1. inversion; 2. geometric mean; 3. arithmetic mean; 4. n-point crossover. We provide examples in Appendix A. Of these operators, number 1 transforms one individual into a different individual; numbers 2 and 3 use two parents to “breed” one child; number 4 uses two parents to produce two children. Consequently, if operator 4 is chosen to produce the last child, it produces two children: the last plus one extra. In this case, to ensure that we have N individuals in the population every generation, we choose to have 1 less individual carried over as elitist. The population size remains constant throughout the calculation.

All operators are given the same weighting, $w_\alpha = 0.25$, for $\alpha = 1$ through 4. A random number on $[0,1]$ is generated and used to determine the operator to be selected. The requisite parents (one or two depending on the operator) are then selected based on their fitness using fresh random numbers.

In order to locate the global minimum, twenty independent runs were carried out. A run was terminated when the potential energy of the bestfit structure did not change for 5 generations. In order to test the robustness and speed of the technique, we then carried out a further 100 runs. In these, the run was terminated either when the previously-obtained global minimum was reached, or when there was no improvement in the fittest individual for 5 generations.

All calculations were carried out on the Bolding-Andersen⁴¹ (BA) silicon potential. We note here that some energies reported here are not exactly those given in the BA paper. It appears that some of the parameters reported there were truncated in the paper, but not in the code. (Dr. B. C. Bolding, private communication). We have used the parameters as they appear in the BA paper.

RESULTS AND DISCUSSION

We show the minimum potential energy structures found using our method for $(\text{Si})_n$, $n=3-10$, in Figure 5. Also shown are the energies of these geometries. We find the same structure as do BA for $n=3,4,9,10$. However, we obtain lower-lying minima (that is, more stable geometries) than do they for the cases $n=5-8$. It must, of course, be emphasized that these are not guaranteed to be the true global minima for this surface. However, they do appear to be the lowest yet reported. These results demonstrate that the technique is capable of finding geometries with low-lying potential energies, and is capable of finding minima not seen in other studies.

In Table 3 we give computational statistics for minimization runs on each of the species, $(\text{Si})_n$, $n = 3-10$. All runs were initiated using random space-fixed

coordinates. It can be seen that the time required to locate the best minimum increases with cluster size. In addition, the number of generations required to converge also increases. By contrast, the percentage of the independent runs in which the global minimum is found decreases with cluster size. It can be seen from Table 3(b) that on several occasions the global minimum was found in the conjugate gradient descent of an initial random cluster from the zeroth generation. However, we have carried out calculations for enough sets of initial conditions that the efficiency of the GA part of the minimization has been thoroughly tested.

This is, of course, to be expected. For $(LJ)_n$ clusters, it has been reported⁴² that the number of local minima scales as $\exp(n^2)$. While the number of minima for the potential used here has not been determined, we must expect it to be extremely large, and to increase dramatically with cluster size.

In Table 4 we give comparable statistics on runs for $(Si)_n$ which were carried out using a seeding procedure. All runs were initiated with the minimized $(n-1)$ structure as a seed. The n^{th} atom was added as described above and elsewhere²⁵. By examining the column headed “%GM”, it can be seen that the probability of finding the global minimum actually decreases on seeding for $(Si)_7$ and $(Si)_9$. In addition, the “center of gravity” of all minima found is higher for these two cases, as evidenced by the worsening of the mean value of the

potential. It is encouraging, however, that the global minimum can still be located by SFMGA, even if the algorithm is forced to work harder.

We use a representative $(\text{Si})_7$ run to attempt to explain this phenomenon. As can be seen from Figure 5, the predicted structures (using the BA potential⁴¹) of $(\text{Si})_6$ and $(\text{Si})_7$ are rather different. The $(\text{Si})_6$ structure could be described as a double-corner-capped rhombus, whereas the $(\text{Si})_7$ structure is a double-edge-capped pentagon. In Figure 6, we show the fittest candidate solution (lowest energy geometry) for the 12 generations of a seeded GA run of $(\text{Si})_7$. The seed geometry is that of $(\text{Si})_6$ in Figure 5.

The bestfit structure of the zeroth generation has a geometry which resembles that of $(\text{Si})_6$, with the seventh atom capping one of the available corners of the rhombus. The pentagon present in the minimized Si_7 structure is, in some sense “four-fifths” completed. By the end of generation 1, the best structure to date is a more open, “chain-like” one which also contains the beginnings of a pentagon. By the end of generation 2 the pentagonal ring has begun to emerge. However, by the end of generation 4, the best geometries found once again have rhombus character. Rhombus-containing structures persist through generation 11. During the twelfth generation the routine finally converges to the double-edge-capped pentagonal minimum.

The move away from the pentagonal structure yields considerable insight into the way the GA functions. Candidates for the various operations are chosen in each generation by selection. While the pentagonal structure has the highest fitness, there are several individuals in the population which are almost as fit. These generally contain the rhombus structure, which was imprinted on the population by the choice of the seed, causing the rhombus-containing structures to dominate the gene pool. It is several more generations before the better genetic material contained in the pentagonal structures asserts itself, and the (presumed) true minimum is found. For both $(\text{Si})_7$ and $(\text{Si})_9$ it seems that the seeding procedure actually inhibited the optimization.

CONCLUSIONS

We have used the Space-Fixed modified GA (SFMGA) approach to obtain global minima for the silicon clusters $(\text{Si})_n$ using the Bolding-Andersen⁴¹ potential. One modification to the usual GA is the use of gradient-driven minimization of each geometry immediately after that geometry has been produced. Another feature of our method is the use of space-fixed atomic coordinates and the absence of binary coding in the GA.

We have shown that the SFMGA approach used here is capable of finding global minima for this potential. In fact, we report here new claimed global minima for $n=5-8$. The method is shown to be numerically robust, and relatively fast (although we have no data for direct comparison). As expected, the problem becomes more resource-intensive as the cluster size increases.

We have demonstrated the portability and convenience of the space-fixed coordinates, which have the advantage of spanning the full configuration space. We have also shown that, under certain circumstances, using seeding structures to generate new clusters can be detrimental. The approach we advocate is unbiased by any information on the sought outcome.

Acknowledgments

We thank Dr. Barry C. Bolding for the code for the potential energy function and its derivatives used here, and for useful correspondence.

Table 3. Statistics for unseeded runs of $(Si)_n$

The column heading %GM is the percentage of independent runs (out of 100) which located the global minimum reported here. V_{\min} , \bar{V} , V_{\max} denote the lowest, the mean, and the highest potentials in kcal/mol of all the individuals in all runs. The mean cpu time, \bar{t} (sec) and mean number of generations, $\bar{\Gamma}$, averaged over all runs are also given in Table 3a. In Table 3b minimum/maximum times and numbers of generations for only those runs which reached the global minimum are given. We use Γ to denote generation number.

Table 3a.

n	%GM	V_{\min}	\bar{V}	V_{\max}	\bar{t}	$\bar{\Gamma}$
3	100	-186.1	-186.1	-186.1	0.24	0.04
4	98	-300.0	-300.0	-299.7	1.66	0.97
5	100	-382.7	-382.7	-382.7	2.74	0.34
6	69	-466.0	-464.7	-453.4	22.94	4.94
7	78	-568.2	-565.9	-541.2	32.61	4.12
8	63	-676.9	-670.7	-633.4	54.21	4.99
9	28	-789.2	-770.9	-736.5	115.3	8.69
10	24	-901.3	-866.9	-814.5	157.6	10.16

Table 3b.

n	t_{\min}	t_{\max}	Γ_{\min}	Γ_{\max}
3	0.15	0.52	0	1
4	0.54	4.46	0	4
5	1.42	10.13	0	4
6	2.36	57.20	0	12
7	5.66	76.94	0	11
8	5.39	126.00	0	11
9	26.21	198.36	1	13
10	35.76	207.13	1	8

Table 4. Statistics for seeded runs of $(Si)_n$

$\Delta \bar{V} = \bar{V}_{\text{seed}} - \bar{V}_{\text{noseed}}$. See text for discussion. Other notation is as in Table 3.

n	%GM	\bar{V}	$\Delta \bar{V}$
3	100	-186.1	0.0
4	100	-300.0	0.0
5	100	-382.7	0.0
6	100	-466.0	-1.3
7	64	-563.6	+2.3
8	95	-676.2	-5.5
9	5	-765.0	+5.9
10	92	-899.5	-32.6

,

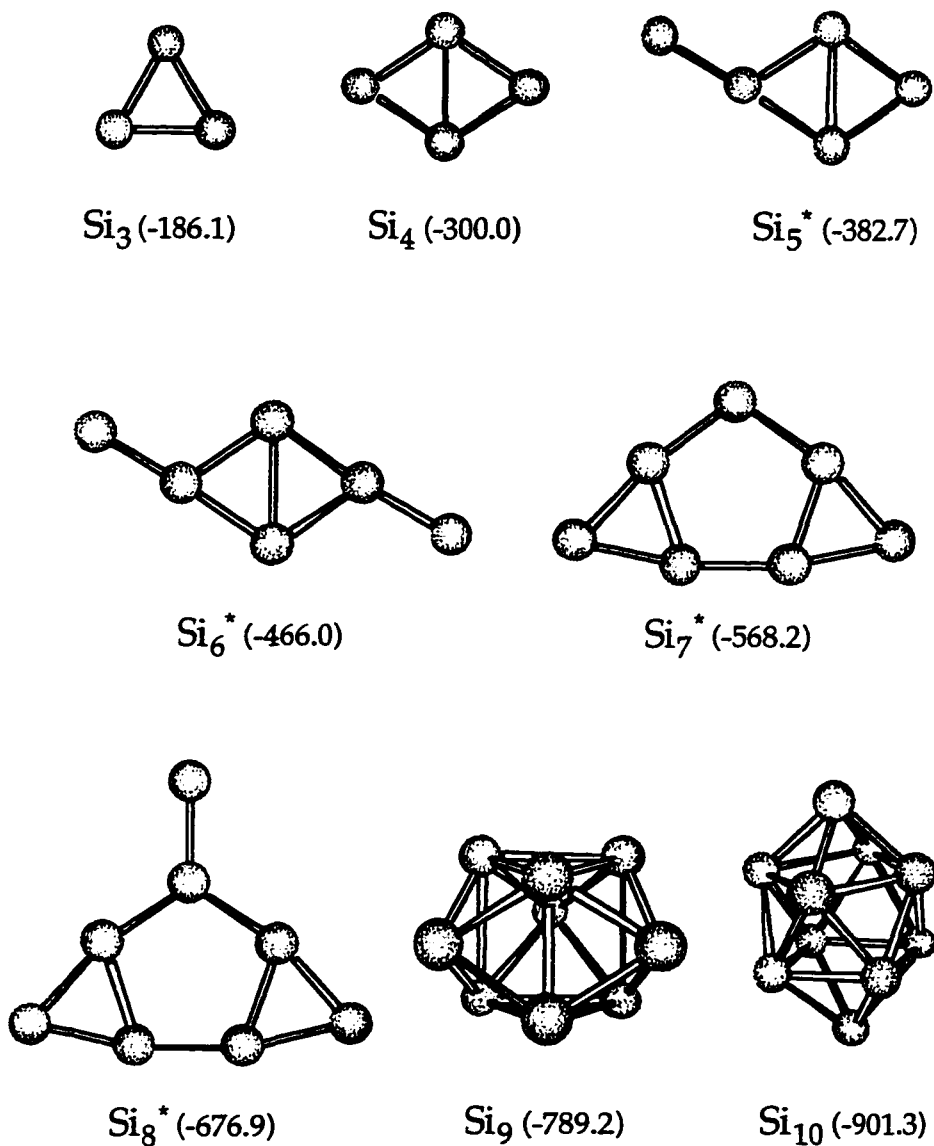
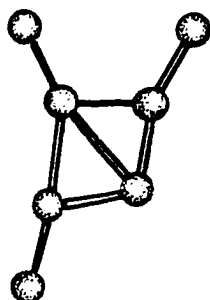
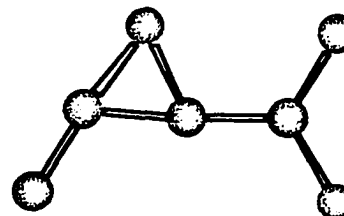
Structures for the $(\text{Si})_n$ cluster, $n = 3-10$ 

Figure 5. Structures for the $(\text{Si})_n$ cluster, $n = 3-10$. The $(\text{Si})_n$ ($n=3-8$) are planar, or very nearly so. Also given are the energies in kcal/mol. Those structures marked with an asterisk lie below the reported global minimum of Ref. 41.

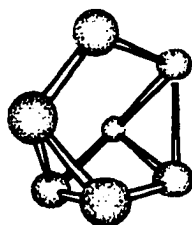
Most fit cluster as a function of generation number



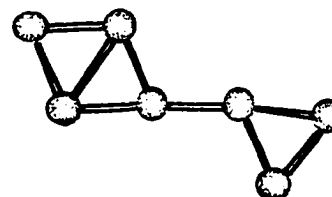
$\Gamma = 0$ (-520.85)



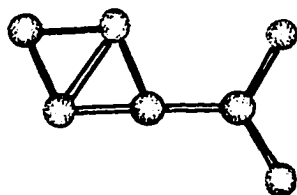
$\Gamma = 1$ (-536.90)



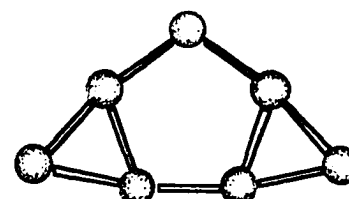
$\Gamma = 2,3$ (-542.05)



$\Gamma = 4,5,6$ (-551.75)



$\Gamma = 7,8,9,10,11$ (-561.96)



$\Gamma = 12$ (-568.17)

Figure 6. Fittest structure as a function of generation number for a run of seeded (6+1) (Si)₇. Energies in parenthesis are in kcal/mol.

CHAPTER III

THE GA OPERATORS FOR ATOMIC AND MOLECULAR CLUSTERS

It is clear from Chapters I and II that the GA approach using real-valued Cartesian variables and appropriate operators is more powerful than was anticipated by traditional GA workers. The method was able to outperform binary coded algorithms for LJ clusters both in terms of timing and in the size of the search space.²⁵ Deaven, Ho and coworkers²⁸ have managed to minimize LJ clusters up to $n=100$ using a similar real-valued GA. They point out that this is the first time any single technique has been used to minimize this entire range of structures. Furthermore, their calculations detected previously-unreported minima in several cases. We reported similar findings in the calculation of silicon clusters³⁸ in Chapter II.

Clearly, the use of real-valued variables and nontraditional genetic operators provides portability of coordinates; there is no need to recast the problem for each new cluster size. In addition, these coordinates in no way restrict the size of the search space. Furthermore, the method seems capable of minimizing problems of chemically-interesting size in reasonable CPU times.

The chief purpose of this chapter is to extend our treatment of clusters to molecular clusters using the same approach. A further goal is to systematically explore the performance of some of the possible operators for real-valued representations in GA calculations.

We assume the molecules within the clusters to be treated as rigid. The most economical coordinates to describe the internal coordinates of a rigid body are the three Euler angles⁴³. The atomic coordinates of the molecule are conveniently first described in body-fixed (BF) coordinates. For instance, for the H₂O molecule chosen as an example here, one could choose the following:

$$r_{\text{O}} = (0,0,0)$$

$$r_{\text{Ha}} = (r_{\text{OH}} \cdot \cos \gamma, r_{\text{OH}} \cdot \sin \gamma, 0)$$

$$r_{\text{Hb}} = (r_{\text{OH}} \cdot \cos \gamma, -r_{\text{OH}} \cdot \sin \gamma, 0)$$

where we have made the particularly simple choice of taking the O atom as the origin, r_{OH} is 0.9572 Å, and γ is 52.26 degrees. A set of Euler angles (θ, ϕ, ψ) is chosen randomly: $0 \leq \alpha \leq 2\pi$, where $\alpha = \theta, \phi, \psi$. Then the molecule is rotated into the SF frame $r_{\text{OHa}}, r_{\text{OHb}}$ by applying a rotation matrix⁴³ $R(\theta, \phi, \psi)$ to the vectors r_{OHa} and r_{OHb} . Finally, the SF coordinates of each atom can be found by adding the displacement of the O atom from the SF origin, R_{O} . Thus, $x_{\text{O}} = R_{\text{O}}$, $x_{\text{OHa}} = R_{\text{O}} + r_{\text{OHa}}$, etc. The potential energy can then be obtained using these SF vectors.

The initial R_O are randomly obtained by analogy with the SF coordinates for the atomic clusters. Thus, a (rigid) molecule is completely described by the six coordinates $(R_{Ox}, R_{Oy}, R_{Oz}, \theta, \phi, \psi)$.

The potential used for the H_2O clusters was the TIP3P potential of Jorgensen et al.⁴⁴ While this is acknowledged not to be a very accurate potential, it has been investigated in some detail. In addition, its derivatives are relatively simple.

Each individual's initial (R_{Ox}, R_{Oy}, R_{Oz}) coordinates are randomly chosen within a box of size L^3 centered at the origin. We take $R_1 = L(\zeta - 0.5)$, etc., where ζ is a freshly-generated random number between 0 and 1. We have used $L = \sqrt[3]{3n} r_e$, where $r_e = 2.75 \text{ \AA}$ is the dimer equilibrium OO distance. Each molecule's Euler angles, θ , ϕ and ψ , are initially generated randomly on the interval $[0, 2\pi]$. A conjugate gradient minimization is performed every generation on each individual to place each structure in the vicinity of the nearest minimum. The same precautions as in the atomic case are taken to avoid evaporation.

The variables for molecular clusters were manipulated in the following fashion. The strings of R and angular coordinates were separated. Thus, for the i th individual, $X_i = (R_{Ox1}, \dots, R_{Ozn})$, $Y_i = (\theta_1, \dots, \psi_n)$. Operations were carried out on the X and Y strings separately. For the X list, this was essentially identical to

the atomic procedure. For the Y list, the angles θ , ϕ and ψ are initially defined on the interval $[0, 2\pi]$. Since the trigonometric functions are valid for all values of the angles, we left the values produced in the CG alone (that is, they were not returned to the $[0, 2\pi]$ interval). After the manipulations, the coordinates for molecule 1 were reassembled by assigning the first three variables of X to be (R_{Ox}, R_{Oy}, R_{Oz}) of molecule 1, the first three variables in Y to be (θ, ϕ, ψ) of molecule 1, and so on. See Appendix A for a detailed description of the genetic operators.

The participation of an individual in future breeding operations depends upon its fitness. Since the function to be optimized for clusters is the (Born-Oppenheimer) potential energy, the fitness is a function of the potential. The operators used were: 1. inversion (In) ; 2. geometric mean (Ge) ; 3. arithmetic mean (Ar) ; 4. n-point crossover (Nx); 5. 2-point crossover (2x); 6. 1-point crossover (1x). We give the complete details in the Appendix.

A typical run contains 10 or 20 individuals in a population. A run was terminated when either the (purported) global minimum was found or the potential energy of the fittest structure did not change for 5 generations.

RESULTS AND DISCUSSION

Atomic Clusters

Results for a range of $(LJ)_n$ and $(Si)_n$ clusters have been presented elsewhere.^{25,38} We focus here on the examples of $(LJ)_{13}$ and $(LJ)_{19}$. We have chosen these as examples of a highly symmetric and a rather asymmetric cluster respectively. For all runs, the CPU time is recorded, as well as the minimum potential energy reported, and the number of generations required to find that minimum. We have previously²⁵, reported timing data. In this chapter, we will focus on the frequency of location of the global minimum as the chief criterion of the GA's performance. Where appropriate, other data will be mentioned.

1. Performance of Operators

Zeiri²⁹ proposed six Genetic Operators which are appropriate for real-valued genomes. We have tested five of these operators, together with the 1-point crossover, on the $(LJ)_{13}$ and $(LJ)_{19}$ clusters, whose global minima are well known.⁶ We have tested each operator individually, and in combination with all other operators. In order to clarify our findings, we anticipate some of the results. It was seen that the operators fell into three natural groupings, both in form and in performance. These were: (I) Inversion (one parent-one child); (II) Averaging (two parents-one child); (III) Crossovers (two parents-two children).

$(LJ)_{13}$

In Table 5 we give the number of times the $(LJ)_{13}$ global minimum was reached out of 100 independent attempts for the operators used individually, and in pairs. The diagonal elements are the runs with individual operators. In the last column is given the average performance over all six combinations. All calculations reported use the range fitness as defined in Chapter I.

From the diagonal elements, it can be seen that both averaging operators perform very well as the only operator present. The crossover and inversion operators are significantly worse. In combination with any other operator from a different class, however, both crossover and inversion significantly improve their performance. On the other hand, the improvement when a crossover is paired with another crossover operator is marginal. Finally, the most robust operators are the averaging operators, performing well with almost any partner.

We now consider $(LJ)_{13}$ minimization with operators taken three at a time – one from each class of operator. The performances are given in Table 6. In all cases, the combination of three operators is extremely efficient. In addition, it outperforms the average performance of the same three operators taken two at a time.

We have also carried out all other possible combinations of operators. Rather than enumerate them all, we give average performances over all combinations in which each operator participates. We also include the combination in which it performs worst, and that in which it performs best. This is given in Table 7. For this particular cluster, it is clear that a judicious combination of operators can easily yield the global minimum on one hundred per cent of the attempts. In addition, we note that the mean CPU time (DEC 2100/500) needed to find the global minimum varied from 1.91 to 6.43 sec per structure, and that most runs took fewer than twenty generations to converge (whether to the global minimum or not). In most cases, the shortest average times and fewest generations were for those cases where the global minimum was located with high frequency.

(LJ)₁₉

In Table 8 we give the same data for (LJ)₁₉ as was shown in Table 5. The most striking feature of this Table is the drop in the number of "hits" in comparison with Table 5. This is not surprising; the search space is roughly 50% larger, and the number of local minima has increased dramatically by a ratio of about $\exp(19^2 - 13^2)$. Clearly, the efficiency of the method does not scale with the increase in the number of local minima.

As before, both averaging operators perform very well as the only operator present. The crossover and inversion operators are significantly worse. In combination with any other operator, however, the crossover significantly improves its performance. The inversion improves its performance if combined with an averaging operator. Similarly, the crossovers do well with averaging operators, but not with each other or inversion. Finally, as before, the most robust operators are the averaging operators, performing well with almost any partner.

In Table 9 we consider combinations of three operators. As for the $(LJ)_{13}$ case, the use of one operator from each of the three classes is always superior to the average performance of the same operators taken two at a time. However, there is no guarantee that the use of three operators will be better than the best binary combination. Unfortunately, of course, the best combination is generally not known in advance. From examination of Table 9, it appears that the 2-point crossover tends to be less effective than either the 1-point or the n-point.

Table 10 is analogous to Table 7. The best combinations of operators clearly always include an averaging operator; the worst a crossover - most frequently the 2-point. From what we have seen above, however, it is a good

strategy to include at least one crossover operation. It would appear that either the 1-point or n-point is in general to be preferred over the 2-point.

For the (LJ)₁₉ case, the mean CPU times range from 4 to 30 sec per structure, and typically 10 to 20 generations are required for convergence.

2. The Slice Operator

Deaven and Ho and coworkers^{28,45} have suggested another Genetic Operator. Roughly, this operation is carried out as follows. Each cluster is cut by a plane which contains the centroid of the cluster, yielding two subclusters. Subclusters arbitrarily considered to be "above" the plane are then added to subclusters "below" the plane, from different original clusters. The resulting clusters (after suitable adjustments to ensure the number of atoms is correct) are relaxed using a CG minimization. Sufficiently fit offspring clusters are accepted into the population. Deaven, Ho and coworkers^{28,45} use all possible above-below combinations in their work. In order to compare their operator with those described above, we use a slightly different procedure here. As with the crossover operators described above, two candidate clusters are selected based on their fitness, and the offspring are the two above-below combinations of the subclusters. More details are given in Appendix A.

The results using this operator alone are as follows. For the (LJ)₁₃ cluster, the global minimum was located in 66 out of 100 independent attempts (mean CPU time 10.35 sec; mean number of generations 29). By comparison with Table 7, we see that this is comparable to the better crossover operators for this cluster size, but less efficient than the averaging operators.

For the (LJ)₁₉ case, the global minimum is found on only 7 out of 100 runs (mean CPU time 18.84 sec; mean number of generations 89). Again, this is comparable to inversion and the 2-point crossover for this cluster size, but inferior to the averaging operators.

3. Other Fitness Functions

In order to explore the possible effects of the choice of fitness parameter we have carried out calculations using an exponential fitness scheme. In this case, the intermediate value F_i is calculated as

$$F_i = \exp[-\alpha V_i] \quad i = 1, n$$

and the f_i are normalized as stated previously for the range fitness function.

We have judiciously chosen the two values of α we investigate here, $\alpha=2$ and $\alpha=5$, to illustrate the possible consequences of using an exponential scheme. We report results for a system of 13 LJ atoms. The minima in this system have potential values ranging from 0 to -0.545 eV.

In Figure 7, we show the fitness functions over this energy interval for the two exponential schemes, and the range scheme, suitably normalized. It can be seen that the range preferentially selects lower values of the potential. The exp-2 is rather less selective, while the exp-5 is strongly weighted towards low potential values.

We show results for $(LJ)_{13}$ using the individual operators and the three fitness schemes in Table 11. In each case, the best performance is underlined. It is clear that there is little to choose between the three schemes. However, exp-5, which weights the fittest structures in the population very heavily performs least well. Therefore, it does not appear to be productive in this case to be highly selective; it appears that maintaining a diverse population is advantageous.

Molecular Clusters

By analogy with our earlier work on atomic clusters²⁵, we have explored the feasibility of locating the global minimum for $(H_2O)_n$ using all the genetic operators. Statistics for the runs with $n=2-8$ are given in Table 12. It can be seen that most of the smaller water clusters are trivial to minimize in the sense that almost any random initial geometry descends to the global minimum. For $n=7$ and 8, the task is more challenging. It can be seen that the computational effort

required to locate the global minimum rises dramatically as the size of the cluster grows. It is possible that we would be able to locate the minimum if we allowed the population to evolve longer. However, this seemed needlessly wasteful of resources.

For this reason, we treat the clusters with $n=9$ through 13 in a slightly different manner. In these cases we start ten individual populations, and evolve each in the usual way until the convergence criteria are met. We then construct a new population from the fittest of each of the initial populations, and evolve this new population. In all cases, there was some improvement in the potential energy. The geometries of $(\text{H}_2\text{O})_n$ for $n=2$ through 13 are shown in Figures 8 and 9. We are in agreement with published geometries and energies⁴⁶ for $n=8$. The larger clusters all show the “fused cube” seen in studies⁴⁷ of water clusters with $n=8,12,16$ and 20.

Performance of operators

The particular interest in our study of molecular clusters lies in the evaluation of the operators on the angular part of the problem in particular. This has received considerably less attention than the treatment of the center-of-mass coordinates. In order to isolate this performance, we carry out calculations on a $(\text{H}_2\text{O})_8$ cluster with the eight O atoms frozen at the geometry of the global minimum. The operators then act only on the Euler angles. Table 13 is similar to Tables 5 and 8, and documents the success of the operators individually and in pairs.

For the Euler angles, the best operators, both individually and in combination are the 2-point crossover and inversion. The remaining operators performance is fair. It is unclear why these results are in such contrast to those for the Cartesian coordinates.

We have also carried out runs with all possible combinations of operators for this system. Our qualitative findings remain unaltered; we therefore refrain from showing this extra detail.

CONCLUSIONS

We have shown that we can extend earlier work^{25,38} using a modified Genetic Algorithm approach in Space-Fixed real variables to rigid molecular clusters. This is done by using a Cartesian space-fixed reference vector to each molecule, and describing the internal coordinates of each molecule by the three Euler angles. The populations of Cartesian and angular coordinates are acted upon separately by the genetic operators. Using this method we were able to locate global minima for $(\text{H}_2\text{O})_n$, $n=2-13$, using relatively little CPU time.

We have also tested the efficiency of the proposed genetic operators singly and in combination with other operators. For atomic clusters, averaging operators are clearly the most efficient if used individually, but may improve their performance in judicious combination. Since the correct combination is not known *a priori* for any given problem, we recommend the use of a mix of operators: one averaging, one crossover, and inversion.

For atomic clusters, we have compared these operators with a "slice" operator similar to that employed by Deaven, Ho, and coworkers.^{28,45} We find this operator to be comparable in efficiency to crossover operators, but less good than averaging operators. It is not clear whether there is an analogous operator

for angular coordinates. We therefore have made no attempt to implement this approach for molecular clusters.

We have also investigated the possibility of using different fitness functions for atomic clusters. It appears that trying to bias the selection towards the very fittest individuals is counterproductive; the Genetic Algorithm appears to work best when there is reasonable diversity in the population.

Table 5. Number of times GM located for (LJ)₁₃

Number of times out of 100 runs global minimum located for (LJ)₁₃ with combinations of operators. Diagonal elements give performance of operator used alone; off-diagonal elements give the performance for two operators used together. "In" denotes inversion; "Ar" arithmetic mean; "Ge" geometric mean; "1x" 1-point crossover; "2x" 2-point crossover; "Nx" N-point crossover.

Operator							Average
In	50						76
Ar	100	97					98
Ge	98	97	91				96
1x	82	97	95	26			67
2x	67	99	94	70	63		77
Nx	61	100	100	33	66	57	70
	In	Ar	Ge	1x	2x	Nx	

Table 6. Performance of operators for $(LJ)_{13}$

Operators are used three at a time. The first three columns indicate the operators used. The fourth column is the number of times the global minimum is located in 100 runs. The fifth column gives the average performance of the three operators used in pairs, taken from the data in Table 5.

Operators			GM located	Pair Average
In	Ar	1x	99	93
In	Ar	2x	100	89
In	Ar	Nx	99	87
In	Ge	1x	98	92
In	Ge	2x	99	86
In	Ge	Nx	99	86

Table 7. Summary of operator's performance, (LJ)₁₃

All possible combinations with other operators for (LJ)₁₃. The second column indicates the worst combination in which the operator was used with (in parentheses) the number of times out of 100 in which the global minimum was located. The third column indicates the average number of times out of 100 the global minimum was located. The fourth column gives the best combination in which the operator participated.

Operator	Worst	Average	Best
In	In (50)	91	In, 2x, Ar (100)
Ar	1x, Nx, Ar (93)	98	Nx, Ar (100)
Ge	Ge (91)	97	2x, Nx, Ge (100)
1x	1x (26)	90	1x, 2x, Ar (100)
2x	In, 2x, Nx (60)	91	2x, Nx, Ge (100)
Nx	1x, Nx (33)	90	Nx, Ar (100)

Table 8. Number of times GM located for (LJ)₁₉.

As for Table 5, except (LJ)₁₉.

Operator							Average
In	7						26
Ar	68	57					64
Ge	57	67	54				61
1x	13	55	62	0			24
2x	4	55	49	13	5		22
Nx	4	81	74	1	5	0	28
	In	Ar	Ge	1x	2x	Nx	

Table 9. Performance of operators for (LJ)₁₃.

As for Table 6, except (LJ)₁₉.

Operators			GM located	Pair Average
In	Ar	1x	75	45
In	Ar	2x	55	42
In	Ar	Nx	79	51
In	Ge	1x	77	44
In	Ge	2x	62	36
In	Ge	Nx	72	45

Table 10. Summary of operator's performance, (LJ)₁₉

As for Table 7, except (LJ)₁₉.

Operator	Worst	Average	Best
In	In, 2x (4)	51	In, Nx, Ar (79)
Ar	2x, Ar, Ge (44)	61	Nx, Ar (81)
Ge	2x, Ar, Ge (44)	62	In, 1x, Ge (77)
1x	1x (0)	51	In, 1x, Ge (77)
2x	In, 2x, Nx (3)	46	In, 1x, 2x, Nx, Ge (70)
Nx	Nx (0)	51	Nx, Ar (81)

Table 11. Fitness function varied

Performance of individual operators for $(LJ)_{13}$ for three different fitness functions

(see text). Data are number of times global minimum is located out of 100 runs.

	range	$\exp(\alpha=2)$	$\exp(\alpha=5)$
In	50	56	52
Ar	97	100	92
Ge	91	97	90
1x	26	24	24
2x	63	69	66
Nx	57	50	50
Average	64	66	62

Table 12. Summary of performance for $(\text{H}_2\text{O})_n$, $n=2-8$

The second through fourth columns give the lowest local minimum potential (which is presumed to be the global minimum), the mean value of the potential in the population, and the highest local minimum found, respectively. The fifth column gives the maximum, the mean, and the minimum number of generations before the convergence criteria were met. The sixth column gives the number of times out of 100 the global minimum was located. The final column gives the mean CPU time for minimizing each structure.

n	-Vmin	$-\bar{V}$	-Vmax	$\Gamma_{\max}, \bar{\Gamma}, \Gamma_{\min}$	$\frac{\#V_{\min}}{100}$	\bar{t} [sec]
2	6.543	6.543	6.543	0, 0.00, 0	100	0.636
3	17.448	17.448	17.448	0, 0.00, 0	100	2.679
4	29.306	29.306	29.306	0, 0.00, 0	100	9.109
5	38.771	38.771	38.771	0, 0.00, 0	100	20.495
6	47.811	47.811	47.811	2, 0.05, 0	100	33.252
7	57.944	57.553	57.425	12, 5.04, 0	22	166.510
8	70.681	68.885	67.431	20, 8.12, 0	13	338.778

Table 13. Number of times GM located for $(\text{H}_2\text{O})_8$

As for Table 5, except for $(\text{H}_2\text{O})_8$, with O atoms fixed (see text).

Operator							Average
In	61						65
Ar	59	46					58
Ge	69	54	44				58
1x	63	60	52	34			54
2x	70	67	64	75	73		70
Nx	66	60	62	42	70	44	57
	In	Ar	Ge	1x	2x	Nx	

Fitness -vs- potential

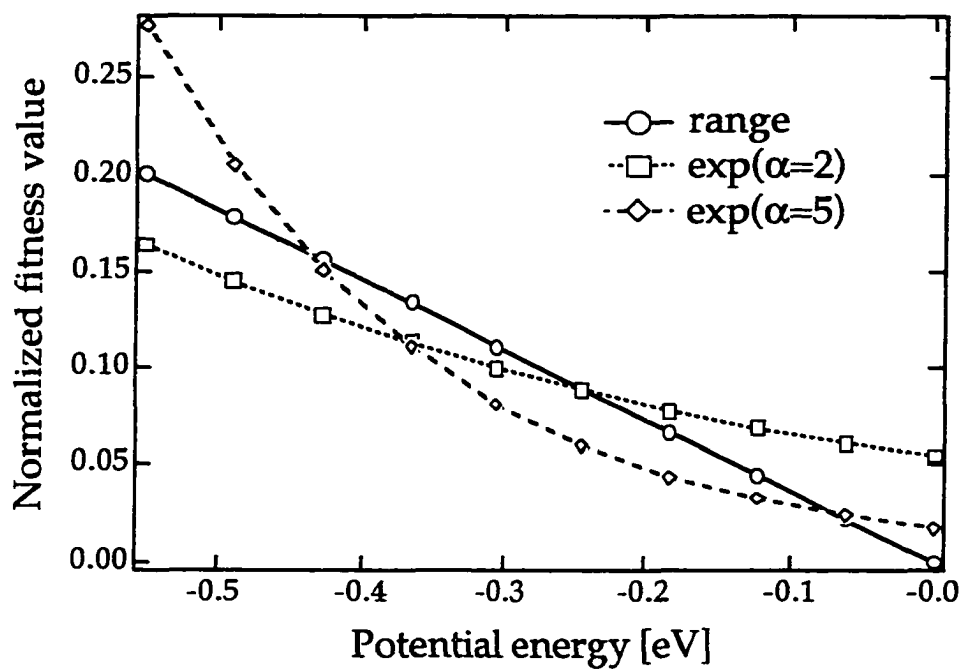


Figure 7. Fitness as a function of energy for three different fitness functions (see text). The minimum energy is that of $(LJ)_{13}$.

Geometry of $(\text{H}_2\text{O})_n$ structures, $n = 2-8$

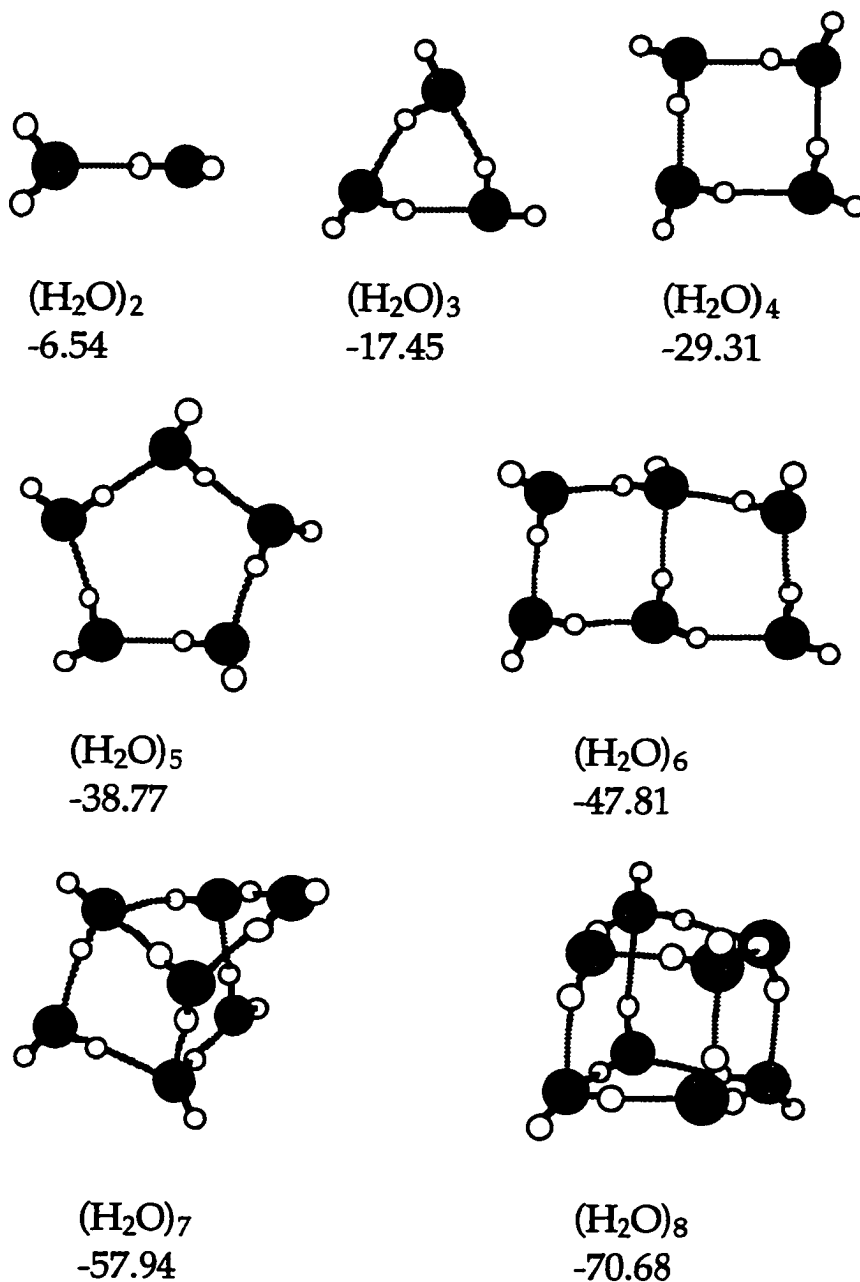


Figure 8. Geometry of lowest potential energy structure found for $(\text{H}_2\text{O})_n$, $n = 2-8$. Energies in kcal/mol.

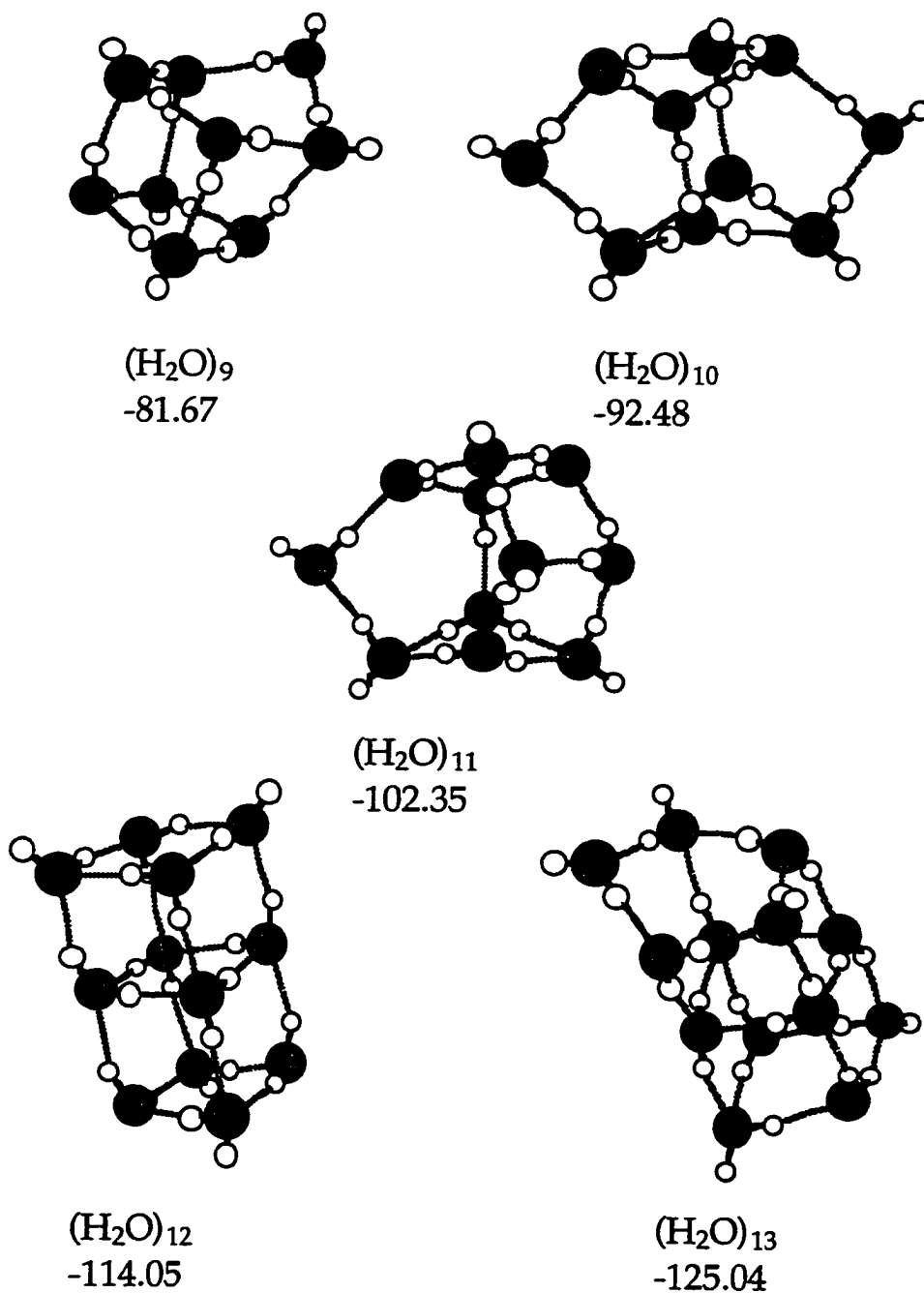
Geometry of $(\text{H}_2\text{O})_n$ structures, $n = 9-13$ 

Figure 9. As for figure 8, except $n=9-13$.

CHAPTER IV

BENZENE, NAPHTHALENE, AND ANTHRACENE

Model potentials are available for these aromatic hydrocarbons,⁴⁸⁻⁵¹ and some have already been used in cluster geometry minimizations.⁴⁸⁻⁵³ In particular, the series benzene-naphthalene-anthracene provides an increase in the number of local minima to be searched without increasing the dimensionality of the coordinate space to be searched.

The Dulles-Bartell Benzene Dimer

Dulles and Bartell⁵¹ recently proposed a new potential energy function for benzene clusters. In their paper, the authors give the geometry and potential energy of a proposed global minimum for the benzene dimer. We have applied The SFMGA method to the benzene dimer problem using the Dulles-Bartell potential.

The runs were carried out as follows. There were ten individuals in the population. Six genetic operators were used with equal weighting: inversion; arithmetic and geometric averaging; one-, two- and N-point crossovers. An elitist strategy was adopted, with the best two parents carrying over intact into the next generation. Several independent runs were carried out.

We report here a lower dimer potential energy ($-10.630 \text{ kJ mol}^{-1}$) than that reported by Dulles and Bartell ($-10.582 \text{ kJ mol}^{-1}$). The geometry of the dimer is given in Figure 10. The coordinates of the atomic sites are given in Table 14. (There is a typographical error in the Appendix of Ref. 51) The C-C intersite distance should be 1.401 \AA .) Roughly 80% succeeded in finding the GM we propose here. The average number of generations needed to find the GM in these runs was 180.

The ability of the SFMGA to find unsuspected global minima has been pointed out for silicon clusters.³⁸ The SFMGA is a powerful minimization tool which explores the potential surface in a highly nonlocal manner. The advantage of this is that the technique is undeterred by high energy saddle points between local minima, which may cause difficulties for searches based on molecular dynamics or Monte Carlo approaches.

Larger Hydrocarbon Clusters

To model the intermolecular potentials between aromatic hydrocarbons and to be able to compare with existing calculations, we chose the (exp-6-1) potentials of Williams and Starr⁴⁹ for benzene, and of Williams and Xiao for naphthalene and anthracene.⁵⁰ In this potential, each nonbonded atom-atom potential is given by:

$$v(r_{ij}) = B_{ij} \exp(-C_{ij}r_{ij}) - A_{ij}r_{ij}^{-6} + cq_iq_jr_{ij}^{-1}$$

where i and j represent atoms on different molecules. The first term models short range repulsion, the second dispersion, the last Coulombic interaction between the partial charges on the atoms.

The constant c has the value $1389.963 \text{ kJ mol}^{-1} \text{ \AA}^{-1} \text{ e}^{-2}$. The total potential is given by the sum over all molecules. The parameters are given in Table 15. Note that the notation in Ref. 50 is rather nonstandard. We have relabelled the atoms according to more usual organic chemistry nomenclature.⁵⁴

RESULTS AND DISCUSSION

In Figures 11 and 12, we show the minimum potential energy geometries we have obtained for naphthalene (Naph) and anthracene (Anth) clusters. We have not shown benzene clusters, since these have been shown and discussed in some detail elsewhere.⁵¹ We note, however, that some of our minima are slightly lower in energy than those reported before, but there are no new qualitative structural differences.

There are a few interesting points to note in the $(\text{Naph})_n$ and $(\text{Anth})_n$. First, the dimer $(\text{Anth})_2$ has D_{2d} symmetry, whereas $(\text{Naph})_2$ is "almost" D_{2d} . $(\text{Naph})_3$ is like $(\text{Ben})_3$ in that it has the centers of masses of the molecules in a ring, giving three times the dimer interaction. By contrast, $(\text{Anth})_3$ is D_{2h} , with the molecules stacked; the species are too sterically hindered to form a ring. The outer molecules have their faces aligned parallel to each other, the nearest neighbors antiparallel. The geometry yields only two dimer interactions, and this is reflected in the binding energies of Table 16. Both tetramers appear to be ring shaped with opposing molecules presenting their faces parallel to each other. It appears that the parallel alignment is favorable at fairly large distances, but the perpendicular is preferred at shorter distances.

It is traditional to report the progress of a GA calculation by plotting the objective function (here, the potential energy) as a function of generation number. However, this is deceptive when a gradient descent is also used; the CG in this case takes many steps per generation. The situation is further complicated by the fact that (in the CG case, at least) the gradient descent routine makes several “monitoring” function calls on each step to determine how to proceed. The evaluation of the potential energy function (and its derivatives) is typically by far the most expensive part of any cluster minimization code. Thus, we propose that a fairer measure of the rate of convergence of a real-coded GA with gradient descent is a plot of best potential versus function calls, rather than generation number. This will also facilitate comparison with other minimization techniques.

In Figure 13, we plot the SFMGA average best V/V_{\min} against function calls for $(\text{Ben})_n, n=6-9$. The performance for the smaller clusters has not been shown, since it occupies such a narrow strip of the early part of the plot. We report on the right hand side of the Figure the number of times in ten independent runs in which the global minimum was located.

In Figure 14, we show the average best SFMGA potential divided by the appropriate global minimum energy for hexamers of benzene, naphthalene and anthracene, versus function calls.

All the results reported so far have used the exp-6-1 potential of Williams and coworkers.^{49,50} We tested to see whether any of our findings were potential dependent by also using our method on an alternative potential for benzene clusters due to Dulles and Bartell.⁵¹ The global minima found are given in Table 15. We have previously reported a global minimum for the dimer.⁵³

In Figure 15, we compare the $\langle V/V_{\min} \rangle$ for $(\text{Ben})_6$ using the exp-6-1 and the Dulles-Bartell potentials. It can be seen that the method fares much less well on the latter potential. It appears that the potential landscape is considerable more "rugged" in this case. A recent paper has discussed the various line minimization techniques available for cluster problems. The authors recommend the BFGS for atomic clusters. We therefore incorporated a BFGS descent³⁴ into our SFMGA. The results are also shown in Figure 15. Clearly, for the Dulles-Bartell potential, the BFGS performs better than does the CG. However, the reverse is true for the exp-6-1 potential. It is therefore important that several linesearch routines be tested for any given problem.

CONCLUSIONS

The SFMGA's performance demonstrates the portability of the algorithm. The sensitivity of the SFMGA's performance to the potential function of the local optimization technique is evidenced by the differing results of the PR-CG and BFGS local minimization routines.

Table 14. Cartesian coordinates for DB benzene dimer

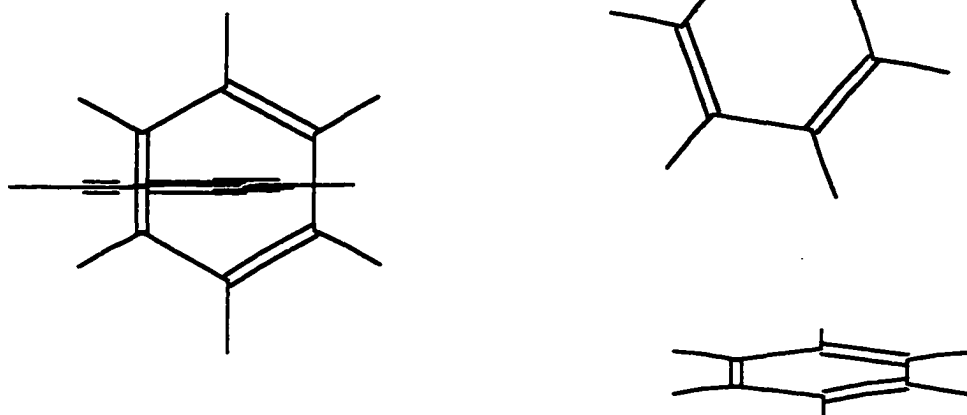
Cartesian Coordinates for each Atomic Site in SFMGA Calculated Global

Minimum of (C₆H₆)₂ Potential Energy. Benzene potential of Dulles and Bartell.

Atom		x [Å]	y [Å]	z [Å]
C	1	1.401000	0.000000	0.000000
C	2	0.700500	1.213302	0.000000
C	3	-0.700500	1.213302	0.000000
C	4	-1.401000	0.000000	0.000000
C	5	-0.700500	-1.213302	0.000000
C	6	0.700500	-1.213302	0.000000
H	7	2.432000	0.000000	0.000000
H	8	1.216000	2.106174	0.000000
H	9	-1.216000	2.106174	0.000000
H	10	-2.432000	0.000000	0.000000
H	11	-1.216000	-2.106174	0.000000
H	12	1.216000	-2.106174	0.000000
C	13	-0.206939	-0.119304	5.988163
C	14	0.986856	0.570408	6.237062
C	15	1.770428	1.023041	5.167511
C	16	1.360205	0.785962	3.849061
C	17	0.166409	0.096249	3.600162
C	18	-0.617162	-0.356384	4.669713
H	19	-0.783572	-0.452398	6.775249
H	20	1.288741	0.744875	7.207314
H	21	2.648945	1.530602	5.350677
H	22	1.936837	1.119056	3.061975
H	23	-0.135475	-0.078218	2.629910
H	24	-1.495680	-0.863945	4.486547

Figure 10.

DB benzene dimer geometry



Benzene (Dulles and Bartell potential⁵¹) dimer geometry for potential energy global minimum proposed here. Two different views are shown.

Table 15. exp-6-1 Potential Parameters for Aromatic Hydrocarbons**All molecules:**

	H-H	H-C	C-C
A (kJ mol ⁻¹ Å ⁶)	136	573	2414
B (kJ mol ⁻¹)	11677	65485	367250
C (Å ⁻¹)	3.74	3.67	3.60

Charges, q (e) :

Benzene [Ref 49]

C	-0.153	H	0.153
---	--------	---	-------

Naphthalene [Ref 50]

C ₁ position	C	-0.3592	H	0.1991
C ₂ position	C	-0.1402	H	0.1617
bridge C	C	0.2772		

Anthracene [Ref 50]

C ₁ position	C	-0.3108	H	0.1895
C ₂ position	C	-0.1653	H	0.1715
C ₉ position	C	-0.6022	H	0.2658
bridge C	C	0.2833		

Table 16. Aromatic hydrocarbon potential energies

Potential energy at claimed global minimum for aromatic hydrocarbon clusters of benzene, naphthalene, and anthracene. Top, using the exp-6-1 potentials of Williams and coworkers^{49,50}; bottom, using the potential of Dulles and Bartell.⁵¹ Binding energies are in kJ mol⁻¹.

Williams et al exp-6-1 potential			
n	(Ben) _{n}	(Naph) _{n}	(Anth) _{n}
2	10.976	25.648	44.505
3	32.098	58.483	90.008
4	55.629	94.919	147.549
5	79.106	133.368	206.225
6	106.479	172.182	266.818
7	134.092		
8	161.660		
9	191.519		
10	221.522		
11	252.187		
12	286.288		
13	324.723		

Table 16. (continued)

Dulles-Bartell Potential

2	10.630
3	29.444
4	55.373
5	79.003
6	109.017

Geometries for naphthalene clusters

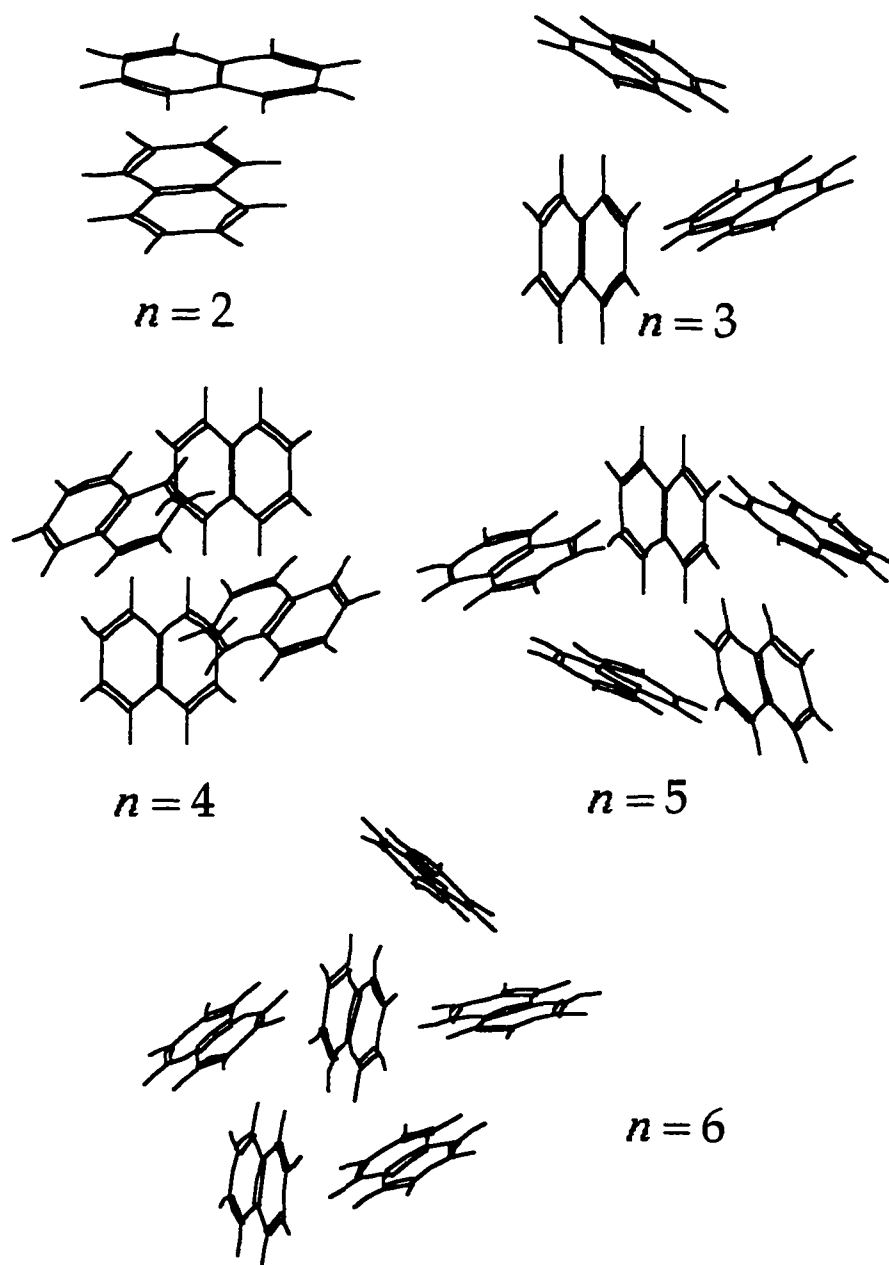


Figure 11. Minimum energy geometries for naphthalene clusters, $(\text{Naph})_n$, using the Williams et al. exp-6-1 potential.

Geometries for anthracene clusters

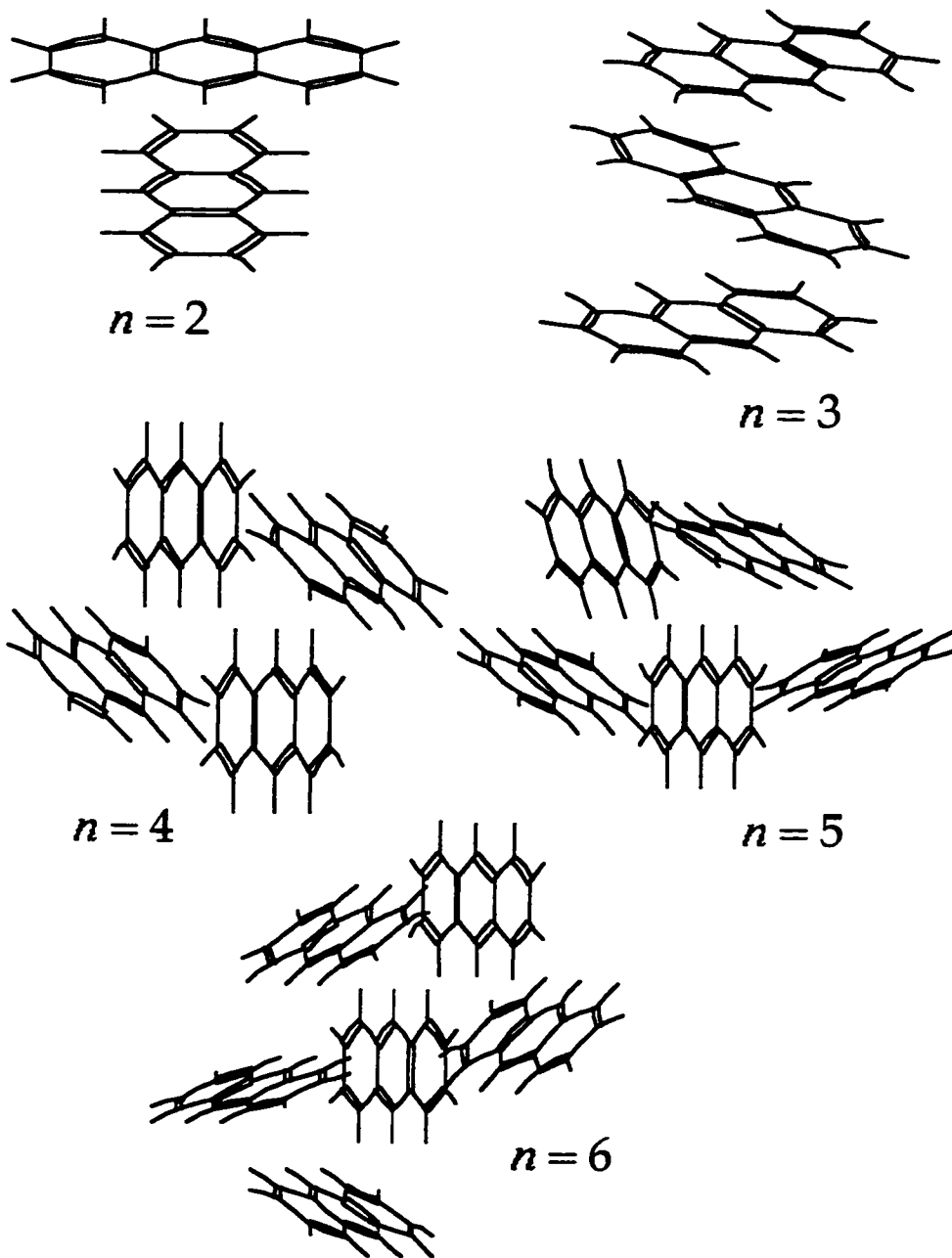


Figure 12. As for Figure 11, except anthracene clusters, $(\text{Anth})_n$.

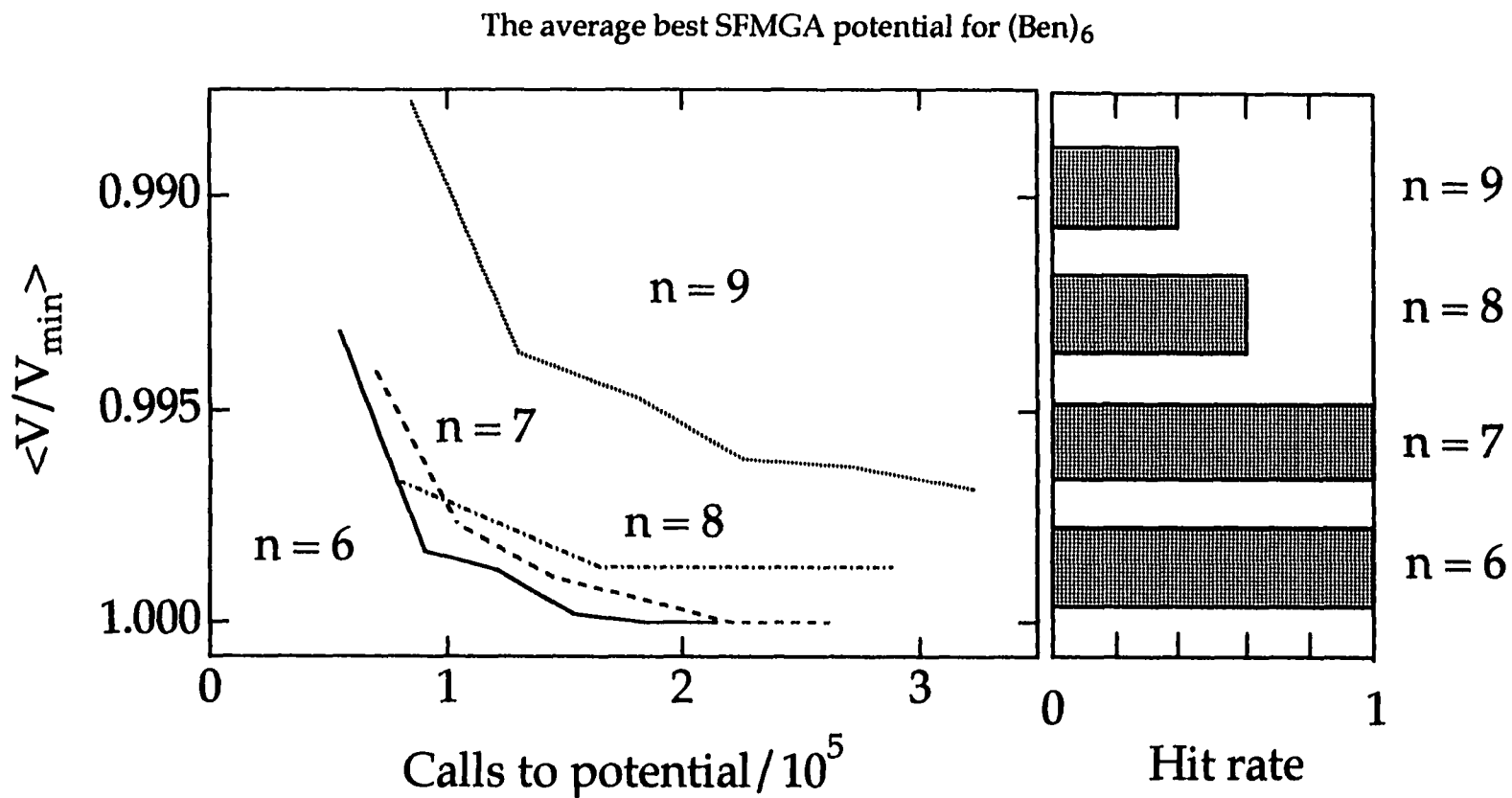


Figure 13. The average best SFMGA potential for $(\text{Ben})_n$ divided by the potential of the appropriate global minimum versus calls to the potential routine. Note the ordinate scale runs from large to small values. Also note the scale in comparison with Fig. 14. The histograms on the right show the number of times in the ten independent runs that the global minimum was reached after 3×10^5 potential calls.

The average best SFMGA potential for hydrocarbons

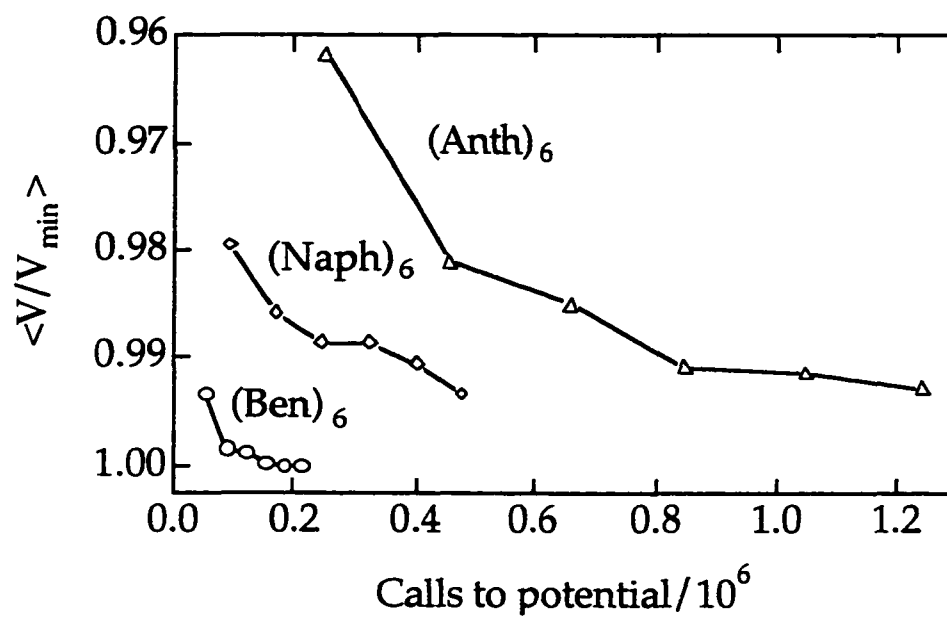


Figure 14. Average best SFMGA potential divided by the appropriate global minimum energy for hexamers versus function calls.

The average best SFMGA potential for $(\text{Ben})_6$

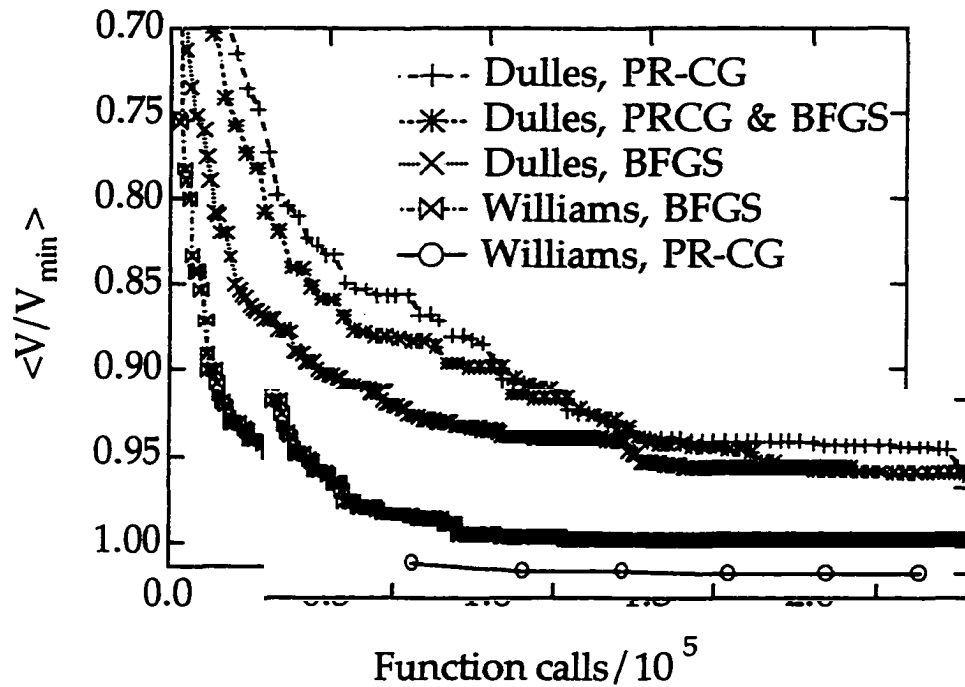


Figure 15. Average best potential divided by V_{GM} versus function calls for $(\text{Ben})_6$ for the Williams et al. exp-6-1 potential and the Dulles-Bartell potential for two different line search minimization methods. PR-CG denotes Polak-Ribière conjugate gradient; BFGS denotes Broyden-Fletcher-Goldfarb-Shanno variable metric.

CHAPTER V

CONCLUDING REMARKS

A modified version of the genetic algorithm approach,¹⁹⁻²¹ based on concepts of Darwinian evolution, has been successfully applied to the structural optimization of various cluster types.

Initially, the choice of space-fixed coordinates was rather controversial. However, this “counterintuitive” approach has been able to cope with considerably larger systems than those which have used binary coding. Furthermore, the most successful applications of the GA to clusters to date have also used derivative information to relax “nascent” offspring to a local minimum on the surface.

In particular, we²⁵ have considered the $(LJ)_n$ system, using real coding on the SF Cartesian coordinates. Versions of Zeiri’s genetic operators²⁹ were employed. Offspring were then relaxed using a conjugate gradient descent, and the relaxed geometry was used as the genotype for the next generation.

With this approach, clusters as large as (LJ)⁵⁵ were minimized, without the need for seeding. In a follow-up work we investigated the efficacy of each of the genotypic operators, and found that the averaging operators were the primary workhorses for atomic clusters.

We⁵⁵ have also used a “slice” operator similar to that of Deaven et al.^{28,45}

The nature of the slice operator will necessarily bias the offspring clusters towards spherically symmetrical shapes. An operator which biases the solution in this way, or others,⁵⁶ is known as phenotypic, while operators which minimize any bias, are known as genotypic. The phenotypic slice operator as implemented in Chapter 3 performed no better than the genotypic operators, indeed, it fared less well than the best. One could also view seeding – using a minimized n-1 cluster as a starting point for the n cluster – as phenotypic information. It has been shown that in some cases³⁸ this can actually be detrimental to the efficiency of the method if the transition from the n-1 to n cluster is marked by a change in morphology. Further, it is the impression of the author that tailoring an effective phenotypic operator requires some knowledge of the expected result. While this information appears to be useful for previously-explored systems,⁵⁶ it could bias the search away from the true minimum for unknown problems.

Since it is to be hoped that the GA can be developed into a general minimization tool, it seems wise to avoid guiding the solution towards intuitive solutions when the true minimum may well be nonintuitive. In our opinion, genotypic operations present a “least-biased” approach to genetic algorithm searches.

The evidence appears overwhelming⁵⁷ that the most successful minimization calculations to date using GA are those in which the genotype is encoded using real numbers, and, further, potential derivative information⁵⁸ is incorporated.

LIST OF REFERENCES

- (1) Leach, A. R. *Rev. Comput. Chem.* 1991, 2.
- (2) Dodziuk, H. *Modern Conformational Analysis*; VCH: New York, 1995.
- (3) Comba, P.; Hambley, T. W. *Molecular Modeling*; VCH: New York, 1995.
- (4) Brooks, C. L.; Karplus, M.; Pettitt, B. M. *Adv. Chem. Phys.* 1988, 71, 1.
- (5) Floudas, C. A.; Maranas, C. D. *J. Chem. Phys.* 1992, 97, 7667.
- (6) Northby, J. A. *J. Chem. Phys.* 1987, 87, 6166.
- (7) Scheraga, H. A.; Kostrowicki, J.; Peila, L.; Cherayil, B. J. *J. Phys. Chem.* 1991, 95, 4113.
- (8) Doll, J. D.; Finnula, A. B.; Gomez, M. A.; Sebenik, C.; Stenson, C. *Chem. Phys. Lett.* 1994, 219, 343.
- (9) Somorjai, R. L.; Sylvain, M. *J. Phys. Chem.* 1991, 95, 4147.
- (10) Straub, J. E.; Amara, P.; Hsu, D. *J. Phys. Chem.* 1993, 97, 6715.
- (11) Anderson, J. B. *J. Chem. Phys.* 1975, 63, 1499.
- (12) Straub, J. E.; Andricioaei, I. *Comp. in Phys.* 1996, 10, 449.
- (13) Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. *Science* 1983, 220, 671.
- (14) Wille, L. T. *Chem. Phys. Letters* 1987, 133, 5.
- (15) Straub, J. E.; Ma, J.; Amara, P. *J. Chem. Phys.* 1995, 103, 1574.
- (16) Doll, J. D.; Frantz, D. D.; Freeman, D. L. *J. Chem. Phys.* 1990, 93.
- (17) Serra, P.; Stanton, A. F.; Kais, S. *J. Chem. Phys.* 1997, 106, 7170.

- (18) Kalos, M. H.; Whitlock, P. A. *Monte Carlo Methods*; John Wiley and Sons: New York, 1986.
- (19) Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*; Addison-Wesley: Reading, 1989.
- (20) Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*, 2 ed.; Springer-Verlag: , 1995.
- (21) *Handbook of Genetic Algorithms*; Davis, L., Ed.; Van Nostrand Reinhold: New York, 1991.
- (22) Judson, R. *Rev. Comput. Chem* **1997**, *10*, 1.
- (23) Hartke, B. *J. Phys. Chem* **1993**, *97*, 9973.
- (24) Hartke, B. *Chem. Phys. Lett.* **1995**, *240*, 560.
- (25) Niese, J. A.; Mayne, H. R. *J. Chem. Phys.* **1996**, *105*, 4700.
- (26) Pullan, W. J. *J. Comput. Chem* **1997**, *18*, 1096.
- (27) Mestres, J.; Scuseria, G. *J. Comp. Chem* **1995**, *16*, 729.
- (28) Deaven, D. M.; Tit, N.; Morris, J. R.; Ho, K. M. *Chem. Phys. Lett.* **1996**, *256*, 195.
- (29) Zeiri, Y. *Phys. Rev. E* **1995**, *51*, R2769.
- (30) Gregurick, S. K.; Alexander, M. H.; Hartke, B. *J. Chem. Phys.* **1996**, *104*, 2684.
- (31) Beauregard, J. N.; Mayne, H. R. *Surf. Sci. Lett.* **1993**, *280*, L253.
- (32) Goldberg, D. E. *Complex Systems* **1991**, *5*, 139.

- (33) Tully, J. C.; Xu, G. Q.; Holland, R. J.; Bernasek, S. L. *J. Chem. Phys.* **1989**, *90*, 3831.
- (34) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in Fortran*, 2 ed.; Cambridge University Press: , 1992.
- (35) Hoare, M. R.; Pal, P. *Adv. Chem. Phys.* **1971**, *20*, 161.
- (36) (MDBNCH found at <<http://www.sissa.it/furio/mdbnch.html>> give the following CPU times: DEC 2100/500 37.4 s; IBM R/6000 320H 165 s; a ratio of 4.41:1) . This benchmark program uses several potential and derivative calls with nested loops in a Molecular Dynamics code. The computational effort appears comparable to to that required for this problem.
- (37) Wales, D. J.; Doye, J. P. K. *Science* **1996**, *271*, 484.
- (38) Niesse, J. A.; Mayne, H. R. *Chem. Phys. Letters* **1996**, *261*, 576.
- (39) Igor (Graphing and Fitting Software); Wavemetrics, 1989.
- (40) Hoare, M. R. *Adv. Chem. Phys.* **1979**, *40*, 49.
- (41) Bolding, B. C.; Andersen, H. C. *Phys. Rev. B* **1990**, *41*, 10568.
- (42) Hoare, M. R.; McInnes, J. A. *Adv. Phys.* **1983**, *32*, 791.
- (43) Goldstein, H. *Classical Mechanics*; Addison-Wesley: Cambridge, 1953.
- (44) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926.

- (45) Deaven, D. M.; Ho, K. M. *Phys. Rev. Lett.* **1995**, *75*, 288.
- (46) Jordan, K. D.; Tsai, C. J. *J. Chem. Phys.* **1993**, *99*, 6957.
- (47) Jordan, K. D.; Tsai, C. J. *J. Phys. Chem.* **1993**, *97*, 5208.
- (48) van de Waal, B. W. *Chem. Phys. Lett.* **1986**, *123*, 69.
- (49) Williams, D. E.; Starr, T. L. *Comp. and Chem.* **1977**, *1*, 173.
- (50) Williams, D. E.; Xiao, Y. *Acta. Cryst.* **1993**, *A49*, 1.
- (51) Dulles, F. J.; Bartell, L. S. *J. Phys. Chem.* **1995**, *99*, 17100.
- (52) Williams, D. E.; Xiao, Y. *Chem. Phys. Letters* **1993**, *215*, 17.
- (53) Niese, J. A.; Mayne, H. R. *J. Phys. Chem.* **1997**, *B101*, 9137.
- (54) Streitwieser, A.; Heathcock, C. H. *Introduction to Organic Chemistry*, Third ed.; MacMillan: New York, 1985.
- (55) Niese, J. A.; Mayne, H. R. *J. Comput. Chem.* **1997**, *18*, 1233.
- (56) Pullan, W. J. *Comp. Phys. Comm.* **1997**, *107*, 137.
- (57) White, R. P.; Niese, J. A.; Mayne, H. R. *J. Chem. Phys.* **1998**, *108*, 2208.
- (58) Wales, D. J.; Doye, J. P. K. *J. Phys. Chem.* **1997**, *B101*, 5111.

APPENDICES

Appendix A

We give here the details needed to fully understand the genetic algorithm operators used herein. We denote the i^{th} geometry of the n -atom cluster as $X_i = (x_1, \dots, x_n)$, where $x_k = (x_k, y_k, z_k)$ is the displacement of the k^{th} atom. (See Chapter 3 for a description of the appropriate coordinate strings for molecular clusters. The behavior of any individual operator upon a string of values is, of course, independent of the coordinate representation.) While the distinction between x , y , and z coordinates is important for evaluating the potential, the operators act simply on a string of reals. To emphasize this we relabel the string of reals as $X_i = (c_1, \dots, c_{3n})$. We use c_k to denote $c_k(i)$ if there is no ambiguity. We summarize below the action of each of the operators. In all the expressions below k runs from 1 to $3n$. In the notation $[c_k(i), c_k(j)] = [c_k(j), c_k(i)]$ simultaneous substitution is implied, with the updated generation on the left hand side, the current generation on the right hand side of the assignment. We also include an example of each operator's behavior.

Inversion: $c_k = c_{q+r-k}$ $r \leq k \leq q$ r, q flat on $[1, 3n]$

A single parent, $[\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$, is required for inversion. For instance if $r = k-2$ and $q = k+1$ the resulting child is

$$[\alpha_1, \dots, \alpha_{k+1}, \alpha_k, \alpha_{k-1}, \dots, \alpha_{3n}].$$

1-point crossover: $[c_k(i), c_k(j)] = [c_k(j), c_k(i)]$
 $s < k \leq 3n$ s flat on $[1, 3n]$

Two parents, $[\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$ and $[\beta_1, \dots, \beta_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}]$, produce two children. For example, $[\beta_1, \dots, \beta_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$ and $[\alpha_1, \dots, \alpha_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}]$ may result, if the crossover occurs after the position labeled "k-1" in the parent strings above.

2-point crossover: $[c_k(i), c_k(j)] = [S_{s+k}(j), S_{s+k+3n}(j)]$ s flat on $[1, 3n]$

$S(ij) = (c_1(i), \dots, c_{3n}(i), c_1(j), \dots, c_{3n}(j))$ and it is understood that $s+k+3n$ is modulo $6n$. The two parents, $[\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$ and $[\beta_1, \dots, \beta_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}]$, may yield two children, $[\alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}, \beta_1, \dots, \beta_{k-2}]$ and $[\beta_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}, \alpha_1, \dots, \alpha_{k-2}]$ for example.

N-point crossover:

$$[c_k(i), c_k(j)] = [c_k(j), c_k(i)] \quad \text{if } \zeta > 0.5$$

$$[c_k(i), c_k(j)] = [c_k(i), c_k(j)] \quad \text{if } \zeta \leq 0.5$$

Two parents, $[\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$ and $[\beta_1, \dots, \beta_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}]$, produce two children. For example, $[\beta_1, \dots, \alpha_{k-1}, \beta_k, \beta_{k+1}, \dots, \alpha_{3n}]$ and $[\alpha_1, \dots, \beta_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \beta_{3n}]$ may result, depending on the $3n$ "fresh" random numbers, ζ .

Arithmetic mean:

$$\alpha_k(i) = 0.5(c_k(i) + c_k(j))$$

Two parents, $[\alpha_1, \dots, \alpha_n]$ and $[\beta_1, \dots, \beta_{3n}]$, produce one child, $[0.5(\alpha_1 + \beta_1), \dots, 0.5(\alpha_{3n} + \beta_{3n})]$.

Geometric mean:

$$\alpha_k(i) = (\text{abs}(c_k(i) \cdot c_k(j)))^{1/2}$$

Two parents, $[\alpha_1, \dots, \alpha_{3n}]$ and $[\beta_1, \dots, \beta_{3n}]$, yield one child, $[\{\text{abs}(\alpha_1 \cdot \beta_1)\}^{1/2}, \dots, \{\text{abs}(\alpha_{3n} \cdot \beta_{3n})\}^{1/2}]$.

Slice Operation: Two parents, $[\alpha_1, \dots, \alpha_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \alpha_{3n}]$ and $[\beta_1, \dots, \beta_{k-1}, \beta_k, \beta_{k+1}, \dots, \beta_{3n}]$, produce two children. For example, $[\beta_1, \dots, \alpha_{k-1}, \beta_k, \beta_{k+1}, \dots, \alpha_{3n}]$ and $[\alpha_1, \dots, \beta_{k-1}, \alpha_k, \alpha_{k+1}, \dots, \beta_{3n}]$ may result, depending on the randomly chosen plane.

The slice operation is carried out as follows. Two parents are chosen as usual, based upon their fitness values. A plane containing the Space Fixed origin is generated by randomly choosing the two spherical polar angles, θ and ϕ . The cosine of the angle θ is chosen flat on -1 to 1, while ϕ is chosen flat on 0 to 2π . These two angles determine the orientation, from the origin, of a vector of length 1. This randomly oriented vector is a normal vector to the plane containing the origin.

Each cluster's center of mass is temporarily translated to the origin and it is determined whether an atom is "above" or "below" the plane by the sign of the dot product of the vector, from the atom to the origin, with the plane's normal vector at the origin. We combine one parent's "top half" with the other parent's "bottom half", and vice-versa. We also assure that each resulting cluster contains the correct number of atoms and add atoms from the opposite "side" of the plane, if necessary. The two resulting child clusters are relaxed using a CG minimization.

Appendix B

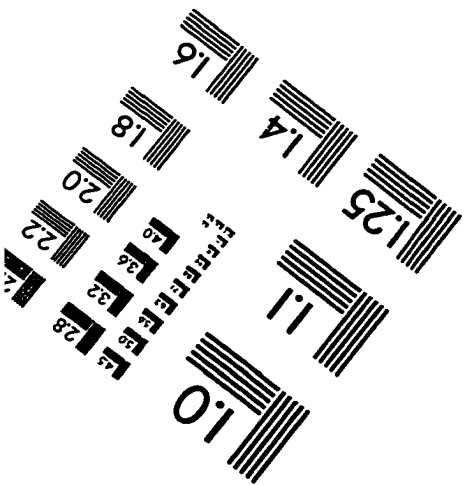
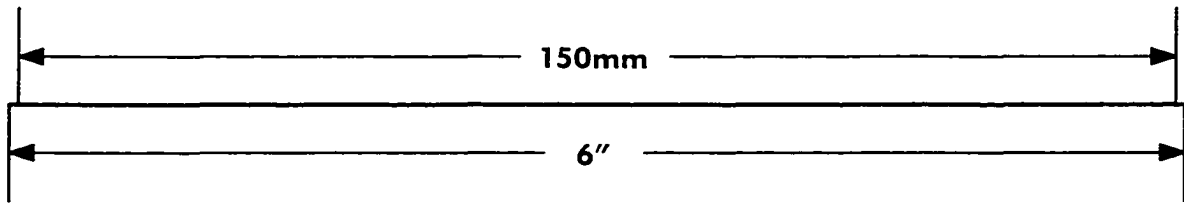
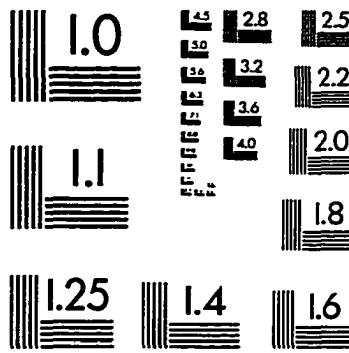
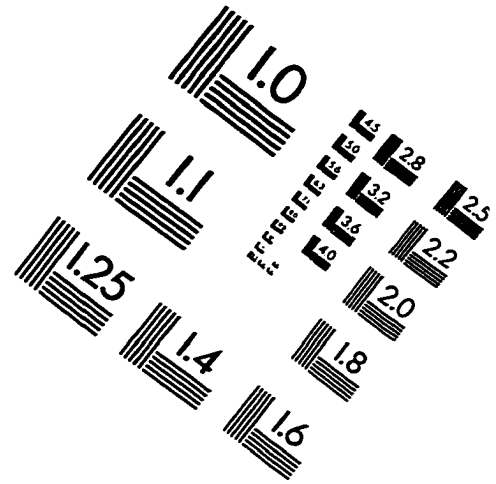
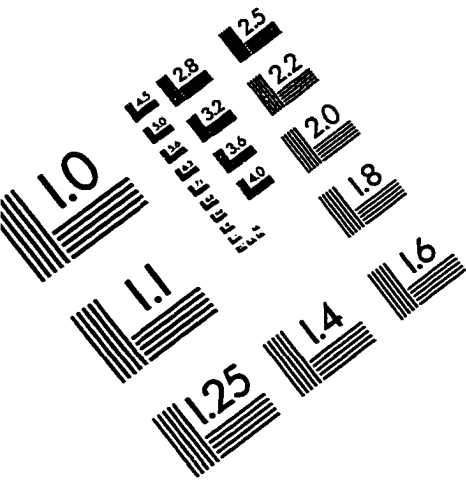
Duplication:

There are several ways in which the above operators can allow duplication of individuals within a population. To be duplicates here means two structures have not only the same potential energy (degenerate), but have the same coordinates as well. The population may become overly weighted with duplicates of the lowest energy structure because it is chosen most often to be a parent (has the highest fitness). A second reason why duplication is undesirable, particularly within populations as small as those used here, is that duplicate parents can exchange information resulting in a child cluster containing two atoms with identical coordinates. We avoid most duplications by preventing certain choices while executing a breeding scheme. The choices we explicitly disallow are:

Inversion:	$k \neq 3n$
1-point crossover:	$k \neq 3n$ and $i \neq j$
2-point crossover:	$k \neq 3n$ and $i \neq j$
N-point crossover:	$i \neq j$ and we ensure that at least one switch occurs.
Arithmetic mean:	$i \neq j$
Geometric mean:	$i \neq j$

However, duplication may still occur through more complicated, but rare, manipulations spanning more than one generation. We have scanned some runs for structures with the same energy within any one generation. We determined if they are duplicates, and not merely degenerate, by simply subtracting corresponding coordinates. A result of zero for each of the $3n$ pairs indicates duplication. We find, after the above restrictions are implemented, that fewer than 0.1% of generations (10 individuals in population, $n = 13$) contained duplicates. As mentioned previously, duplication may result in a child cluster which contains two or more atoms with identical coordinates, causing divide by zero errors upon calculation of the clusters potential. We prevent these errors by artificially setting any $r_{ij}^2 < 8.0 \text{ \AA}^2$ to 8.0 \AA^2 (for LJ clusters) during the calculation of the potential, without actually altering the coordinates themselves. Similar appropriate precautions are taken with the molecular clusters. This avoids a divide-by-zero error, but ensures that the offending structure, unless it is significantly improved during the local minimization, receives an extremely low fitness and is subsequently eliminated from the population.

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

