

Isotropic Effective Energy Simulated Annealing Searches for Low Energy Molecular Cluster States

THOMAS COLEMAN

Department of Computer Science and Advanced Computing Research Institute, Cornell University, Ithaca, NY 14853.

DAVID SHALLOWAY

Section of Biochemistry, Molecular and Cell Biology, Cornell University, Ithaca, NY 14853.

ZHIJUN WU

Advanced Computing Research Institute, Cornell University, Ithaca, NY 14853.

Received December 10, 1992; Revised April 8, 1993

Abstract. The search for low energy states of molecular clusters is associated with the study of molecular conformation and especially protein folding. This paper describes a new global minimization algorithm which is effective and efficient for finding low energy states and hence stable structures of molecular clusters. The algorithm combines simulated annealing with a class of effective energy functions which are transformed from the original energy function based on the theory of renormalization groups. The algorithm converges to low energy states asymptotically, and is more efficient than a general simulated annealing method.

Keywords: global/local minimization, simulated annealing, renormalization group, parallel computation, protein folding

1. Introduction

We are interested in developing an efficient global search algorithm for finding low energy states of molecular clusters of identical atoms. This provides a model for the study of an important, but more difficult, molecular conformation problem in biochemistry—the protein folding problem, i.e., the study of how proteins attain their native spatial structures. The computational approach to this problem is based on the hypothesis that native protein structures correspond to global minima of protein energy. Given a molecular system of n atoms, let $x = \{x_i \in \mathbf{R}^3, i = 1, \dots, n\}$ represent the system structure with each x_i specifying the spatial position of atom i , and let the energy function be defined by $f(x)$ for all x , and

$$f(x) = \sum_{i=1, j>i}^n h_{ij}(\|x_i - x_j\|) \quad (1)$$

where h_{ij} is the pairwise energy function determined by the distance between atoms i and j . Typically, only a small number (≤ 10) of different forms of the h_{ij} are required. A useful model problem (and the one considered in this study) is where $h_{ij} = h$ is identical for all pairs of atoms. In any case, the conformation problem for the given system can be formulated as a global minimization problem:

$$\min_{x \in S} f(x) \quad (2)$$

where S is the set of all possible structures of the system. This problem is extremely difficult because in theory even simple versions are \mathcal{NP} -complete [7], and in practice the energy function usually contains numerous local minima so that the search of global minima can hardly be made efficient even for small problem instances. However because of its great practical importance in biochemistry, the problem has motivated many research projects in computational sciences, especially in the area of numerical optimization. A number of approaches exploiting problem-specific structures and heuristics have been proposed and studied to find efficient solutions to the problem, either deterministic or undeterministic, e.g., [3, 10, 11, 14, 16, 17, 18, 20, 22]. Most approaches have been applied to small problems, but not to the large problems (typically $n = 1,000$ – $10,000$) of biological interest.

In this paper, we present a global search algorithm which is effective and efficient for finding low energy states of a molecular system. The algorithm combines simulated annealing with a class of effective energy functions which are transformed from the original energy function based on the theory of renormalization groups in statistical physics [21]. The algorithm converges to low, or even the lowest, energy states asymptotically, and more efficiently than a general simulated annealing method. It can be used to find low energy states and provide good starting points for local minimization procedures to locate low, or the lowest, energy minima.

The paper is organized as follows. Section 2 describes briefly the simulated annealing algorithm and its asymptotic convergence properties. The effective energy is introduced in Section 3. Section 4 further discusses the isotropic effective energy and its approximation. Section 5 presents our algorithm—the isotropic effective energy simulated annealing. Section 6 describes numerical experiments on the Intel iPSC/860 hypercube computer. Numerical comparisons with a general simulated annealing method are presented. Annealing strategies, application of local minimization, and parallelism are discussed. Section 7 contains concluding remarks.

2. Simulated annealing

In the physical annealing of a solid system, the temperature is first increased to a sufficiently high degree so that the solid melts and then is decreased slowly until the solid solidifies in its stable state where the system energy is

globally minimized. The Metropolis algorithm [13] has been used to simulate the physical process. Using the same philosophy, Kirkpatrick et al. [9] introduced the simulated annealing algorithm to global and combinatorial optimization. Since then, the algorithm has been popular as an approximation method with a variety of applications in solving discrete and continuous optimization problems such as traveling salesman problems, scheduling problems, molecular conformation, etc.

Let x represent the state of a physical system, and $f(x)$ be the system energy at x . The simulated annealing algorithm generates a sequence $\{x\}$ converging to some x^* with minimum energy, simulating the change of the system to the most stable state during the physical annealing process. The sequence is generated in such a way that in each step a new state x_+ is obtained by perturbing the current state x ; x_+ is accepted as the new current state if $f(x_+) \leq f(x)$, or it is accepted with probability

$$e^{-\frac{f(x_+) - f(x)}{k_B T}} \quad (3)$$

where k_B is the Boltzmann constant and T is the system temperature which is lowered gradually as the algorithm proceeds. The algorithm assumes the stochastic model of the physical annealing process, and can be proved to converge asymptotically if the trial sequence generated is sufficiently long.

The simulated annealing algorithm is applied in a similar way to a general optimization problem by regarding its objective function $f(x)$ as the energy function of a hypothetical physical system.

Figure 1 presents a formal description of the algorithm. Note that $\{T_k, 1 \leq k \leq m\}$ is a sequence of temperatures monotonically decreasing to a sufficiently low temperature T_m , and l_T is a bound on the number of trials at temperature T . These values can be prescribed, or determined in each iteration dynamically. The function **random**[0,1) returns a random number in $[0,1)$.

We state the following two facts about the convergence properties of the simulated annealing algorithm without proofs. For detailed convergence analysis, readers are referred to [1].

Fact 1. At each temperature T of simulated annealing, if l_T is sufficiently large, i.e., if sufficiently many trials are made, the probability for the algorithm to reach a system state x with energy $f(x)$ is given by the Gibbs-Boltzmann distribution:

$$p_T(x) = \frac{1}{Z(T)} e^{-\frac{f(x)}{k_B T}} \quad x \in \mathbf{S}, \quad (4)$$

where \mathbf{S} is the set of all possible system states, and $Z(T)$ is the partition function,

$$Z(T) = \sum_{y \in \mathbf{S}} e^{-\frac{f(y)}{k_B T}}. \quad (5)$$

From Fact 1, it is not difficult to show that:

Algorithm 1 {*The simulated annealing algorithm*}

```

0 {Initialization}
   $x = x_{start}$ 
1 {Iteration}
  for  $T = T_1, \dots, T_m$  do {cooling step}
    for  $l = 1, \dots, l_T$  do {random trial}
       $x_+ = x + \Delta x$ 
      if  $f(x_+) \leq f(x)$ 
         $x = x_+$ 
      else if  $e^{-\frac{f(x_+) - f(x)}{k_B T}} > \text{random}[0, 1)$ 
         $x = x_+$ 
      end if
    end do
  end do
end do

```

Figure 1. The simulated annealing algorithm.

Fact 2. Let x^* be a global optimal state with the lowest energy $f(x^*)$. Then the probability for the algorithm to find x^* converges to 1 as temperature T goes to 0.

The simulated annealing algorithm has been used for the study of molecular conformation as well as protein folding, e.g., [11, 20]. The advantages of this algorithm are that it is a general global minimization algorithm, and is easy to implement. However, the success of the algorithm depends strongly on large numbers of trials and hence large numbers of function evaluations. For molecular conformation problems, the cost for each function evaluation usually increases quadratically with problem dimension, i.e., the number of atoms within the molecule. This causes the algorithm to be too expensive for large problems.

3. Effective energy

The algorithm proposed here applies simulated annealing to a class of modified energy functions constructed in accord with renormalization group ideas from

statistical mechanics. More specifically, we consider a new class of energy functions, called effective energy functions, which are transformed from, but converge to, the original energy function. These functions are smoother than the original function and possibly have fewer local minima. As the effective energy functions converge to the original function, they become less smooth, and all minima of the original function are recovered. More importantly, each of these functions reflects the variation of the original function at a coarse level; therefore, its global minima of the effective energy function approximate the global minima of the original function. The approximation becomes exact as the functions converge to their limit.

We apply the simulated annealing algorithm with the energy function replaced by a different effective energy function at each different temperature T . Let the effective energy function converge to the original energy function as T goes to 0. Then the simulated annealing will at each temperature T try to locate the global minimum of corresponding effective energy function, and by tracing the change of the global minimum, the algorithm will hopefully reach the global minimum of the original energy function at the end of the annealing. The entire process here is analogous to the homotopy method for local minimization. Similar ideas can also be found in [10, 17, 18].

The effective energy function was first introduced to molecular conformation by Shalloway [17, 18] in the packet annealing algorithm. In this section, we define the central concepts, and show some of its properties with examples. For more mathematical and physical background of this type of function, readers are referred to [12, 17, 18, 21].

Definition 3.1. Given energy function $f(x)$, for a fixed temperature T and sampling parameter Λ , the effective energy function $\tilde{f}_{\Lambda,T}(x)$ is defined by:

$$e^{-\frac{\tilde{f}_{\Lambda,T}(x)}{k_B T}} = \tilde{p}_{\Lambda,T}(x) = c_{\Lambda} \int e^{-\frac{f(x')}{k_B T}} e^{-\|(x-x')/\Lambda\|^2} dx' \quad (6)$$

where c_{Λ} is a normalization factor such that

$$c_{\Lambda} \int e^{-\|(x-x')/\Lambda\|^2} dx' = 1 \quad (7)$$

Recall that in statistical theory, given a random variable y and its probability distribution $P(y)$, the average value of any function $g(y)$ is

$$\langle g \rangle_P = \int g(y) P(y) dy. \quad (8)$$

So, the left-hand side of (6) is simply the average value of the probability distribution corresponding to the energy when sampled by a Gaussian distribution function centered at point x . Parameter Λ , called the sampling parameter, can represent either a matrix or a scalar, depending on the implementation. It

prescribes the extent of the dominant sampling region and plays an important role. The averaging can be either coarse or fine depending on how large Λ is. As $\Lambda \rightarrow 0$, the average becomes arbitrarily close to the exact function, i.e., for any fixed x and T ,

$$\lim_{\Lambda \rightarrow 0} \tilde{f}_{\Lambda, T}(x) = f(x). \quad (9)$$

For convenience we also state a special case of the effective energy function:

Definition 3.2. Given a pairwise energy function $h(\|y\|)$, for a fixed temperature T the pairwise effective energy function $\tilde{h}_{\lambda, T}(\|y\|)$ for some parameter λ is defined implicitly by the equation

$$e^{-\frac{\tilde{h}_{\lambda, T}(\|y\|)}{k_B T}} = \tilde{p}_{\lambda, T}(\|y\|) = c_\lambda \int e^{-\frac{h(\|y'\|)}{k_B T}} e^{-\|(y-y')/\lambda\|^2} dy' \quad (10)$$

where c_λ is a normalization factor such that

$$c_\lambda \int e^{-\|(y-y')/\lambda\|^2} dy' = 1 \quad (11)$$

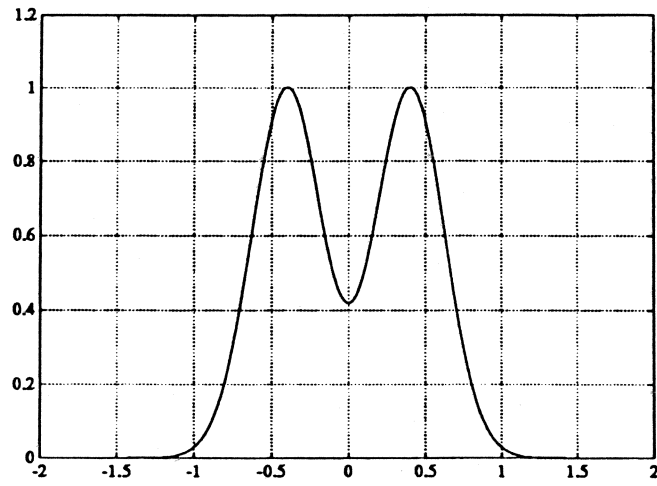
For the pairwise effective energy function, we also have the property that for any fixed y and T ,

$$\lim_{\lambda \rightarrow 0} \tilde{h}_{\lambda, T}(\|y\|) = h(\|y\|). \quad (12)$$

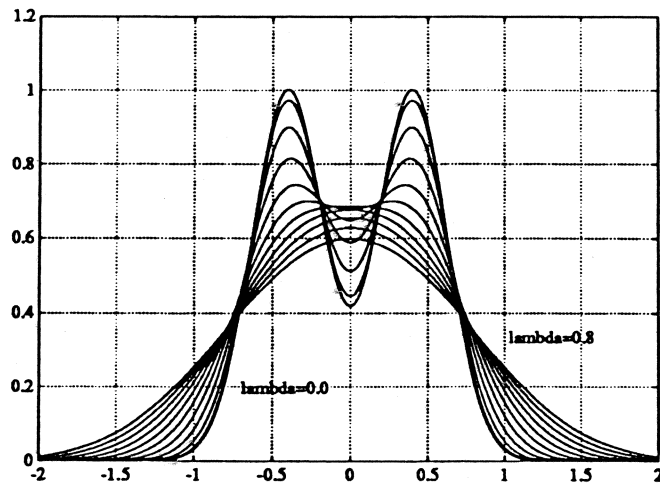
Given $\{\Lambda_k, k = 1, \dots, m\}$, a sequence of sampling parameters, we can define a sequence of effective energy functions $\{\tilde{f}_{\Lambda_k, T}(x), k = 1, \dots, m\}$. When Λ_k is large, the effective energy $\tilde{f}_{\Lambda_k, T}(x)$ estimates coarsely the energy $f(x)$ with all its variations within small regions averaged out, and hence $\tilde{f}_{\Lambda_k, T}(x)$ appears much smoother than $f(x)$. As Λ_k goes to 0, $\{\tilde{f}_{\Lambda_k, T}(x), k = 1, \dots, m\}$ converges to $f(x)$. These properties are illustrated with two examples in Figures 2 and 3. (For convenience, the unnormalized Gibbs-Boltzmann distributions corresponding to the effective energy functions actually are displayed.)

In Figure 2, (a) is the unnormalized probability distribution $p(y) = e^{-f(y)/k_B T}$ for an energy function $f(x)$, and (b) shows the distributions $\tilde{p}_{\Lambda, T}(y)$ for effective energy $\tilde{f}_{\Lambda, T}(y)$ with different Λ values. We see from this example that when $\Lambda = 0$ the function in (b) is exactly the same as in (a), while as Λ increases, the function is deformed and becomes smoother and smoother, and eventually one of the extrema disappears.

Figure 3 shows the behavior of the effective energy in a 2-dimensional example, where (a) is the probability distribution for a 2-dimensional energy function, and (b)–(f) are those for corresponding effective energies with increasing Λ values. Notice that as Λ gets sufficiently large, the graph becomes very smooth, and all four high and steep mountains are removed.



(a)



(b)

Figure 2. A sequence of effective energy probability distributions (b) transformed from (a) at fixed temperature T and different values of λ .

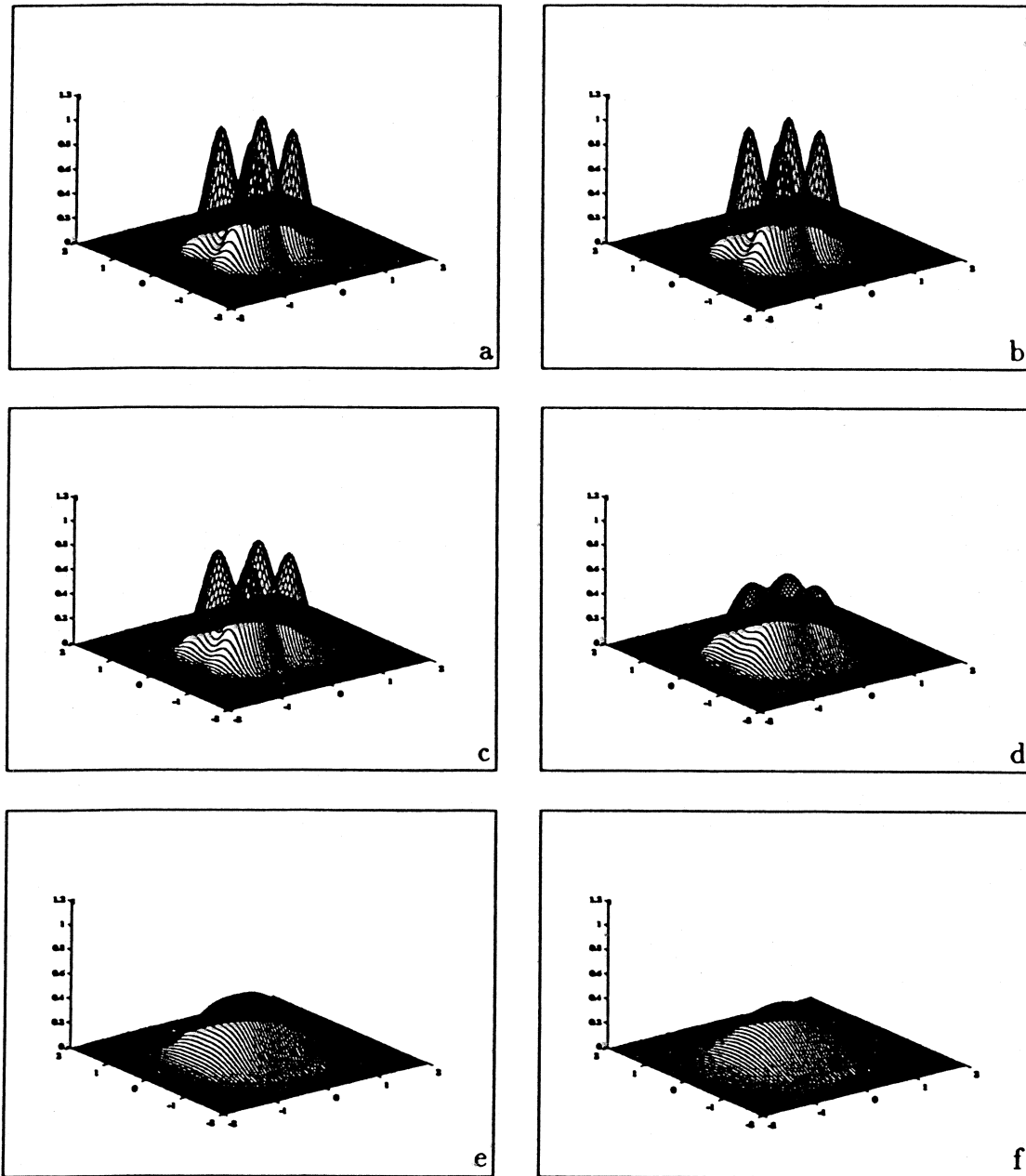


Figure 3. A sequence of effective energy probability distributions (b)–(f) transformed from (a) for increasing values of Λ at constant T .

As can be seen from (6), the lowest effective energy state of an effective energy function will be located in a region that on average has low energy. Physically, it represents a region where the low energy states are most likely to occur; in other words, a region where the system is most likely to be stable in the sense that it can maintain low energy even with small perturbations. As the effective energy function changes to the original energy function, the lowest effective energy state converges to the original lowest energy state. So, the global minimum of effective energy function has both physical and mathematical significance as an asymptotic approximation to the original lowest energy state.

4. Isotropic effective energy and its approximation

In general, the sampling parameter Λ in (6) should be a matrix, which corresponds to a high dimensional anisotropic sampling region. Here, we only consider the case where Λ is a scalar. The effective energy defined this way is called isotropic.

The effective energy is hard to compute because, in general, the high dimensional integral in (6) cannot be calculated either analytically or numerically. Shalloway [17, 18] has suggested the following approximation to the isotropic effective energy:

$$\tilde{f}_{\Lambda, T}(x) \approx \sum_{i=1, j>i}^n \tilde{h}_{\lambda, T}(\|x_i - x_j\|) + c \quad (13)$$

where $\tilde{h}_{\lambda, T}$ is the pairwise effective energy; c is a constant; and $\lambda = g(n, \Lambda, T)$ (g is an unknown function, but $\lim_{\Lambda \rightarrow 0} g = 0$). The formula provides a good approximation when the energy function is close to a quadratic form. For general functions, the approximation could be good or poor, depending on given functions. In any case, to compute the effective energy with this approximation, we only need to calculate all pairwise effective energies, which can be done numerically.

For minimization, the constant c is not important. So, for convenience, we will always refer to the isotropic effective energy as only the first term of (13), the sum of all pairwise effective energies. Let the sum be denoted by $\tilde{f}_{\lambda, T}(x)$. Then

$$\tilde{f}_{\lambda, T}(x) = \sum_{i=1, j>i}^n \tilde{h}_{\lambda, T}(\|x_i - x_j\|). \quad (14)$$

Note that the isotropic effective energy defined this way still maintains the property that for any fixed x and T ,

$$\lim_{\lambda \rightarrow 0} \tilde{f}_{\lambda, T}(x) = f(x) \quad (15)$$

since $\lim_{\lambda \rightarrow 0} \tilde{h}_{\lambda, T}(\|y\|) = h(\|y\|)$ for any fixed y and T . This property guarantees that even if the approximation to the effective energy is poor, we will still get

Algorithm 2 {The IEESA algorithm}

0 {Initialization}

 $x = x_{start}$ $\theta_1 = \theta_{start}$

1 {Iteration}

for $k = 1, \dots, m$ do {cooling step} $T = T_k, \lambda = \lambda_k$ for $l = 1, \dots, l_k$ do {random trial} $x_+ = x + \Delta x = x + \theta_k s$ if $\tilde{f}_{\lambda, T}(x_+) \leq \tilde{f}_{\lambda, T}(x)$ $x = x_+$ else if $e^{-\frac{\tilde{f}_{\lambda, T}(x_+) - \tilde{f}_{\lambda, T}(x)}{k_B T}} > \text{random}(0, 1)$ $x = x_+$

end if

end do

calculate θ_{k+1}

end do

Figure 4. The isotropic effective energy simulated annealing algorithm.

a sequence of functions converging to the original energy function, and our approach will still be applicable.

5. Isotropic effective energy simulated annealing

Figure 4 contains an outline of the isotropic effective energy simulated annealing algorithm (IEESA). At the top level, the algorithm is quite similar to simulated annealing. It decreases the temperature from high temperature T_1 to sufficiently low temperature T_m , and at each temperature T_k , $1 \leq k \leq m$, generates a sequence of random trials. In contrast with simulated annealing, the trials are not performed with the original energy function. Instead, at each T_k , the isotropic effective energy function $\tilde{f}_{\lambda_k, T_k}$ defined by (14) is used:

$$\tilde{f}_{\lambda_k, T_k}(x) = \sum_{i=1, j>i}^n \tilde{h}_{\lambda_k, T_k}(\|x_i - x_j\|). \quad (16)$$

where λ_k , $1 \leq k \leq m$ is a sequence of λ values converging to 0.

Note that Fact 1 in Section 2 for simulated annealing also applies to this algorithm. That is, at each temperature T_k , the probability distribution of the random trials generated by the algorithm is characterized by the Gibbs-Boltzmann distribution with respect to the effective energy $\tilde{f}_{\lambda_k, T_k}(x)$. Therefore, the algorithm can find low effective energy states at each temperature with high probabilities, and at the end, it will find low, or the lowest, energy states for the original energy function with a probability close to 1. (A complete convergence analysis will not be discussed in this paper.)

Usually, if the number of trials is not large enough, the simulated annealing method can be trapped at high local minima. This is less likely for the IEESA algorithm in most cases since it uses effective energy functions which are smoother than the original energy function (when λ is large, many high local minima may even be removed). Even though λ eventually becomes small, and all local minima of the original energy function are recovered, the algorithm still has a better chance of avoiding high local minima traps since it is guided, during the early iterations with large λ and T , away from these regions and toward regions containing many low local minima. For example, if the original energy function contains many deep and narrow local minima, simulated annealing will most likely be restricted to very small random jumps and the search will hardly be sufficient to jump out of local minima. However, the local minima will be deformed shallower and wider on effective energy functions. Therefore, the IEESA algorithm will tend to escape from high energy local minima. Further, if the magnitude of λ is properly matched to T , then each evaluation of the effective energy will provide appropriately averaged information which otherwise is only obtained by multiple evaluations of the original energy function in conventional simulated annealing. Thus IEESA might be more effective and efficient in the case when only a limited number of random trials are allowed. This is always the case in practice because sufficiently large numbers of random trials for large problem instances are not affordable even on today's most powerful supercomputers.

Even if the low energy state found within a limited number of trials by the IEESA algorithm is not the lowest energy state, it is still physically interesting since it lies in a low effective energy region where most nearby states have low energy. Hence it is a stable energy state in the sense that there is a larger probability that the system in this region will maintain low energy even after small perturbations. This is essential because a molecular system, or protein structure, changes dynamically [2], and is most likely to be found in a region highly populated by low energy states. (Using the terminology of statistical mechanics, the system seeks the state of lowest *free energy*.) Thus, it makes more sense to use the effective energy to locate low and stable energy states of molecules.

In practice, besides the quality of the approximation to the effective energy calculation, the performance of the IEESA algorithm depends on other factors such as the cooling schedule $T = T_1, \dots, T_m$, the choice of the sampling parameters $\lambda = \lambda_1, \dots, \lambda_m$, the number of random trials at each cooling step, and the strategy for generating the random jump from x to x_+ , etc. We will address these issues in detail in the following sections.

6. Preliminary numerical experiments

In this section we discuss our numerical experiments with the IEESA algorithm in searching for low energy states of molecular clusters interacting with the Lennard-Jones potential. The algorithm is compared with a general simulated annealing algorithm (SA), which is obtained by setting all λ values to 0 in IEESA. Results for small molecular clusters with 3 to 27 atoms as well as larger ones with 36, 54, and 100 atoms are presented.

The computer system

The IEESA algorithm has been parallelized and implemented on an Intel iPSC/860 hypercube located at Cornell University. The machine we used has 32 processors with eight megabytes local memory for each processor. Our programs are written in C with some extensions for interprocessor communication.

The Lennard-Jones potential

The pairwise energy for a pairwise distance $\|y\|$ was defined in this experiment by the Lennard-Jones potential:

$$h(\|y\|) = \frac{1}{\|y\|^{12}} - \frac{2}{\|y\|^6}, \quad (17)$$

a potential that is widely used in molecular simulations. The function that results when (17) is used in (1) provides a good model for the conformational energy function that governs the behavior of a particularly simple physical system—a molecular cluster (microcluster) of chemically inert atoms (e.g., argon). The coordinates of the global minimizer of this function correspond to the molecular structure that is most stable at very low temperatures. Local minimizers describe metastable structures. The conformations of these types of molecular clusters have been studied by generating polyhedral sequences [8], by stochastic approaches [3, 20], by searching special lattice structures [14, 22] and by some deterministic methods [10, 17, 18]. These results provide standards for testing new algorithms.

Effective energy evaluation

Efficient effective energy evaluation is important for the IEESA algorithm since thousands of function evaluations usually are required. With (14), the effective energy is computed by summing all pairwise effective energies defined implicitly in (10). As described in [17], the right-hand side of that equation can be reduced to a 1-dimensional integral:

$$\frac{c\lambda\pi\lambda^4 e^{-\|y\|^2/\lambda^2}}{\|y\|} \int_0^1 e^{-h[\lambda\sqrt{\log(1/v)}]} \sinh \left[\frac{2\|y\|\sqrt{\log(1/v)}}{\lambda} \right] dv. \quad (18)$$

that can be calculated by numerical integration.

In our implementation, to reduce the cost for computing the pairwise effective energy, a function value look-up table first is constructed for $\tilde{h}_{\lambda,T}(\|y\|)$ at different T , λ and $\|y\|$ values. Then, function value requests by the algorithm are obtained by cubic spline interpolation. Using this special way of evaluating the effective energy, the cost for each individual function evaluation is efficient and the same for both IEESA and SA.

Generating random jumps

Given $x = \{x_i \in \mathbf{R}^3, i = 1, \dots, n\}$, a state of a given molecular system of n atoms, a random jump to x_+ from x is generated with:

$$x_+ = x + \Delta x, \quad (19)$$

where $\Delta x = \theta s$ for some step size θ and random step $s = \{s_i \in \mathbf{R}^3, s_i = \pm \text{random}[0, 1]^3, i = 1, \dots, n\}$. Note that the step size determines the acceptance rate of trial sequences (see discussion in [20]). The scalar θ is adjusted adaptively at each cooling step so that 25%–50% of total random jumps can be accepted.

The number of random trials

In general, we set l_k , the maximum number of random trials at the k th cooling step, to be a polynomial function of n . The number also is made proportional to $(1 - \log(T_k))$ (as in [20]) to provide more trials at low temperatures. More specifically, we always let l_k be determined by

$$l_k = n \log(n)(1 - \log(T_k)), \quad (20)$$

except for cases where even more trials are necessary.

The cooling schedule and sampling parameters

The cooling schedule $T = T_1, \dots, T_m$ as well as the choice of sampling parameters $\lambda = \lambda_1, \dots, \lambda_m$ are crucial to the performance of our algorithm. A simple strategy to determine these values is to choose them uniformly between given $T_1(\lambda_1)$ and $T_m(\lambda_m)$, i.e., for $1 \leq k \leq m$,

$$T_k = T_1 - \frac{k-1}{m-1}(T_1 - T_m), \quad (21)$$

$$\lambda_k = \lambda_1 - \frac{k-1}{m-1}(\lambda_1 - \lambda_m), \quad (22)$$

where T_m and λ_m are set to 0; and T_1 , λ_1 , and m are chosen arbitrarily, or based on some preliminary experiments. This strategy is simple and straightforward, but to find proper T_1 , λ_1 , and m , many experiments have to be conducted. Moreover, good choices for such sequences may not necessarily be uniformly spaced. For example, a typical simulated annealing procedure reduces temperature T with $T_{k+1} = \chi T_k$, $k = 1, \dots, m-1$, for some χ , $0 < \chi < 1$ [1, 20].

In our experiments, we generated the temperature sequence $\{T_k, k = 1, \dots, m\}$ with the formula

$$T_{k+1} = \chi T_k, \chi = 0.8, k = 1, \dots, m-1 \quad (23)$$

where $T_1 = 1.0$; and $m = 22$ so that $T_m = 0.009$ was sufficiently close to 0. Also, the sampling parameters $\{\lambda_k, k = 1, \dots, m\}$ are calculated to adapt to the changes of step sizes of random jumps:

$$\lambda_k = \alpha \theta_k \sqrt{3}, k = 1, \dots, m \quad (24)$$

where α is a constant; θ_k is the step size of random jumps at the k th cooling step; and the product $\theta_k \sqrt{3}$ corresponds to the maximum distance that each individual atom might jump. The reason for using (24) is that we want the sampling parameter λ_k to be roughly proportional to, though smaller than, the random jump scale (i.e., the maximum value of $\|\Delta x\|$ for all Δx generated at the k th cooling step), which is proportional to θ_k by (19). In this way, the random search conducted by the algorithm will sample the "long wavelength" behavior of the system after the "short wavelength" information (i.e., fluctuations of spatial extent $< \lambda_k$) have been averaged out.

Experiments for $3 \leq n \leq 27$

We have conducted experiments on searches for low energy states of the Lennard-Jones molecular clusters with the IEESA algorithm. Clusters with up to 27 atoms have first been considered. The algorithm has been compared with a simulated annealing method (SA) by comparing energy levels of final states found by both algorithms within the same number of trials (fixing the number of trials at each cooling step by (20)). All results were obtained on an Intel iPSC/860 hypercube using 16 of the 32 available processors.

The IEESA algorithm performs differently when α in (24) is set to different values, and particularly yields just the results of SA when $\alpha = 0$. To be "effective," the algorithm must have $\alpha > 0$. A good choice of $\alpha > 0$ also is critical. A typical situation is that with a limited number of trials, the energy level of the final state decreases with increasing α from 0, and starts increasing when α is beyond a certain point.

Let the energy level of the final state found by the algorithm for a cluster of n atoms be denoted by E_n , then $E_n = E_n(\alpha)$ is a function of α . $E_n(0)$ is the energy level reached by SA. To make the comparison between IEESA and SA, we applied the algorithm for all $3 \leq n \leq 27$ with $\alpha = 0, 0.1, 0.2, \dots$, etc. Then we obtained a set of function values for $E_n(\alpha)$. (Actually, the process was repeated five times, and the average value of $E_n(\alpha)$ was recorded.) Figure 5 shows typical $E_n(\alpha)$ functions ($n = 8, 10, 12, 14, 16, 18, 20, 22$) obtained from these experiments. The bottom line of each graph corresponds to the energy level of the best known solution based on previous studies by [8, 14, 20]. All $E_n(\alpha)$ functions appear roughly parabolic: they start at $\alpha = 0$ with relatively high values, and decrease with increasing α ; at some $\alpha > 0$, they achieve their lowest values, which are very close to the bottom lines of the graphs. This implies that with a limited number of trials, SA can only find relatively high level energy states of tested clusters, while with proper choices of α , those found by IEESA are at very low energy levels. The difference can further be observed in Table 1, where the lowest energy levels reached by SA and IEESA are listed. Figure 6 shows differences between the two sets of energy levels. Apparently, IEESA outperforms SA for all tested clusters.

Extension to larger problems

For the Lennard-Jones clusters with $n > 27$, we empirically approximate the optimal value of α as a function of n . Let the function be denoted by $\alpha^*(n)$. As we have discussed above, the sampling parameter λ_k is chosen to be proportional to the random jump scale—the maximum value of $\|\Delta x\|$ for all Δx generated at the k th cooling step. By (19) this value is equal to $\theta_k \sqrt{3n}$. Thus

$$\lambda_k \propto \theta_k \sqrt{3n}. \quad (25)$$

Comparing (25) with (24), we must have $\alpha \propto \sqrt{n}$. Because of this and also the empirical fact that $\alpha^*(3) \approx 0$, we choose our approximation function to be

$$\alpha^*(n) = c\sqrt{n-3} \quad (26)$$

where c is a constant to be determined. Using a least squares calculation,

$$c = \frac{\sum_{i=3}^{27} \alpha_i^* \sqrt{i-3}}{\sum_{i=3}^{27} (i-3)} = 0.3413, \quad (27)$$

where α_i^* is the empirical values for $\alpha^*(i)$, $3 \leq i \leq 27$, shown in Table 1. Then we obtain

$$\alpha^*(n) = 0.3413\sqrt{n-3} \quad (28)$$

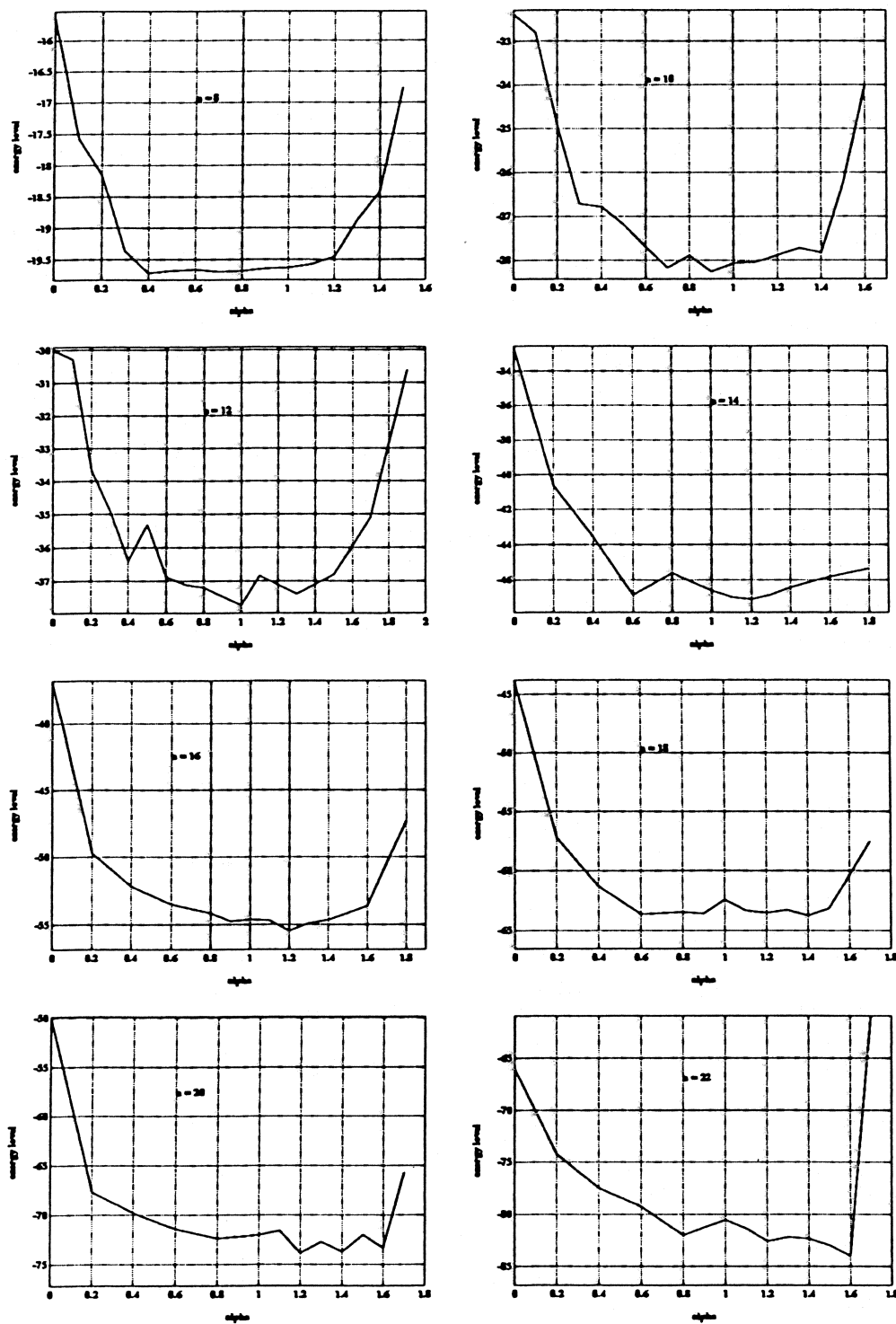


Figure 5. Low energy levels reached by IEEA for different α values.

Table 1. Low energy values obtained by IEESA and SA with adaptive reduction for molecular clusters interacting with the Lennard-Jones potential.

<i>Low Energy Levels Found by IEESA and SA</i>					
# of atoms	best-known	IEESA		SA	
		energy level	α	energy level	α
3	-3.000000e + 00	-2.9981e + 00	0.1e0	-2.9977e + 00	0.0e0
4	-6.000000e + 00	-5.9892e + 00	0.4e0	-5.7018e + 00	0.0e0
5	-9.104000e + 00	-9.0817e + 00	0.4e0	-8.5161e + 00	0.0e0
6	-1.271200e + 01	-1.2638e + 01	0.5e0	-1.0974e + 01	0.0e0
7	-1.650500e + 01	-1.6414e + 01	0.7e0	-1.4253e + 01	0.0e0
8	-1.982200e + 01	-1.9692e + 01	0.7e0	-1.5635e + 01	0.0e0
9	-2.411300e + 01	-2.3974e + 01	0.8e0	-1.8768e + 01	0.0e0
10	-2.842000e + 01	-2.8256e + 01	0.9e0	-2.2390e + 01	0.0e0
11	-3.276500e + 01	-3.2479e + 01	0.7e0	-2.4369e + 01	0.0e0
12	-3.796700e + 01	-3.7730e + 01	1.0e0	-3.0004e + 01	0.0e0
13	-4.432700e + 01	-4.3814e + 01	1.4e0	-2.9484e + 01	0.0e0
14	-4.784500e + 01	-4.7091e + 01	1.2e0	-3.2643e + 01	0.0e0
15	-5.232300e + 01	-5.1220e + 01	1.1e0	-3.5032e + 01	0.0e0
16	-5.681600e + 01	-5.5457e + 01	1.2e0	-3.6883e + 01	0.0e0
17	-6.131800e + 01	-5.9391e + 01	1.2e0	-4.1419e + 01	0.0e0
18	-6.653100e + 01	-6.3742e + 01	1.4e0	-4.4888e + 01	0.0e0
19	-7.266000e + 01	-6.8408e + 01	1.5e0	-4.5110e + 01	0.0e0
20	-7.717700e + 01	-7.3858e + 01	1.2e0	-4.9980e + 01	0.0e0
21	-8.168500e + 01	-7.7983e + 01	1.4e0	-6.3856e + 01	0.0e0
22	-8.681000e + 01	-8.4000e + 01	1.6e0	-6.6034e + 01	0.0e0
23	-9.284400e + 01	-8.8037e + 01	1.6e0	-7.2082e + 01	0.0e0
24	-9.734900e + 01	-9.2708e + 01	1.7e0	-7.7758e + 01	0.0e0
25	-1.023730e + 02	-9.7823e + 01	1.9e0	-8.2202e + 01	0.0e0
26	-1.083160e + 02	-1.0214e + 02	1.2e0	-9.2653e + 01	0.0e0
27	-1.128740e + 02	-1.0767e + 02	1.7e0	-9.8123e + 01	0.0e0

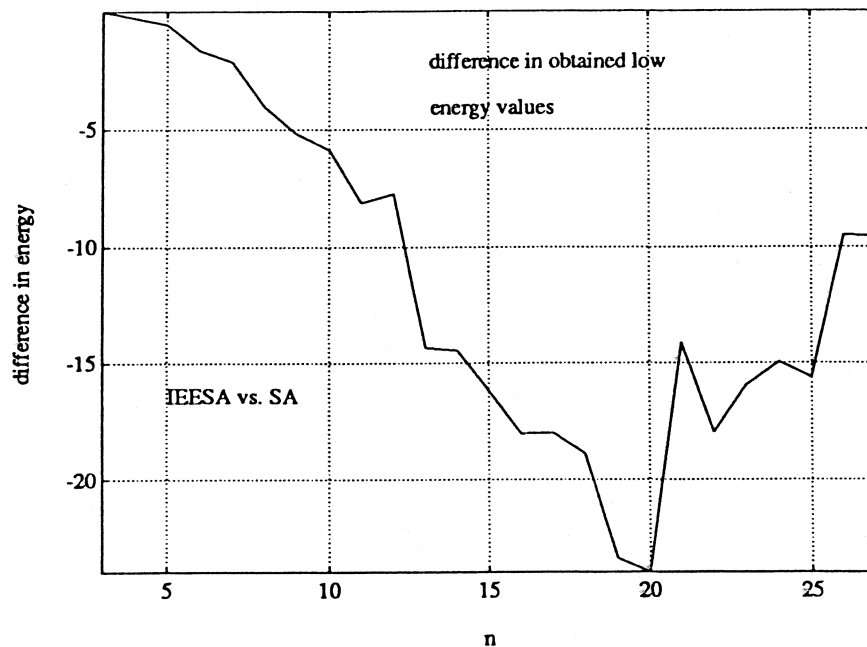


Figure 6. Differences in energy values obtained by IEESA and SA.

Figure 7 shows the empirical values α_n^* and the predicted values $\alpha^*(n)$ for $n = 3$ to 27. The predicted function fits the empirical data well.

With (28), we can approximate the optimal choice for α for any $n > 27$. Table 2 contains some results from applying IEESA to the Lennard-Jones clusters with $n = 36, 54, 100$, where α values are obtained from (28). Although the energy levels reached by IEESA are not very close to the best-known results, they are much better than those reached by SA. The performance may be further improved by testing α values around $\alpha^*(n)$ to find the optimal choice.

Increasing numbers of trials

The experiments described above were conducted with the number of trials at temperature T limited by $l_T = n \log(n)(1 - \log(T))$. Figure 8 illustrates the behavior of IEESA and SA as the number of trials increases. Typically, with larger numbers of trials, the energy levels reached by SA tend toward those that were reached by IEESA with smaller numbers of trials. That is, IEESA performs significantly better. Note that the way we increased the number of trials was to set $l_T = n \log(n)(1 - r \log(T))$ for $r = 1, 2, 3$, etc.

Exploiting parallelism

There are at least two reasons to exploit parallelism for the IEESA algorithm. First, the computation can be very intensive when the algorithm is applied to

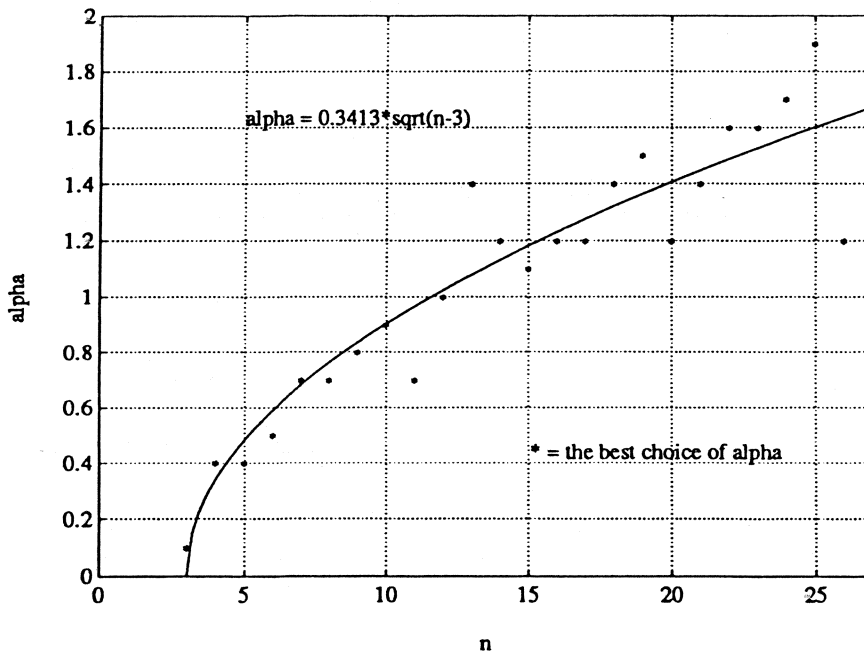


Figure 7. The least squares fit of function $\alpha^*(n)$.

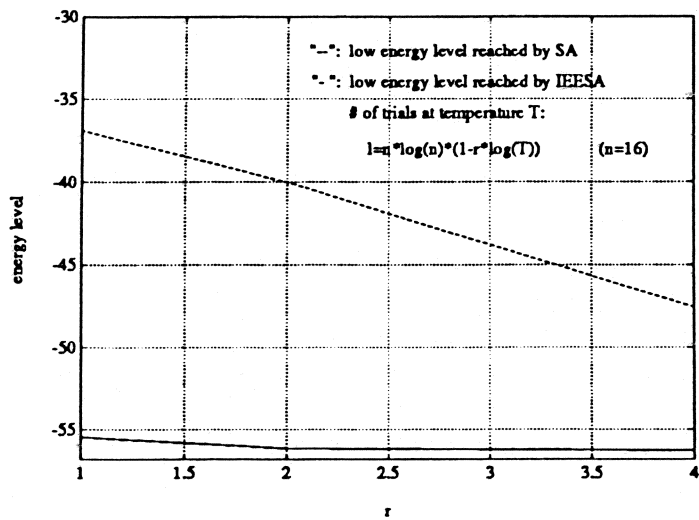
Table 2. Low energy levels reached by IEESA and SA for $n = 36, 54, 100$.

Low Energy Levels Found by IEESA and SA					
# of atoms	best-known	IEESA		SA	
		energy level	α	energy level	α
36	$-1.618250e + 02$	$-1.5203e + 02$	1.9606e0	$-1.4088e + 02$	0.0e0
54	$-2.722090e + 02$	$-2.5088e + 02$	2.4374e0	$-2.3464e + 02$	0.0e0
100	$-5.570400e + 02$	$-4.9125e + 02$	3.3614e0	$-4.6191e + 02$	0.0e0

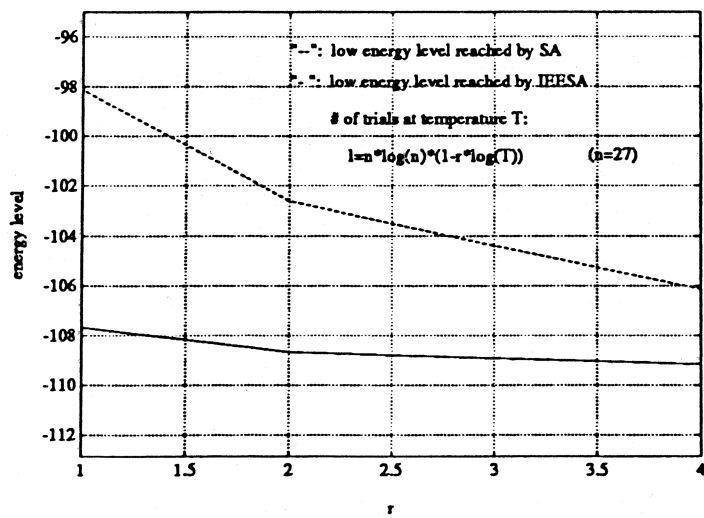
large problems, and the speed-up from parallel computation is extremely helpful. Second, the algorithm is easy to parallelize, and the parallel efficiency can be made close to 100%.

The algorithm has been parallelized with the following simple strategy: with p processors, the algorithm generates p trial sequences at each cooling step; then the total number of accepted trials is calculated across all processors; based on this number, the step size for random jumps is adjusted; multiple trial sequences are generated at next cooling step with the new step size; the process goes on until the last cooling step, where the energy states found by all processors are compared and the best one is chosen as the final solution.

The parallel efficiency of the algorithm is close to ideal because all processors generate their own trial sequences independently with little communication. In



(a)



(b)

Figure 8. The performance of IEESA and SA with increasing numbers of random trials.

addition, they make the same number of trials, and hence the computation load is perfectly balanced. We have studied the performance of IEESA for varying p (the number of processors). Usually, as shown in Figure 9, when the number of processors is increased, the energy level reached by the algorithm is decreased significantly, while the computation time increases little. This suggests that IEESA could be used for solving large molecular conformation problems on massively parallel computers.

Including local minimization

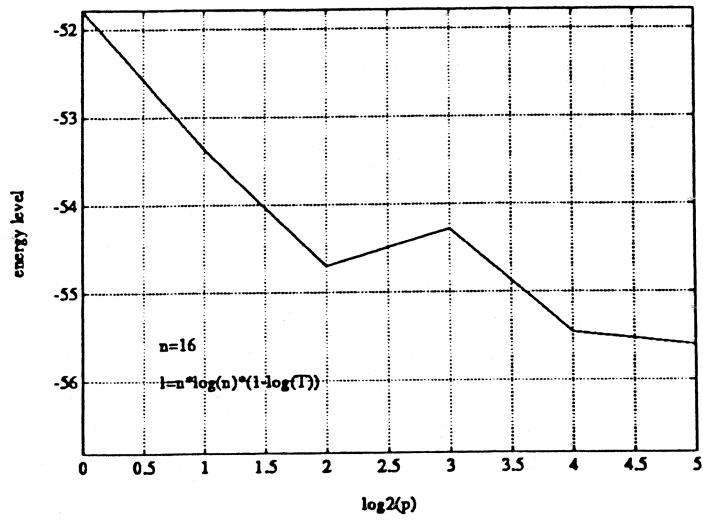
Both IEESA and SA can find global energy minima of molecular clusters when sufficiently many random trials are allowed. However, in practice it would be better to use them to only provide low energy states in fixed computation time as we have done in the above experiments. Then by using local minimization procedures with these energy states as starting points lower energy minima can be located more efficiently.

A simple way to integrate the local minimization procedure into IEESA or SA is to apply the procedure at the very end of the IEESA or SA process. In this case, the low energy state found by IEESA or SA is taken as the starting point for the local minimization. However, for IEESA, the local minimization can also be used at the end of *each* cooling step so that lower effective energy states can be found.

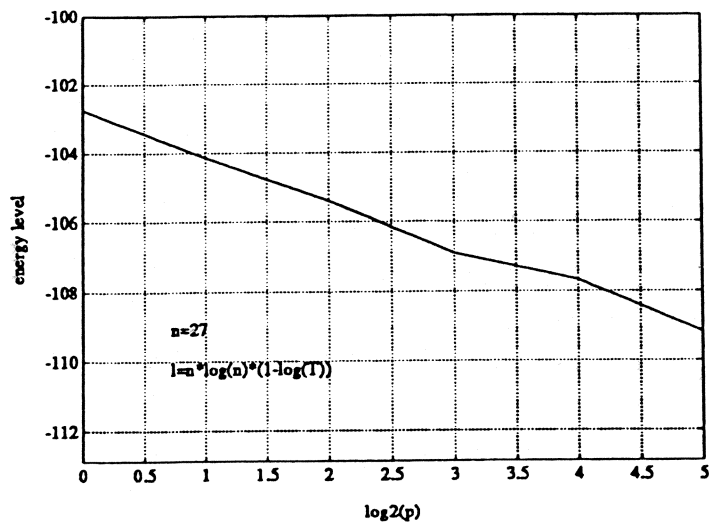
We have implemented both of these approaches using a line search quasi-Newton's method based on [5] as a local minimization procedure. Thus, three integrated algorithms were studied: IEESA and SA with local minimization at the very end and IEESA with local minimization at the end of *each* cooling step. We denote these methods by IEESA', SA', and IEESA'', respectively. Table 3 shows results from applying the algorithms to the Lennard-Jones molecular clusters with $n = 3$ to 27 atoms. For each n , five runs were made for all three algorithms. A + sign is marked if the lowest energy state was found, and a - sign otherwise. Note that α values for IEESA' and IEESA'' were the same as those (best choices) for IEESA in Table 1, and the values for SA' were 0. Also the number of random trials at each cooling step still was determined by (20).

From Table 3 we see that in most cases, SA' did not find the lowest energy states, but IEESA' did. While IEESA' did not find solutions for some clusters, IEESA'' found them for all. Furthermore, IEESA'' found solutions for most clusters in almost all five runs. So IEESA'' apparently performs the best among three algorithms.

Finally, we also applied IEESA'' to some larger clusters, e.g., containing 36, 54, and 100 atoms (Table 4). For the clusters of 36 and 54 atoms, the algorithm found the global solutions twice in five runs. We only made one run for the cluster of 100 atoms since it was very time-consuming. The number of trials



(a)



(b)

Figure 9. The performance of IEESA when different number of processors are used.

Table 3. Searches for global energy minima for the Lennard-Jones molecular clusters by IEESA'', IEESA', and SA'. Notation: n , number of atoms; p , number of processors; +, the lowest energy state found; -, the lowest energy state not found.

<i>Searches for Global Energy Minima</i>					
n	lowest energy	ieesa''	ieesa'	sa'	p
3	-3.000000e + 00	+++++	+++++	+++++	1
4	-6.000000e + 00	+++++	+++++	+++++	1
5	-9.103852e + 00	+++++	+++++	+++++	1
6	-1.271206e + 01	+++++	-----	-----	1
7	-1.650539e + 01	+++++	+ - + - -	-----	2
8	-1.982149e + 01	- - - + -	+++++	+ + - + +	2
9	-2.411336e + 01	+++++	- - + + +	- - - + -	2
10	-2.842254e + 01	+ + + - +	+ - - + -	-----	2
11	-3.276597e + 01	+++++	+ + - + +	- + - - -	4
12	-3.796761e + 01	+++++	+++++	- - + - -	4
13	-4.432681e + 01	+++++	+++++	- + - + -	4
14	-4.784517e + 01	+++++	+++++	- - - + -	4
15	-5.232265e + 01	+++++	+ - + + +	+ - - - -	4
16	-5.681575e + 01	+++++	+++++	+ - - - -	8
17	-6.131801e + 01	- + + - -	- + - - -	- + - - -	8
18	-6.653097e + 01	- + + + +	+ + - - +	-----	8
19	-7.265979e + 01	+++++	- - + + +	-----	8
20	-7.717701e + 01	+++++	- - + - -	-----	8
21	-8.168460e + 01	+++++	- - + - +	-----	16
22	-8.680981e + 01	+++++	+ - + + +	-----	16
23	-9.284451e + 01	+++++	-----	-----	16
24	-9.734884e + 01	+++++	- + - - -	-----	16
25	-1.023727e + 02	+++++	- + - + +	-----	16
26	-1.083156e + 02	+++++	-----	-----	16
27	-1.128736e + 02	+ - + - +	- + - - -	-----	16

Table 4. Searches for global energy minima for the Lennard-Jones molecular clusters by IEESA". Notation: n , number of atoms; p , number of processors; +, the lowest energy state found; -, the lowest energy state not found.

Searches for Global Energy Minima			
n	36	54	100
lowest energy	$-1.618250e + 02$	$-2.72209e + 02$	$-5.570400e + 02$
ieesa"	- - + - +	- + - + -	$-5.379205e + 02$
p	32	32	32

determined by (20) seemed insufficient for this instance, and hence only a local (but not global) energy minimum was found.

7. Concluding remarks

We have presented a new global minimization algorithm for finding the low energy states of molecular clusters. The algorithm combines simulated annealing with a class of effective energy functions transformed from the original energy function using renormalization group ideas from statistical mechanics. The effective energy functions appear to be smoother than the original energy function with possibly fewer local minima. So the algorithm works on these "easier" functions, first to locate low effective energy states, and then to trace their changes to low energy states as the effective energy function is changed gradually to the original energy function.

The algorithm (with and without local minimization) has been parallelized and implemented on an Intel iPSC/860 hypercube computer. Numerical experiments have been done with the algorithm applied to molecular clusters of identical atoms interacting with the Lennard-Jones potential. Small clusters with 3 to 27 atoms as well as larger ones with 36, 54, and 100 atoms have been tested. The algorithm was significantly more effective and efficient than those obtained with conventional simulated annealing.

Strategies for choosing proper sampling parameters for the effective energy functions have been proposed. Experiments show that with proper choices of these parameters, the algorithm performs well. A method for adaptively determining good sampling parameters has been demonstrated.

While the algorithm is promising as a feasible approach to large molecular conformation, many lines of investigation can be pursued to further improve performance. First, the current algorithm still is costly in terms of computational complexity in function evaluation. Usually, at each cooling step, $O(n \log(n))$ function evaluations are conducted. Each function evaluation costs $O(n^2)$ op-

erations since we at least need to calculate $\mathcal{O}(n^2)$ pairwise distances. So the total function evaluation for the whole algorithm costs $\mathcal{O}(n^3 \log(n))$ operations. This may require too much time when n is large, say, 1,000. One way to reduce the complexity might be to reduce the cost for each function evaluation to $\mathcal{O}(n \log(n))$ by doing a partial update for each random jump, e.g., only perturbing the positions of a subset of atoms.

Second, the isotropic effective energy averages the energy variation equally along all directions in the sampling space. But the variation itself should be different along different directions. A better algorithm should use the effective energy functions with the sampling parameters Λ as matrices (i.e., anisotropic effective energy functions). However, the algorithm will be complicated by the need to compute the sampling matrices.

Third, to date the algorithm has only been applied to, although it is not specific to, molecular clusters of identical atoms interacting with the Lennard-Jones potential. For application to protein folding, we will consider some real proteins of relatively small sizes, which have different types of atoms and more complicated potential functions.

Acknowledgments

This research was supported partially by the Cornell Theory Center, which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute, and supported, in part, by a grant from the National Institutes of Health (1P01GM48874), and by NSF, AFOSR, and ONR through NSF grant DMS-8920550.

References

1. Emile Aarts, and Jan Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons: New York, NY, 1989.
2. Charles L. Brooks III, Martin Karplus, and B. Montgomery Pettitt, *Proteins: A Theoretical Perspective of Dynamics, Structure, and Thermodynamics*, John Wiley & Sons: New York, NY, 1988.
3. Richard H. Byrd, Elizabeth Eskow, Robert B. Schnabel, and Sharon L. Smith, "Parallel Global Optimization: Numerical Methods, Dynamic Scheduling Methods, and Application to Molecular Configuration," Technical Report CU-CS-553-91, (1991), Department of Computer Science, University of Colorado at Boulder, Boulder, CO.
4. Thomas F. Coleman, "Large-scale numerical optimization: Introduction and overview," in *Encyclopedia of Computer Science and Technology*, Williams and Kent, eds. Marcel Dekker: New York, NY, (1993), 167-195.
5. J.E. Dennis Jr., and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall: Englewood Cliffs, NJ, 1983.
6. Philip E. Gill, Walter Murray, and Margaret H. Wright, *Practical Optimization*, Academic Press: London, 1981.

7. Bruce A. Hendrickson, "The molecular problem: Determining conformation from pairwise distances," Ph.D. Thesis, (1991), Department of Computer Science, Cornell University, Ithaca, NY.
8. M.R. Hoare, "Structure and dynamics of simple microclusters." *Adv. Chem. Phys.* **40**, (1979), 49-135.
9. S. Kirkpatrick, C.D. Gellat, Jr., and M.P. Vecchi, "Optimization by simulated annealing," *Sci.* **220**, (1983), 671-680.
10. J. Kostrowicki, L. Piela, Binny J. Cherayil, and Harold A. Scheraga, "Performance of the diffusion equation method in searches for optimum structures of clusters of Lennard-Jones atoms," *J. Phy. Chem.* **95**, (1991), 4113-4119.
11. Z. Li, and Harold A. Scheraga, "Monte Carlo approach to the multiple-minima problem in protein folding," *Proceedings of National Academy of Science, USA*, **84**, (1987), 15-29.
12. Shang-keng Ma, "Introduction to the renormalization group," *Rev. of Modern Physics* **45**, (1973), 589-614.
13. N. Metropolis, A Rosenbluth, M. Rosenbluth, A Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phy.* **21**, (1953), 1087-1092.
14. J.A. Northby *Structure and binding of Lennard-Jones clusters: $13 \leq n \leq 147$* . *J. Chem. Phy.* **87**, (1987), 6166-6178.
15. P.M. Pardalos, and J.B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, Springer-Verlag: New York, NY, 1987.
16. L. Piela, J. Kostrowicki, and Harold A. Scheraga, "The multiple-minima problem in the conformational analysis of molecules. Deformation of the protein energy hypersurface by the diffusion equation method," *J. Phy. Chem.* **93**, (1989), 3339-3346.
17. David Shalloway. "Packet annealing: A deterministic method for global minimization. Application to molecular conformation," in *Global Optimization*, C. Floudas and P. Pardalos, eds. Princeton University Press: Princeton, NJ, 1992.
18. David Shalloway, "Application of the renormalization group to deterministic global minimization of molecular conformation energy functions," *Global Optimization* **2**, (1992), 281-311.
19. Aimo Törn, and Antanas Zilinskas, *Global Optimization*, Springer-Verlag: New York, NY, 1987.
20. L.T. Wille, "Minimum-energy configuration of atomic clusters: New results, obtained by simulated annealing," *Chem. Phy. Letters* **133**, (1987), 405-410.
21. K.G. Wilson, "The renormalization group: Critical phenomena and the Kondo problem," *Rev. of Modern Phy.* **47**, (1975), 773-840.
22. Guo-liang Xue, *Improvement on the Northby algorithm for molecular conformation: Better solutions*, (1992), Army High Performance Computing Research Center, University of Minnesota, MN.