A genetic algorithm based technique for locating first-order saddle point using a gradient dominated recipe

Pinaki Chaudhury [a], S.P. Bhattacharyya [a,*], W. Quapp [b]

[a] *Department of Physical Chemistry, Indian Association for the Cultivation of Science, Jadavpur, Calcutta 700032, India*
[b] *Mathematical Institute, Leipzig University, Augustusplatz, D-04109 Leipzig, Germany*

Abstract

The present paper proposes and tests the workability of a genetic algorithm based search technique for locating the first-order saddle points on the potential energy surfaces of Lennard-Jones clusters with $n \leqslant 30$. A modified objective function using gradients only is used to locate the saddle points. The cost effectiveness of the proposed method vis-a-vis that of an earlier formulation, where an explicit determination of Hessian eigenvalues was required, is demonstrated. The method could be useful in the construction and the analysis of the reaction paths in complex systems. © 2000 Elsevier Science B.V. All rights reserved.

1. Introduction

Finding the critical points of Lennard-Jones (LJ) clusters is a problem which has received little attention in the recent years. Several workers have strived to locate critical points, especially the global minimum of these systems with a host of techniques. The simple fact as to why these systems are interesting and challenging to study is that, as the size of these clusters increases, the number of critical points increases at an astronomical rate. A concrete example will make things more convincing. A cluster of moderate size such as $n = 13$ is known to possess at least 1328 minima [1–4], and for a really large cluster with $n = 147$ the number is 10^{60}, to say nothing of characteriz-

ing the transition states that connect these minima. Hence, it is clear that locating the global structure among the myriad possibilities is not an easy task. A number of excellent applications have come out over the last decade in this field. Hoare and Pal [5,6] developed a general growth algorithm, and used it to generate large numbers of stable structures ($n \leqslant 55$). Northby [7] used an efficient lattice-based search to find the optimal structure in the range of $13 \leqslant n \leqslant 147$. Genetic Algorithm (GA) itself has been used in this field by several workers with encouraging results [8–12]. Wales and Doye [13] used a "basin hopping" technique to arrive at global minimum energy structures of different sizes, upto $n = 110$. They succeeded in finding in some cases structures that had energies lower than what were previously described as the lowest energy structures of the clusters concerned.

Other than the global minimum, a first-order saddle point (SP) is another important structure, which the topography of a LJ cluster supports

---
* Corresponding author. Fax: +91-33-4732805.
E-mail address: pcspb@mahendra.iacs.res.in (S.P. Bhatta-charyya).

[14,15]. A saddle point of order 1 is defined as a critical point where only one of the eigenvalues of the Hessian matrix is negative. Their importance lies in the fact that the associated structures may often define the transition state structures on the minimum energy path of the structural transformations of the clusters. SPs are in general difficult to locate as little can be guessed about their location and disposition to start with. In LJ clusters, the problem is sometimes compounded by the relatively flat nature of the potential energy surface around the saddle points [15]. Nevertheless, studies on SPs of LJ clusters have also received attention. Wales [14] located SPs in LJ clusters of various sizes by the eigenvector following method. Quapp [15] located SPs in an Ar_4 cluster by a quasi-gradient-only technique. There has been an extensive study of the SPs of Ar_8 cluster by Jensen [16]. The paper reported an identification of 42 SPs. Simulated annealing and GA based techniques for locating the SPs and constructing reaction paths have also been proposed recently [12,17]. In our previous studies, the objective function contained information about the local gradients and curvatures, although the search was based on a stochastic optimizer, namely the simulated annealing method (SAM).

Our purpose in this paper has been to explore the usefulness of a stochastic all purpose global optimizer known as GA [18–20] in the SP search, avoiding diagonalization of the full Hessian all through the search. Our argument for using a stochastic method rather than a conventional deterministic algorithm lies in the topography of the LJ surface. The presence of a large number of minima can stall the progress of any deterministic algorithm when the target of the search is the global minimum. It is here that stochastic search algorithms can be quite useful as they are capable of exploring the search space globally. For a given starting point, the deterministic algorithm, on the other hand, can search the space locally.

In the following sections, we give an account of the methodology proposed, including the operational details of the algorithm, and then, present the results of a few applications we have made.

2. Methodology

2.1. The gradient-only scheme

The methodology that we present below has been adapted to GA. However, the same technique can be adapted to another stochastic optimizer, namely the SAM also, with suitable changes. The energy of an LJ cluster of n atoms is given by

$$E_n = 4\epsilon \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right]. \tag{1}$$

For locating critical points on the PES of such a cluster, we start by defining an objective function [12,17]

$$F = (E_n - E_{\mathrm{L}})^2 + \sum_{i=1}^{3n} \beta \left(\frac{\partial E_n}{\partial x_i} \right)^2 + \sum_{i=1}^{3n} \eta_i e^{p_i \lambda_i} \tag{2}$$

and a fitness function

$$f = e^{-F}. \tag{3}$$

The objective is to find the absolute minimum of F or, equivalently, the absolute maximum of f. In Eq. (2) E_{L} is an estimated lower bound to the energy which has to be supplied or deduced. This term is introduced to tune the search to locate a critical point around E_{L}. E_{L} can be updated at regular intervals, if needed. The second term is the gradient norm multiplied by β, the penalty weight factor. Since F has the dimension of energy squared, the dimension of β should be length squared. Its value is set equal to unity at the start of the search. Later on, as the gradient norm decreases to 0.1, β is allowed to increase 10-fold for every 10-fold decrease in the norm of the gradient vector. The third term in F is the curvature constraint term. The λ_is in this term are the eigenvalues of the Hessian matrix, p_is and η_is are penalty weight factors associated with the curvature constraint. If λ_i is positive, p_i is negative and η_i is positive. If λ_i is negative, then p_i is positive and η_i is negative. It may be mentioned that the six eigenvalues of the Hessian associated with the translational and rotational motion of the centre of mass, should be excluded from the sum in the curvature constraint term. The p_is are chosen to

avoid exponential underflow while the choice of η_i depends on the purpose of the search (seen later). Thus, for locating a minimum on the PES, the curvature constraint term is dropped (i.e. to say, we set $\eta_i = 0$ for $i = 1, 2, \ldots, 3n - 6$). The simplified fitness function ϕ_m is then given by

$$\phi_m = \mathrm{e}^{-\mathscr{F}_m}, \tag{4}$$

where

$$\mathscr{F}_m = (E_n - E_{\mathrm{L}})^2 + \sum_{i=1}^{3n} \beta \left(\frac{\partial E_n}{\partial x_i} \right)^2 \tag{5}$$

x_is in Eq. (5) are the cartesian coordinates of the space fixed r_i vector. E_{L} is set to a specified value and ϕ_m is maximized till a stationary point around E_{L} is reached. At this point, the gradient norm vanishes. It is clear from Eqs. (2) and (3) that ϕ_m will be maximally 1.0 at the stationary point if E_n is also equal to the current value of E_{L}. So, the achievement of a fitness of 1.0 will signify that a critical point has been hit, at $E_n = E_{\mathrm{L}}$ and that it is a minimum.

For locating an SP on the PES, (starting from a minimum, local or global) the lower bound term E_{L} is reset ($E'_{\mathrm{L}} = E_{\mathrm{L}} + \Gamma, \Gamma > 0$) in Eq. (2) and the curvature constraint term in the objective function (Eq. (2)) is brought back so that the new objective function ($\mathscr{F}$) reads

$$\mathscr{F} = (E_n - E'_{\mathrm{L}})^2 + \sum_{i=1}^{3n} \beta \left(\frac{\partial E_n}{\partial x_i} \right)^2 + \sum_{i=1}^{3n-6} \eta_i \mathrm{e}^{p_i \lambda_i}, \tag{6}$$

and the new fitness function ϕ is given by

$$\phi = \mathrm{e}^{-\mathscr{F}}. \tag{7}$$

We may mention here that the summation in the curvature constraint term in Eq. (6) runs over $3n - 6$ degrees of freedom implying that the translational and rotational motions of the centre of mass have been excluded. A straightforward application of the GA to the problem of maximizing ϕ of Eq. (7) is possible. Although we will not make use of this approach, it would be profitable to look into the salient features of the steps involved.

First, a new set of configurations of the atoms in the cluster with significant energy variations are generated for fresh genetic evolution with a view to maximizing the new fitness function ϕ. All η_is are set to have values around 0.01 or so, whereas p_is are set equal to 1.0 as long as all the Hessian eigenvalues are positive. If during the evolution, a configuration is generated in the population that corresponds to an arrangement of the interacting atoms for which one of the Hessian eigenvalues $(\lambda_i) < 0$, the corresponding η_i is reset to have a much higher absolute value (0.1 or so) but with a negative sign. Since the term $\eta_i \mathrm{e}^{-p_i \lambda_i}$ $(\eta_i \lambda_i < 0, | p_i \lambda_i | \sim 1)$, lowers the value of the objective function (F), there is an abrupt increase in the fitness of the specific string. After a few generations, the population begins to get dominated by similar strings as the selection process ensures that strings with a higher fitness have a higher multiplicity in the population. If we take care to exclude strings with more than one negative eigenvalues of the Hessian from the population, the evolution will eventually lead to a first-order saddle point on potential energy surfaces (PES), as close to E'_{L} as possible. E'_{L} itself is slowly raised by a predefined updating procedure [24].

A glance at the objective function (cf. Eq. (6)) would suggest that, for large systems, the technique mentioned in the preceeding paragraph might become costly. If at each step, the Hessian matrix has to be constructed and diagonalized for each string, the method might not be as appealing for larger systems. Hence, we propose to put forward a modified objective function where an approximate estimate of the first eigenvalue of the Hessian can be made from the gradients of the current and the immediately preceeding steps, thereby avoiding a costly step involving the calculation of eigenvalues of the Hessian. The modified method is as follows:

(i) We start by noting that for the searches leading to an SP of order 1, the full expression $\sum_{i=1}^{3n-6} \eta_i \mathrm{e}^{p_i \lambda_i}$ is not needed to enforce the curvature constraint for SP search since, the higher terms do not contribute much to increase the fitness value. So an evaluation of the first eigenvalue would be enough to guide the search. Of course, since only the first eigenvalue is needed, one can invoke any of the few roots technique to get the first eigenvalue after eliminating the six zero eigenvalues

associated with translation and rotation, which has to be done carefully. With this simplification, the curvature constraint term in the objective function would read $c' = \eta_1 e^{p_1 \lambda_1}$, where λ_1 is the smallest eigenvalue of the Hessian associated with non-translational-rotational modes of the LJ cluster.

(ii) We have already made our lives simpler. Further simplification can be achieved by noting that along a minimum energy path (MEP) to a first-order SP, the eigenvector corresponding to the negative eigenvalue of the Hessian should point nearly parallel to the gradient. The particular eigenvector can therefore be approximated by the gradient itself. Since only in the redesigned objective function the first term, $\eta_1 e^{p_1 \lambda_1}$, in the curvature constraint remains, only one of the Hessian eigenvalues (λ_1) is substituted by the simple difference of gradients

$$\lambda_{1\text{new}} = \mid \text{grad}_{\partial} \mid - \mid \text{grad} \mid, \tag{8}$$

where grad_{∂} is the local gradient at a second test point obtained by taking a small step from the given point towards the nearest saddle along the direction of the local gradient. Near an SP on an MEP, $\lambda_{1\text{new}}$ should be negative. The curvature constraint term at the designated point on the surface then becomes

$$c'' = \eta \, \text{sign}(\mid \text{grad}_{\partial} \mid - \mid \text{grad} \mid) \, e^{p_1 \{\mid \text{grad}_{\partial} \mid - \mid \text{grad} \mid\}}, \tag{9}$$

where η is the positive penalty factor which should lead the method to an SP. Of course, we cannot exclude the possibility of obtaining an SP of higher order. The modified algorithm could gain a significant computational advantage from two sources. First, it avoids costly diagonalization of the Hessian. Second, the negative eigenvalue of the Hessian is not easily lost once a string with a negative eigenvalue of Hessian has been identified, so that the search remains correctly constrained.

Having established our objective function and our fitness function, we now describe the stochastic optimizer, GA used by us for exploring the search space.

2.2. The optimization strategy

For finding the critical points of an n-atom LJ cluster we have to optimize $3n$ cartesian coordinates. This is achieved by the GAs in the following manner:

(i) GA creates a pool of probable trial solutions (the population of size n_p) to begin with. This is a fundamental difference which the GA has with other optimization techniques. Most techniques start with a single solution and iteratively improve it.

(ii) Suppose, we have created m solutions randomly. They are each a string of $3n$ cartesian coordinates of the atoms of the cluster. All of them are not equally fit. By fit, we mean the closeness of the trial solutions to the actual solution. A measure of this is provided by the value of the fitness function defined in Eq. (3). So, the first step, in the GA is a *selection* step, which ensures that the relatively better solutions are retained in the pool and the rest discarded. However, the size of the solution pool (n_p) is kept constant, and hence, the better solutions receive multiple copies in the post-selection population. This is in keeping with the Darwinian theory of survival of the fittest.

(iii) The selection operator increases the average fitness of the solution pool, but does not introduce new features or variations to it. If we are to achieve our goal of locating the true solution, extensive changes must take place in the trial solutions. Two operators are invoked to carry out this operation: The first is the *crossover* operator. Let S_1 and S_2 be a pair of trial solutions in the pool where

$$S_1 = x_1^1, x_1^2, x_1^3, \ldots, x_1^t, x_1^{t+1}, x_1^{t+2}, \ldots, x_1^{3n},$$
$$S_2 = x_2^1, x_2^2, x_2^3, \ldots, x_2^t, x_2^{t+1}, x_2^{t+2}, \ldots, x_2^{3n}.$$

We choose a crossover site randomly (say the tth site) with probability $p_c = 0.25$ or so. After choosing the crossover site, we exchange information between the two solutions S_1 and S_2 and generate a pair of new solutions S_1' and S_2' where

$$S_1' = x_1^1, x_1^2, x_1^3, \ldots, x_1^t, x_1^{t+1'}, x_1^{t+2'}, \ldots, x_1^{3n'},$$

$$S_2' = x_2^1, x_2^2, x_2^3, \ldots, x_2^t, x_2^{t+1'}, x_2^{t+2'}, \ldots, x_2^{3n'},$$

$$x_1^{r'} = f x_1^r + (1-f) x_2^r \quad \text{and}$$

$$x_2^{r'} = f x_2^r + (1-f) x_1^r \quad \text{for}$$

$$r = t+1, \ldots, 3n, \tag{10}$$

where $0 < f < 1$, and f is allowed to assume values in the specified range *randomly*.

Once the solution set is subjected to the crossover, the set is ready to undergo the next operation, called mutation, which introduces only subtle changes in the strings.

(iv) We define a mutation probability p_m (0.01–0.05). For each variable in the solution set, we generate a random number r between 0 and 1. If $p_m > r$, then that particular variable is selected for mutation. Suppose x_k^n (nth variable in the kth solution) has been selected for mutation. After mutation it becomes

$$x_k^{n'} = x_k^n + (-1)^l r \Delta, \tag{11}$$

where l is a random integer, r is a random number and Δ is a mutation amplitude that itself can be either drawn from a Gaussian distribution or selected *randomly* from a specified range.

The action of these three operators selection, crossover and mutation complete one generation in a GA. The transformed solution set is again subjected to the same sequence of GA operations and the process is iterated till a maximization of fitness is achieved to the extent possible.

3. Results and discussions

All the calculations have been carried out on Argon clusters using typical LJ parameters for argon [21], with $\sigma = 3.4$ Å and $\epsilon = 1.0$ kJ/mol. The systems investigated are (Ar_n) with $n = 7, 8, 15, 20, 30$. First the global minimum energy structures of these clusters were found out by starting from a randomly generated set of argon coordinates and then invoking GA for locating the minima. To be specific, the cartesian coordinates x_i, y_i, z_i of each argon atom were allowed to have values in the range of $0 \leqslant x, y, z \leqslant 14$ Å randomly,

in the starting population pool. The results agree well with those existing in the literature [13]. We omit details of the optimization procedure as the technique of locating the global minimum energy structure is well established [12].

Once the global minimum is located, we start looking for the first-order saddle point of (hopefully) the lowest height starting from the global minimum. A new set of strings are generated for creating the initial population for the SP search by perturbing the coordinates of the global minimum energy structure by ± 0.1Å, randomly. Fig. 1. displays a typical example, of the fitness distribution in the starting population pool used in the search for SP in the Ar_{20} cluster. Here we note that the fitness here refers to the fitness computed without the inclusion of the curvature constraint. The fitness values of the individual strings would get reduced drastically if curvature constraints are introduced (all λ_is > 0). A single point crossover is used with $p_c = 0.25$, while mutation probability p_m is taken to be 0.05. The intensity (Δ) of mutation is randomly chosen from an interval $l_1 \leqslant \Delta \leqslant l_2$, where $l_1 = 0.05$ and $l_2 = 0.20$. The E_L' in the objective function (Eq. (6)) is raised a little bit so that the objective function drives the structure out of the attractive basin. The fitness is then sought to be maximized. In the process, the gradient term is lowered as much as it can be around the new E_L'. If, a string with one negative eigenvalue of the
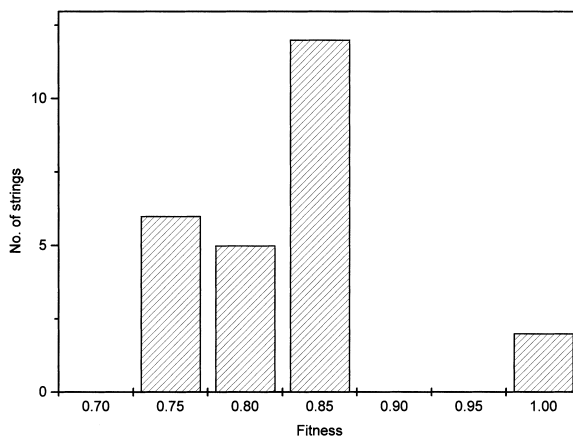

Fig. 1. Fitness distribution in the starting population for saddle point search in a Ar_{20} cluster.

Hessian is generated, the corresponding η is set to a large negative value (from $+0.01$ to -0.1) causing abrupt decrease in the objective function and a concomitant increase in the fitness function. The curvature constraint term in the objective function therefore forces the search to proceed along the direction of the eigenvector with negative eigenvalue. When it is seen that the gradient norm cannot be lowered further, E'_L is further increased, and the search steps are executed. The process is carried on till a SP of order 1 has been reached. So to say, this process creates an approximation to the minimum energy path.

The energies at the global minimum, SP and the lowest eigenvalue of the Hessian of a number of clusters at the saddle point are given in Table 1.

For Ar_{15}, Ar_{20} and Ar_{30}, the SP search was initiated from the globally minimum energy geometry in each case. E'_L was slowly updated and in each the search converged to the first-order SP reported in Table 1. Ar_7 and Ar_8 behave somewhat differently. In the case of Ar_7, the search converged to a higher energy saddle of order 1 when E'_L was updated at the standard rate (1% of current E'_L after every 100 generations). However, on reducing the rate of upward revision of E'_L, the search could be forced to converge to the lower saddle at $E = -15.444$. For Ar_8, the search from the global minimum converged to the SP at $E = -18.806$ although a lower saddle has been reported in the literature [23]. This happened irrespective of the updating rate used for E'_L. This lower saddle could be reached only if the search was initiated from a different local minimum, rather away from the global minimum.

Figs. 2(a)–(c), 3(a),(b), 4(a),(b), 5(a),(b) and 6(a),(b) depict the structures at the global minima
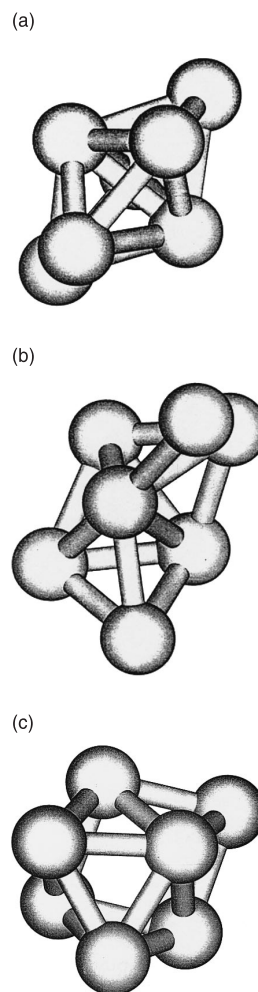

Fig. 2. (a) Structure at the global minimum of Ar_7 cluster. (b) Structure at the higher saddle point SP_2, of Ar_7 cluster. (c) Structure at the lower saddle point SP_1, of Ar_7 cluster.

Table 1
The global minimum energy (in ϵ), SP energy (in ϵ) and the only negative Hessian eigenvalue at the saddle point for Ar_n systems

n	Energy (global)	Energy (saddle)	First Hessian eigenvalue (saddle)
7	−16.505	−15.444 (−14.596)[a]	−0.8654
8	−19.821	−18.806 (−19.281)[b]	−0.8198
15	−52.322	−48.798	−0.8603
20	−77.177	−73.980	−0.8662
30	−128.286	−123.864	−9.0965

[a] A higher saddle point, also accessible from the global minimum.
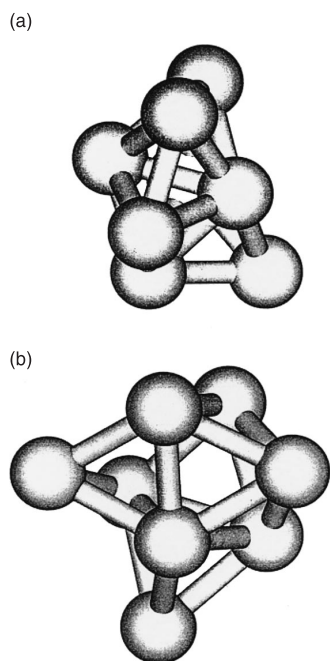[b] A lower energy first-order SP [23] on the PES which could not be reached from the global minimum.

(a)

(b)

Fig. 3. (a) Structure at the global minimum of Ar_8 cluster. (b) Structure at the saddle point of Ar_8 cluster.

and the SPs for Ar_7, Ar_8, Ar_{15}, Ar_{20} and Ar_{30} clusters, respectively. The convergence of the algorithm to the SP of the lowest height is generally fast, but not guaranteed, specially when the number of such points on the given PES is large and more than one of them are accessible from the stationary point from where the SP search is initiated. This difficulty is not intrinsic to the present method, but is shared by many. This aspect has been discussed by Quapp and Heidrich [22]. In our case, only a very slow rate of annealing (the rate at which E'_L is updated) can perhaps ensure that the MEP passes through the first-order SP of the lowest height. Thus, unless annealing is done very slowly, the MEP starting from the global minimum in Ar_7 ends up in the higher SP (SP_2, Fig. 2(b)) while slow annealing leads to the lower one (SP_1, Fig. 2(c)).

Fig. 7(a) depicts the evolution of energy of an Ar_{20} cluster with the number of generations using the gradient-only scheme. The search starts from the global minimum and the desired SP is reached in about 600 generations. The working population
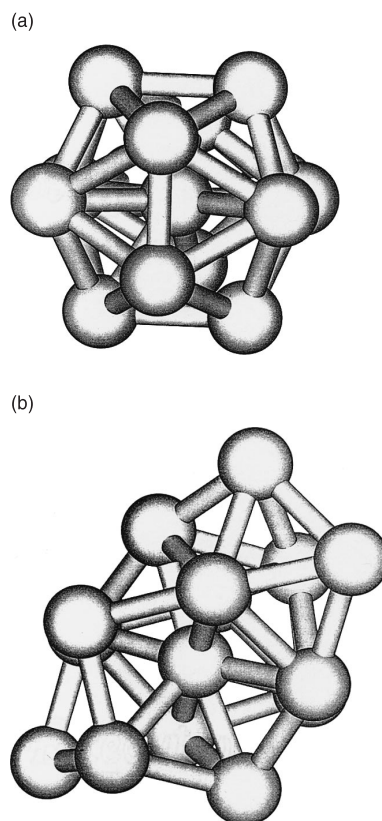
(a)

(b)

Fig. 4. (a) Structure at the global minimum of Ar_{15} cluster. (b) Structure at the saddle point of Ar_{15} cluster.

contained 25 trial solutions and the entire search was accomplished in 20 min of the CPU time on a 200 MHz Pentium. For comparing the performance of the present algorithm with that of the method proposed earlier [11], where explicit evaluation of all the Hessian eigenvalues was needed, we carried out a calculation for tracing the same SP with the previous method. The search took 800 generations to locate the saddle point (Fig. 7(b)). The time consumed was 50 min of the CPU on the 200 MHz Pentium. However, in the second case also (full Hessian case), the best string appeared in the population at around 600th generation. The evolution was allowed to continue for another 200 generations to ensure that there was no further scope for improvement. The economy achieved in the computation time can be accounted for by the fact that the present
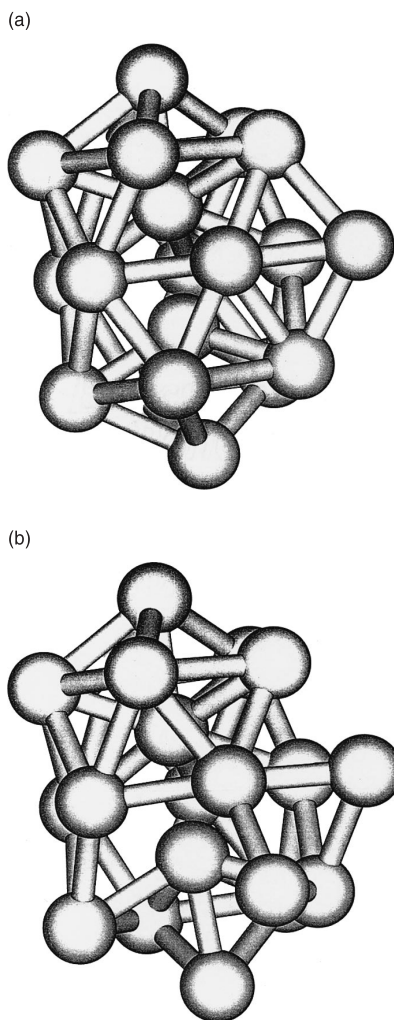
Fig. 5. (a) Structure at the global minimum of Ar_{20} cluster. (b) Structure at the saddle point of Ar_{20} cluster.

Fig. 6. (a) Structure at the global minimum of Ar_{30} cluster. (b) Structure at the saddle point of Ar_{30} cluster.

algorithm does not require the diagonalization of the full Hessian during the evolution. A comparison of the lowest eigenvalue of the Hessian during evolution in the present scheme with that in the unmodified GA based algorithm, shows that the negative eigenvalue is not lost more frequently in the previous method as speculated earlier. It may be noted that the fitness in the present method is calculated using only the negative eigenvalue of the Hessian in the curvature constraint term. That makes the fitness appear to improve faster.
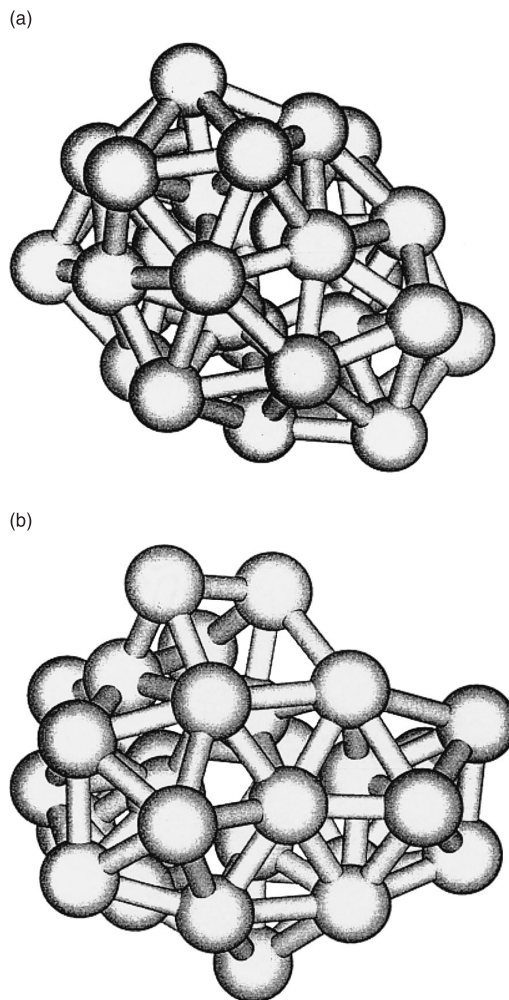
We have also located an SP of order 1 for the Ar_{30} system by using the gradient only scheme. Here also the population size was kept fixed at 25. The saddle point was located in about 4000 generations and the search took about 1 h CPU time on a 200 MHz Pentium Machine. The use of the gradient information only for constructing the objective function and hence the fitness function, clearly improves the efficacy of the GA based search for the saddle points on the complex PES. The same basic technique would be equally applicable to the SAM based algorithm proposed earlier
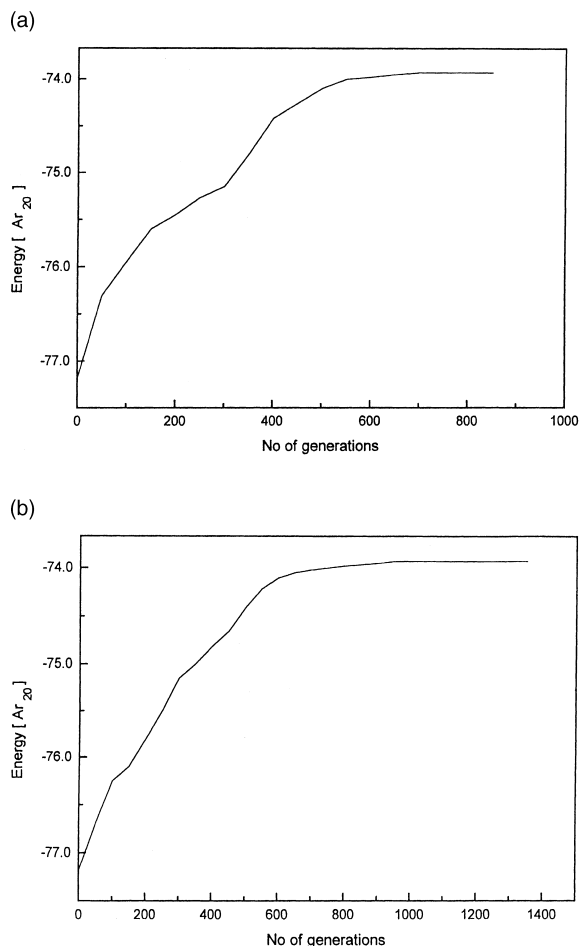
(a)

(b)

Fig. 7. (a) The evolution of energy with generations during saddle point search using the gradient dominated scheme for Ar_{20} clusters. (b) The evolution of energy with generations when the SP search is carried out with explicit evaluation of eigenvalues of the Hessian matrix in a Ar_{20} cluster.

[24], but it converges slowly when the cluster size is large. For clusters upto $n = 10$, the two strategies work more or less with equal facility.

4. Conclusions

We have demonstrated the workability of a new GA based scheme for finding saddle points of LJ clusters containing upto 30 atoms. The evaluation of the relevant Hessian eigenvalue from the gra-dient information coupled with the GA based stochastic search makes it a potent method for studying these multidimensional and extremely flat surfaces.

References

[1] M.R. Hoare, Adv. Chem. Phys. 40 (1979) 49.
[2] M.R. Hoare, M. McInnes, Faraday Discuss. Chem. Soc. 61 (1976) 12.
[3] M.R. Hoare, M. McInnes, Adv. Phys. 32 (1983) 791.
[4] C.J. Tsai, K.D. Jordan, J. Phys. Chem. 97 (1983) 11227.
[5] M.R. Hoare, P. Pal, Adv. Phys. 20 (1971) 161.
[6] M.R. Hoare, P. Pal, Adv. Phys. 24 (1975) 645.
[7] J.A. Northby, J. Chem. Phys. 87 (1987) 6166.
[8] D.M. Deaven, N. Tit, J.R. Morris, K.M. Ho, Chem. Phys. Lett. 256 (1996) 195.
[9] J.A. Niesse, H.R. Mayne, J. Chem. Phys. 105 (1996) 4700.
[10] S.K. Gregurick, M.H. Alexander, B. Hartke, J. Chem. Phys. 104 (1996) 2684.
[11] J. Mestres, G.E. Scuseria, J. Comput. Chem. 16 (1995) 729.
[12] P. Chaudhury, S.P. Bhattacharyya, Chem. Phys. 241 (1999) 313.
[13] D.J. Wales, J.P.K. Doye, J. Phys. Chem. A 101 (1997) 5111.
[14] D.J. Wales, J. Chem. Phys. 91 (1989) 7002.
[15] W. Quapp, Chem. Phys. Lett. 253 (1996) 228.
[16] F. Jensen, J. Chem. Phys. 102 (1995) 6706.
[17] P. Chaudhury, S.P. Bhattacharyya, Int. J. Quant. Chem. 76 (2000) 161.
[18] D.E. Goldberg, Genetic Algorithms in Search, Optimisation, and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[19] F. Hoffmeister, T. Bäck, Genetic Algorithms and Evolution Strategies, in: H.P. Schewefel, R. Männer (Eds.), Parallel Solving from Nature (PPSN I), Lect. Notes Comp. Sci., vol. 496, Springer, Berlin, 1991.
[20] K.A. De Jong, Machine Learning. 3 (1988) 121.
[21] W.J. Pullan, J. Comp. Chem. 18 (1997) 1096.
[22] W. Quapp, D. Heidrich, Theor. Chim. Acta. 66 (1984) 245.
[23] J.P.K. Doye, D.J. Wales, J. Chem. Phys. 102 (1995) 9673.
[24] P. Chaudhury, S.P. Bhattacharyya, Theochem. 429 (1998) 175.