Geometry optimizations of benzene clusters using a modified genetic algorithm

CAI, Wen-Sheng*,[a](蔡文生) YU, Fang[a](于芳) SHAO, Xue-Guang[b](邵学广)
PAN, Zhong-Xiao[a](潘忠孝)
[a] *Department of Applied Chemistry, University of Science and Technology of China, Hefei 230026, Anhui, China*
[b] *Department of Chemistry, University of Science and Technology of China, Hefei 230026, Anhui, China*

A modified genetic algorithm with real-number coding, non-uniform mutation and arithmetical crossover operators was described in this paper. A local minimization was used to improve the final solution obtained by the genetic algorithm. Using the exp-6-1 interatomic energy function, the modified genetic algorithm with local minimization (MGALM) was applied to the geometry optimization problem of small benzene clusters $(C_6H_6)_N$ ($N = 2$—7) to obtain the global minimum energy structures. MGALM is simple but the structures optimized are comparable to the published results obtained by parallel genetic algorithms.

Keywords Geometry optimization, genetic algorithms, benzene clusters

Introduction

Determination of the lowest energy configurations of molecular clusters is prohibitively difficult in computational chemistry due to the large number of local minima which even small molecular clusters possess. And the number of subsidiary minima increases exponentially with the number of molecules in clusters, making it a member of NP-hard problems. In order to solve the problem, Williams developed several methods such as the Newton-Raphson local optimization with assumption of a center of symmetry,[1] and the off-ridge eigenvector minimization with annealing (OREMWA) without the above assumption which is capable of starting from subsidiary minima and proceeding to global minima.[2] Van de Waal investigated structures of small benzene clusters by starting with an icosahedral 13-cluster and removing appropriate molecules from the icosahedron.[3]

As a powerful stochastic search strategy, genetic algorithms (GAs) have been tried in optimizing the structures of clusters. Williams proposed a binary-coded genetic algorithm (GAME) to obtain the global energy minimum of the dimer structures of benzene, naphthalene and anthracene.[4] Niesse carried out global optimization of atomic Ar clusters and molecular water clusters using the space-fixed modified genetic algorithm.[5] Gregurick and Alexander developed a modified deterministic/stochastic genetic algorithm (DS-GA) for optimization of $(Ar)_n$ and $B(Ar)_n$ clusters.[6] Recently, an efficient parallel genetic algorithm was proposed by Pullan to predict the structures of benzene clusters.[7] It can find the minimum energy structures for cluster of two to fifteen benzene molecules, but the restriction that the algorithm can be run only on parallel computers has greatly limited its application in structure optimization problems since parallel machines are not generally available to all chemists. In this study a modified genetic algorithm was constructed to solve the task on a conventional personal computer. The modification of the standard GA method involves non-uniform mutation operators,[8] arithmetical crossover operators[8] and local minimization of the best individual at the end of generation. By comparison of the optimized structures of benzene clusters obtained by MGALM with the published results obtained by the parallel GA and the other methods, it was shown that MGALM is a relatively simple but efficient algorithm that

* Received August 19, 1999; accepted January 4, 2000.
Project (No. 29775001) supported by the National Natural Science Foundation of China.

can be widely used in solving optimization problems.

Theory and algorithm

The most stable structure of a molecular cluster is often the geometry with the lowest potential energy. Because the global minimum structure lies in a large and complex energy hypersurface, many stochastic searching techniques have been tried to search for the most stable structure, such as random search, Monte-Carlo,[9] simulated annealing,[10] evolutionary programming,[10] and genetic algorithm.[10,11] Among them, the genetic algorithm is much simpler to implement. Initially, a random population of individuals is created, and then it is evaluated by a fitness function. Only the fitter individuals are allowed to survive and are selected as the parents for the next generation. The fitter individuals passed their genetic information onto the next generation through crossover and mutation operations. This process is repeated until the best individual (solution) is found.

The fitness function is the key factor in a GA. It determines the performance of a GA. In the geometry optimization problem of benzene clusters, we choose the potential energy of each geometry as a measure of fitness. The individual is evaluated and selected according to this criterion. The evolution of the genetic algorithm is thus straight forwardly directed to lower the potential energy till the global minimum is located.

With the assumption that the benzene molecule be a rigid planar structure, only the intermolecular nonbond interactions including van der Waals interactions and electrostatic interactions are considered. The exp-6-1 interatomic energy function for calculating the potential energy[4] is adopted:

$$\nu(r_{ij}) = B\exp(-Cr_{ij}) - Ar_{ij}^{-6} + q_iq_jr_{ij}^{-1} \quad (1)$$

$$V = \sum_{m=1}^{N-1}\sum_{n=m+1}^{N}\nu(r_{ij}) \quad (2)$$

where, V is the sum of all nonbonded pair potentials between N benzene molecules, $\nu(r_{ij})$ is the potential energy of the interacting atom pair (atom i of molecule m and atom j of molecule n at distance r_{ij}). Eq. (1) describes nonbonded interactions as repulsion interaction, dispersion interaction and Coulombic interaction, where B and C are the exchange repulsion coefficients, A is the dispersion energy coefficient and q_i, q_j are the net atomic charges for atom i and j. The numeric value of these coefficients are associated with the type of atoms involved in the interaction. In this study, for H – H, H – C, C – C interactions, A is 1.36×10^{-4}, 5.73×10^{-4} and 2.414×10^{-3} kJ·mol^{-1}·nm^6, respectively, B is 11677, 65485 and 367250 kJ·mol^{-1}, respectively, C is 0.374, 0.367 and 0.360 nm^{-1} respectively. The net atomic charges for H and C are 0.153e and −0.153e, respectively. These values for parameters A, B, C and q were obtained by Williams[12] and used in the previous studies.[1-4,7] The distance of C—C and C—H in the benzene molecule is 0.1397 and 0.1027 nm, respectively.

Each geometry of an N molecule benzene cluster is represented by a chromosome with the following structure,

```
typedef struct{
        float x;
        float y;
        float z;
        float θ;
        float φ;
        float ψ;
        }GENE;
typedef struct{
        GENE chrome[nChrome]; //nChrome
        = N - 1
        float fitness; // fitness of individual
        }CHROME;
```

Here, each gene represents a benzene molecule, the first three parameters of each gene x, y, z are the translation parameters which specify the center of the molecule. The other three parameters θ, ϕ, ψ are Euler angles which specify the molecular orientation. Since the first molecule in a cluster is fixed in the space to define the coordinate system (the same atomic coordinates as in Ref. 6 are used in our study), therefore for an N molecule benzene cluster, only $N-1$ genes, *i.e.*, $6N-6$ parameters are used to fully specify the cluster. All these parameters are encoded in a real-value interval [0, 1], when decoding, the three translation parameters are decoded back to interval [−8.0, 8.0], while the Euler angle θ is decoded back to interval $[0,\pi]$, and the other two angles ϕ and ψ are both decoded back to interval $[0,2\pi]$. This real-value encoding is a plausible encoding strategy that is superior to binary encoding in opti-

mization problems proven by Deaven's work.[13]

Besides the real-value encoding strategy, our MGALM has other modifications of standard GA method to make it more efficient. The first modification is non-uniform mutation. It means that the search space can be automatically adjusted as the optimization process proceeds. For the non-uniform mutation, suppose a chromosome $x_i^t = (v_1, \cdots, v_k, \cdots, v_n)$ will mutate to $(v_1, \cdots v_k' \cdots v_n)$, the value of v_k' is determined in the following random way:

$$v_k' = \begin{cases} v_k + \Delta(t, \mathrm{UB} - v_k), & \text{random} = 0 \\ v_k - \Delta(t, v_k - \mathrm{LB}), & \text{random} = 1 \end{cases} \quad (3)$$

where LB and UB are the lower and upper limit of v_k, respectively. Function $\Delta(t, y)$ is selected as follows:

$$\Delta(t, y) = y \cdot [1 - r^{(1-t/T)^b}] \quad (4)$$

where r is a random number from interval $[0,1]$, T is the total generation number, b is a descending parameter that determines the degree of mutation nonuniformity (in this study b is 0.8). It will return a value in the range of $[0, y]$ and this value is approaching zero along with the increase of generation number t. Therefore, during the initial stage, big creep mutation giving a big disturbance around a real number is used to enlarge the search space. Along with the increase of the generation number t, mutation creep declined gradually, and, at the end of the evolution, little creep mutation giving a little disturbance around a real number is used to locally optimize the chromosome. In MGALM, both non-uniform mutation and uniform mutation in a ratio of 4:1 are used.

The second modification is arithmetical crossover. Take two (real-valued) parent' genes p and q, calculate their offspring genes p' and q' as a linear combination of the parent' genes by

$$\begin{aligned} p' &= k \cdot p + (1 - k) \cdot q \\ q' &= (1 - k) \cdot p + k \cdot q \end{aligned} \quad (5)$$

with parameter $k \in [0, 1]$. For each individual gene participating in the crossover, the parameter k is a uniformly random choice from the interval $[0, 1]$. The arithmetical crossover is more appropriate in real-number encoding genetic algorithm than other crossover methods, which has been proven in our previous works.[14,15]

The tournament selection mechanism based on fitness is used. In addition, an elitist selection is adopted to ensure the best chromosome for each population is placed into the next generation unchanged, therefore the best fitness score from one population to the next will never decrease.

However, we cannot ensure that the GA procedure has reached the global minimum energy. Additional modification to the standard GA involves a local minimization of the best individual of the population at the end of generation to improve the final result. It randomly searches a superior solution within the neighboring area (small ranges of each parameters) of the current solution and then replaces it. This local search procedure will be repeated until certain stop criterion is satisfied. To ascertain the speed of the MGALM, the local minimization procedure is not for each member of the population at every generation[6] but only for the best solution (the best individual) in the last generation. It greatly accelerates convergence to the global minimum.

MGALM optimization procedures and results

The program MGALM is written in C++ language, compiled with a BC compiler, and implemented on a Pentium 233/64M.

Optimization of mathematical functions using MGA

In order to test the performance of the modified genetic algorithm (MGA), both MGA and the standard GA (SGA) procedures are applied to find the global minimum of several mathematical functions with multiminima described in Table 1. The following values of parameters are used: the population size N_p is 11, the crossover probability P_c is 0.9, and the mutation probability P_m is 0.05. Both the results over 50 runs obtained by MGA and SGA are given in Table 2. From the average and minimum number of function calculations and the quality of the solutions (the stop criteria) in Table 2, it is clear that MGA method is superior to SGA.

Table 1 Function description

Function	Equation	Parameter interval	Global minimum
F1	$f_1(x) = nA + \sum_{i=1}^{n}[x_i^2 - A\cos(2\pi x_i)]$ $A = 8$	$n = 5$ $-5.12 \le x_i \le 5.12$	$x_i = 0$ $f_{min} = 0$
F2	$f_2(x) = -\sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	$n = 5$ $-500 \le x_i \le 500$	$x_i = 420.9687$ $f_{min} = -2094.9144$
F3	$f_3(x) = k_3\{\sin^2(\pi k_4 x_1) + \sum_{i=1}^{n-1}(x_i - k_5)^2[1 + k_6\sin^2(\pi k_4 x_{i+1})] + (x_n - k_5)^2[1 + k_6\sin^2(\pi k_7 x_n)]\}$ $k_3 = 0.1,\ k_4 = 3,\ k_5 = 1,\ k_6 = 1,\ k_7 = 2$	$n = 5$ $-5 \le x_i \le 5$	$x_i = 1$ $f_{min} = 0$
F4	$f_4(x_1, x_2) = \{\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\}$ $\{\sum_{i=1}^{5} i\cos[(i+1)x_2 + i]\}$	$-10 \le x_1, x_2 \le 10$	18 global minima $f_{min} = -186.7309$

Table 2 Comparison of MGA with SGA (The stop criteria of SGA and MGA are $|f_{cal} - f_{min}| < 0.001$, and $|f_{cal} - f_{min}| < 0.00001$, respectively. Results over 50 runs were sampled)

Function	Number of function calculations					
	SGA (1X)[a]		SGA (UX)[b]		MGA[c]	
	Average	Minimum	Average	Minimum	Average	Minimum
F1	1.1×10^5	11251	7.1×10^4	41391	9.0×10^3	5201
F2	3.7×10^5	195551	2.6×10^5	52481	1.3×10^4	5271
F3	1.3×10^4	2721	1.3×10^4	2461	3.7×10^3	1461
F4	1.4×10^5	29411	1.7×10^5	14961	7.2×10^3	881

[a] Standard Genetic Algorithm using single point crossover. [b] Standard Genetic Algorithm using uniform point crossover. [c] Modified Genetic Algorithm using arithmetic crossover and non-uniform mutation.

The local minimization for the best solutions obtained by MGA will improve their quality. For F1 with 10 dimensions, twenty runs of MGALM in 5000 generations were tested. The average minima obtained by MGA and MGALM are 2.9×10^{-5} and 1.1×10^{-5}, respectively.

Optimization of benzene clusters using MGALM

Investigations were performed to check the effect of changing the global parameters for this algorithm, the results show that the best results can be obtained with small population size ($N_p = 15$), a low mutation probability ($P_m = 0.05$) and a high crossover probability ($P_c = 0.9$).

The local minimization procedure randomly generates a new solution (x') near the current solution (x), *i.e.*, $x' \in [x - q,\ x + q]$. If the new solution is better than the current one, it will replace the current one and become a new current solution. In this study, q is 0.0001. The stop criterion of the local minimization is no improvement for the best solution in 2000 steps.

MGALM was used to study the geometry optimization problems of small benzene clusters $(C_6H_6)_N$ (N = 2—7). The optimized structures were obtained after 5000 generations and the local minimization procedure. The running time for each run varied with the molecule number N of the corresponding benzene clusters. For a benzene dimer it only needs less than one minute to locate the global minimum, but for a 7 molecule benzene cluster, it will take about 1 hour to find the global minimum.

The global energy values of benzene clusters $(C_6H_6)_N$ for N = 2—7 obtained by MGALM are listed

in Table 3, and the published results obtained by the parallel genetic algorithm and the other methods are also listed in it. From the comparison of these two sets of values, it can be seen clearly that MGALM is in no way inferior to parallel genetic algorithms in geometry optimization problems for small benzene clusters ($N \leqslant 6$), only a very small error exists for the 7 benzene cluster.

Table 3 Comparison of MGALM calculated energy with published results

N	Energy (ref.)	Energy (cal.)	N	Energy (ref.)	Energy (cal.)
2	-11.0	-11.0	5	-79.1	-79.1
3	-32.1	-32.1	6	-106.4	-106.4
4	-55.6	-55.6	7	-134.1	-134.0

The energy values are measured in unit kJ/mol. The reference energies are from the Refs. 2, 3 and 7.

Fig. 1 to Fig. 6 display all the energy-optimized structures of the benzene clusters $(C_6H_6)_N$ for $N = 2$—7 found by MGALM.

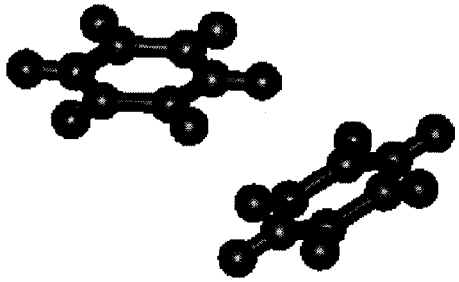

Fig. 1 Optimized structure of benzene dimer.

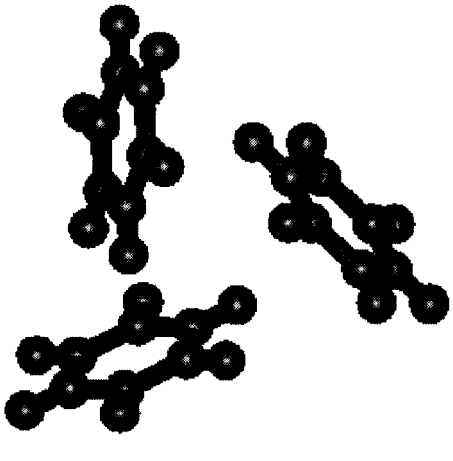

Fig. 2 Optimized structure of benzene trimer.

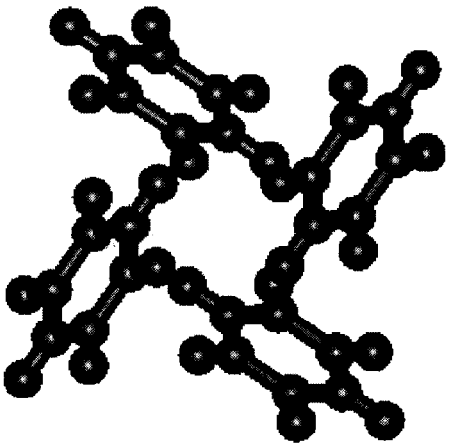

Fig. 3 Optimized structure of benzene tetramer.

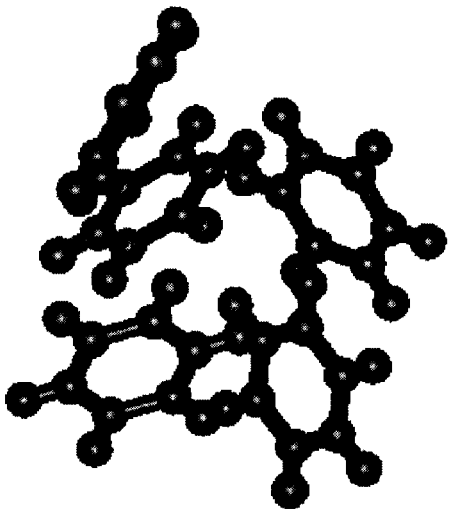

Fig. 4 Optimized structure of benzene pentamer.

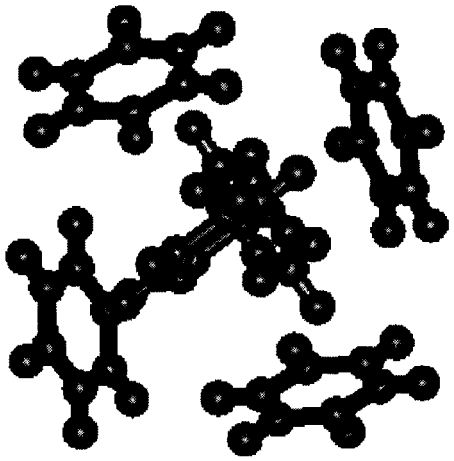

Fig. 5 Optimized structure of benzene hexamer.

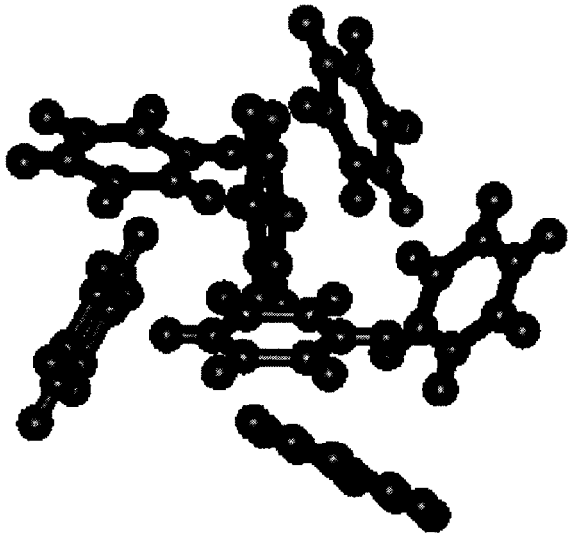

Fig. 6 Optimized structure of benzene septamer.

The two molecular faces in benzene dimer (Fig. 1) have a dihedral angle of 26.3° and an intermolecular center to center distance of 0.469 nm. In benzene trimer (Fig. 2) all the dihedral angles are 61.1° and all the intermolecular center to center distances are 0.498 nm. For adjacent pair of benzene molecules in tetramer (Fig. 3), they have dihedral angles of 77.6°, and for the diametrically opposing pairs they have dihedral angles of 55.2°, the respective intermolecular distances are 0.504 and 0.576 nm. In benzene pentamer (Fig. 4) the dihedral angles range from 32 to 95° and the intermolecular distances range from 0.50 to 0.57 nm. All the other dihedral angles and intermolecular distances in benzene hexamer (Fig. 5), septamer (Fig. 6) also agree with the published results.[2,3,7]

From Fig. 1, the configuration of the dimer is not a parallel arrangement, nor a T-arrangement. Previous studies showed that the calculated structure of dimer might be very sensitive to the applied point charge distribution.[3] It was found that point charges smaller than 0.13e resulted in essentially parallel displaced configurations, while larger charges ($\approx$0.17e) resulted in T-shaped configurations.[3]

Conclusion

In this paper, a modified genetic algorithm with a local minimization procedure (MGALM) was designed to perform structural optimizations on small benzene clusters ($(C_6H_6)_N$ N = 2—7). The results of test functions using MGA showed that it was more efficient than SGA. With the tailored genetic operators, our MGALM successfully located all the currently accepted global minima for these clusters and the results of MGALM presented above can be viewed as analogous to those of the parallel genetic algorithm. The results showed that this algorithm is efficient and practical, it is hopeful to become a reliable means of performing geometry optimizations in a wide variety of chemical systems.

References

1. Williams, D.E., *Acta Crystallogr.*, **36**, 715(1980).
2. Williams, D.E., *Chem. Phys. Lett.*, **192**, 538(1992).
3. Van de Waal, B.W., *Chem. Phys. Lett.*, **123**, 69 (1986).
4. Xiao, Y.L.; Williams, D.E., *Chem. Phys. Lett.*, **215**, 17(1993).
5. Niesse, J.A.; Mayne, H.R., *J. Comput. Chem.*, **18**, 1233(1996).
6. Gregurick, S.K.; Alexander, M.H., *J. Chem. Phys.*, **104**, 2684(1996).
7. Pullan, W.J., *J. Chem. Inf. Comput. Sci.*, **37**, 1189 (1997).
8. a) Liu, Y.; Kang L.S.; Chen Y.P., *Non-Arithmetic Parallel Algorithms-Genetic Algorithm*, Science Press, Beijing, 1997.
   b) Pan, Z.J.; Kang, L.S.; Chen, Y.P., *Evolutionary Computation*, Tsinghua University Press, Beijing, 1998.
9. Saunders, M., *J. Am. Chem. Soc.*, **109**, 3151(1987).
10. Tuffery, P.; Etchebest, C.; Hazout, S.; Lavery, R., *J.*

Comput. Chem., **14**, 790(1993).

11. Judson, R.S.; Jaeger, E.P.; Treasurywala, A.M.; Peterson, M.L., *J. Comput. Chem.*, **14**, 1407(1993).
12. Williams, D.E.; Starr T.L., *Comput. Chem.*, **1**, 173 (1977).
13. Deaven, D.M.; Ho, K.M., *Phys. Rev. Lett.*, **75**, 288 (1995).
14. Shao, X.-G.; Yu, F.; Kou, H.-B.; Cai, W.-S.; Pan, Z.-X., *Anal. Lett.*, **32**, 1899(1999).
15. Cai, W.-S.; Yu, F.; Shao, X.-G.; Pan, Z.-X., *Anal. Lett.*, **33**, 373(2000).

(E9908103 SONG, J.P.; LING, J.)