Computers Chem. Vol. 14, No. 4, pp. 305–311, 1990
Printed in Great Britain.

0097-8485/90 $3.00 + 0.00

A COMBINED SIMULATED ANNEALING AND QUASI-NEWTON-LIKE CONJUGATE-GRADIENT METHOD FOR DETERMINING THE STRUCTURE OF MIXED ARGON–XENON CLUSTERS

I. M. NAVON[1,3] F. B. BROWN[2,3] and DANIEL H. ROBERTSON[2,3]
Departments of [1]Mathematics and [2]Chemistry and [3]Supercomputer Computations Research Institute, Florida State University, Tallahassee, FL 32306-4052, U.S.A.

(*Received in revised form* 8 *January* 1990)

Abstract—This paper shows how various limited-memory quasi-Newton large-scale unconstrained minimization methods can be used to speed up the location of global minima of potential energy surfaces related to the structures of mixed Ar–Xe clusters. Both a simulated annealing method and a finite-temperature lattice-based Monte Carlo method are accelerated by the various quasi-Newton limited-memory methods which are then compared for computational efficiency.

1. INTRODUCTION

Atomic and molecular clusters play important roles in a number of phenomena including astrophysical processes, atmospheric reactions, nucleation and catalysis. Clusters are also intrinsically important in any fundamental theory of matter as they form a natural bridge between isolated atoms and molecules on the one hand, and liquids and solids on the other.

It has been known for a long time that the structures of small clusters in the gas phase generally differ from the packing structure of the substance in its crystalline form (Germer, 1939). During the past 50 years a number of schemes and methods have been employed to try to determine and understand the equilibrium structures of clusters. [A brief summary of this previous work is given in Robertson *et al.* (1989)].

From a molecular viewpoint, the potential energy of the ground state of the cluster can be described by a hypersurface or potential energy surface (PES) of large dimensionality ($3N - 6$, where N is the number of atoms in the cluster). The structure of the cluster's most stable conformation is the geometry corresponding to the global minimum of the cluster's PES. Determining the global minimum of a particular PES is complicated by the fact that these PESs contain many local minima, and, in fact, the number of local minima grows exponentially with N (Hoare & McInnes, 1983). Therefore, to determine the energy-optimized structures of these clusters, one must employ a method that will locate the global minimum on a PES and not become trapped in one of its many local minima.

One of the few optimization methods that is capable of locating the global minimum of a hypersurface containing multiple local minima is the simulated annealing (SA) method (Kirkpatrick *et al.*, 1983). Wille (1987) has recently used this method in conjunction with a local minimization algorithm of the PARTAN (Shah *et al.*, 1964) type, to determine the energy-optimized geometries of rare gas clusters (RGCs) containing up to 25 atoms. In this study, the PESs were constructed from pairwise-additive, Lennard–Jones (L–J) potentials. Using these methods, Wille discovered new, energy-optimized conformations for RGCs containing 23 and 25 atoms as compared to those previously reported by Hoare & Pal (1971).

Northby (1987) has recently studied the structures of L–J RGCs containing from 13 to 147 atoms using a lattice-based searching method followed by a local minimization algorithm of the PARTAN type. In a similar vein, the present authors (Robertson *et al.*, 1989) have developed the finite-temperature lattice-based Monte Carlo (FTLBMC) method for determining optimized structures of mixed RGCs containing both Xe and Ar atoms. The FTLBMC method is designed for investigating clusters that have an underlying structure (namely, that of the corresponding cluster containing the same number of atoms—but only one type of atom) which is relaxed or perturbed in the mixed RGC because of the presence of the two different types of atoms. In the present paper, we discuss the algorithmic details of the FTLBMC method and the particular local optimization methods that we employed in these studies and compare the efficiency of this method with the SA method.

The remainder of the paper will be organized as follows. Section 2 will define the PES used to model the atomic interactions. Section 3 will present the SA algorithm coupled with a local minimizer and Section 4 will discuss the numerical results obtained

from this algorithm. The FTLBMC algorithm will be presented in Section 5 and a comparison of the four local minimizers used in the FTLBMC method will be discussed in Section 6. A brief summary of the limited-memory quasi-Newton methods used will be given in Section 7 and finally, the overall conclusions will be presented in Section 8.

2. THE POTENTIAL ENERGY SURFACES (PESs)

The PESs used to model the RGCs in this study consisted of pairwise-additive L–J two-body potentials. The interaction between the two atoms i and j which are separated by a distance, r_{ij} is given by

$$v(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right], \tag{1}$$

where v_{ij} is the potential in units of the well depth, and ϵ and σ are parameters for the pair of interactive atoms which were used in our previous work (Robertson *et al.*, 1989). The PES for a cluster containing N atoms is given by

$$V(r^N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} v_{ij}(|r_i - r_j|). \tag{2}$$

As indicated in the introduction, these types of PESs contain very large numbers of local minima even for quite small clusters. For example, for a cluster of size $N = 13$, there are 998 minima known (Hoare & McInnes, 1983).

3. THE SIMULATED ANNEALING (SA) ALGORITHM

One of the few optimization algorithms capable of determining a global minimum is the simulated annealing algorithm. The method was first applied for discrete combinatorial optimization problems (Kirkpatrick *et al.*, 1983). Vanderbilt & Louie (1984) extended the method to problems with continuous variables. Consider a function $E(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ defined over the n-dimensional parameter space. We wish to minimize E w.r.t $\mathbf{x}$, where E is the energy of the physical system for example.

The Metropolis *et al.* (1953) Monte-Carlo algoritm proceeds by choosing an initial starting point $\mathbf{x}_0$ and making random steps $\Delta\mathbf{x}$. At each step the change

$$\Delta E = E(\mathbf{x} + \Delta\mathbf{x}) - E(\mathbf{x}) \tag{3}$$

in the objective function is evaluated. If ΔE is negative, the step is accepted. If ΔE is positive, the step is accepted with a probability

$$p = \exp(-\Delta E/T). \tag{4}$$

The series of accepted steps generates a random walk which explores the parameter space. The parameter T plays the role of temperature: i.e. as T is decreased slowly the volume $\Omega(T)$ of the phase space with nonnegligible $P(\mathbf{x})$ [where $P(\mathbf{x})d^n\mathbf{x}$ is the probability that the random walk will be in the volume $d^n\mathbf{x}$ on any given step at long times] shrinks until the system is eventually forced to anneal into the configuration of lowest E.

If the annealing is carried out slowly enough, the system will avoid being trapped into local minima as the method accepts steps which increase E temporarily to get over a barrier into a new local minimum, i.e. the algorithm is not confined to a single catchment region. At each temperature T random steps are generated using a random number generator.

The size of the steps should be such that approximately half of the steps are accepted. [See Vanderbilt & Louie (1984) for typical techniques.] The number M of random steps in the random walk at a given temperature should be chosen large enough so as to generate reasonable statistics so that representative regions of the parameter space are sampled. A factor χ_T by which the temperature is reduced after each random walk was chosen to be 0.9. As soon as the energy dropped below a certain value, determined from a preliminary simulation, we switched to an efficient memoryless quasi-Newton-like conjugate-gradient algorithm for local minimization. This procedure was carried out in the hope to gain computational efficiency by saving many function evaluations when the current point is close to the final (global) minimum and the temperature is too low to allow an escape from its catchment region. As an additional research guide, we applied the conjugate-gradient local minimizer at the end of each random walk, to gain information about the depth of the local minimum [See the Appendix for a flowchart of SA with the local minimizer algorithm.]

In our numerical experiments we will aim to quantify the computational savings and efficiency in terms of CPU time between this coupled procedure vs the classical SA method applied to this global minimization problem.

4. NUMERICAL RESULTS

In order to test the impact of the computational gains of the implementation of the SA in conjunction with the conjugate-gradient local minimizer we tested a case of a mixed Ar–Xe cluster with 19 atoms consisting of 10 Xe and 9 Ar atoms.

The SA performed 6000 steps per walk with an initial temperature of 0.6, scaled in intervals of energy units of 120 K, and a temperature lowering factor of 0.9 down to a final temperature of 0.2. The SA algorithm (see Kirkpatrick *et al.*, 1983) on its own required 3267 s of CPU time on a VAX-8700 to reach the global minimum, whereas the SA algorithm in conjunction with the local minimizer took only 682 s of CPU time to locate the same global minimum, i.e. a ratio of 4.79 speed-up. This ratio was typical for many other experiments with various numbers of atoms. Only about $<1\%$ of the total computer time for cluster optimization was spent in

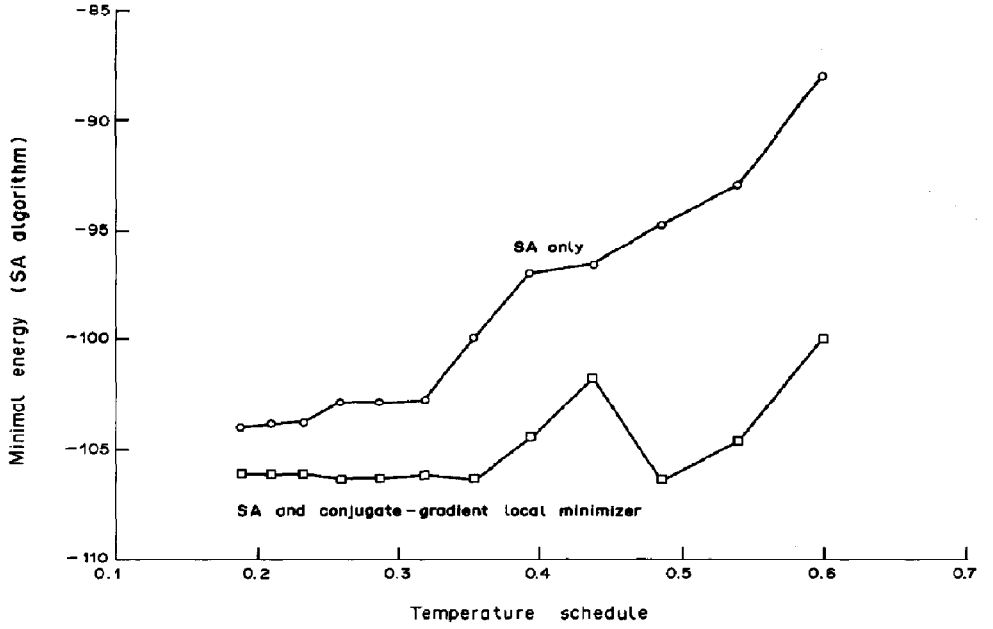


Fig. 1. Application of the SA method to determine global minima of the PES with and without the conjugate-gradient local minimizer when the SA algorithm reaches low energies (low temperatures).

the limited memory quasi-Newton-like conjugate-gradient program (Shanno & Phua, 1980). (See also Fig. 1 for the impact of the conjugate-gradient local minimizer on the critical SA low-temperature slow-down.)

5. FINITE-TEMPERATURE LATTICE-BASED MONTE-CARLO METHOD (FTLBMC)

The SA method proves to be computationally expensive in terms of the number of evaluations of the potential energy. This is due, in part, to the fact that long walks must be performed at each temperature to ensure that representative regions of the configurational space have been sampled. A more efficient approach was suggested by the works of Li & Scheraga (1987) and Saunders (1987). They proposed a Monte-Carlo sampling strategy which samples only the minima of the PES rather than walking over all of the PES. They sampled from minimum to minimum by random displacements in the coordinates of the structure, followed by subsequent optimization to the nearest minimum from this distorted structure. The acceptance of the new minimum energy structure was based on the Metropolis *et al.* (1953) algorithm. The random step sizes were adjusted so that the acceptance ratio was approx. 0.5 to allow for maximum information to be obtained as in normal Monte-Carlo calculations.

The above ideas were used in the formulation of the FTLBMC method for the search of the global energy minimum of the mixed Ar–Xe RGC (Robertson *et al.*, 1989). The underlying premise of the FTLBMC method is that the global minimum for the mixed or heterogeneous cluster is a distortion from the global minimum of the pure or homogeneous cluster containing only one type of atom. (See Fig. 2 for two representative mixed Ar–Xe clusters.) Therefore, the FTLBMC method uses the structure of the N atom pure RGC, Ar_N, as the underlying lattice for the mixed cluster global energy minimum search. The number of Xe atoms present in the mixed N atom RGC are randomly mapped onto the positions of Ar atoms in the pure structure (or lattice) and then, since this structure is not in equilibrium due to the different optimal lengths of the Ar–Xe and Xe–Xe bonds, a local optimizer, is called to minimize the energy of this structure. This now gives an energy optimized structure, S_1, with energy E_1. Then using the unoptimized lattice, a random pair of Ar and Xe are interchanged and the structure is optimized to generate a new structure, S_2, with energy E_2. The acceptance of the new structure is based on the Metropolis criteria and the procedure continues until a set number of new structures have been sampled. (See Fig. 3 for a plot of the minima sampled as a function of the step number.) The upper bound for the global energy minimum is then taken as the lowest energy structure found during the calculation. As in the above work, the main difference between the FTLBMC and SA methods is that the FTLBMC method samples and compares only the minima on the PES while the SA method samples any accessible point on the PES. (See the Appendix for a flowchart of the FTLBMC algorithm.)

The use of random interchanges as opposed to a systematic search of the lattice is due to the factorial nature of the number of possible configurations as the

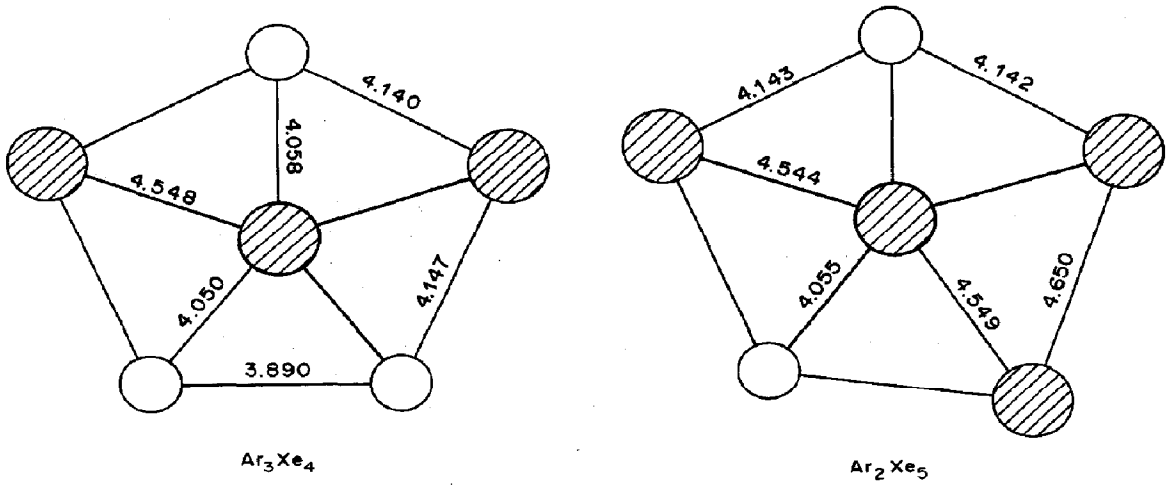


Fig. 2. The energy optimized structures of the Ar_3Xe_4 and Ar_2Xe_5 mixed RGCs. The hatched and open circles represent Xe and Ar atoms, respectively. The perspective is directly along the axis containing the apex atoms; there is an additional Xe atom hidden directly below the central Xe atom.

size of the cluster is increased. The temperature in the Metropolis acceptance criteria was used to modify the acceptance ratio as in other Monte-Carlo calculations. For the most efficient sampling of the lattice, the temperature must not be so high that there are too many accepted steps or too low so that the region of configuration space sampled is limited. The possibility of being trapped in a local minimum for a long time is also present at temperatures that are too low. Thus for our calculations, a temperature of 0.5 (60 K) was found to give an acceptance ratio of approx. 0.5. As in the SA algorithm, one should run the FTLBMC method several times using different sets of random numbers to increase the probability that the lowest minimum energy is the desired global minimum.

Numerically, the FTLBMC method peforms faster than either the SA algorithm alone or the SA algorithm combined with a local minimizer. This is

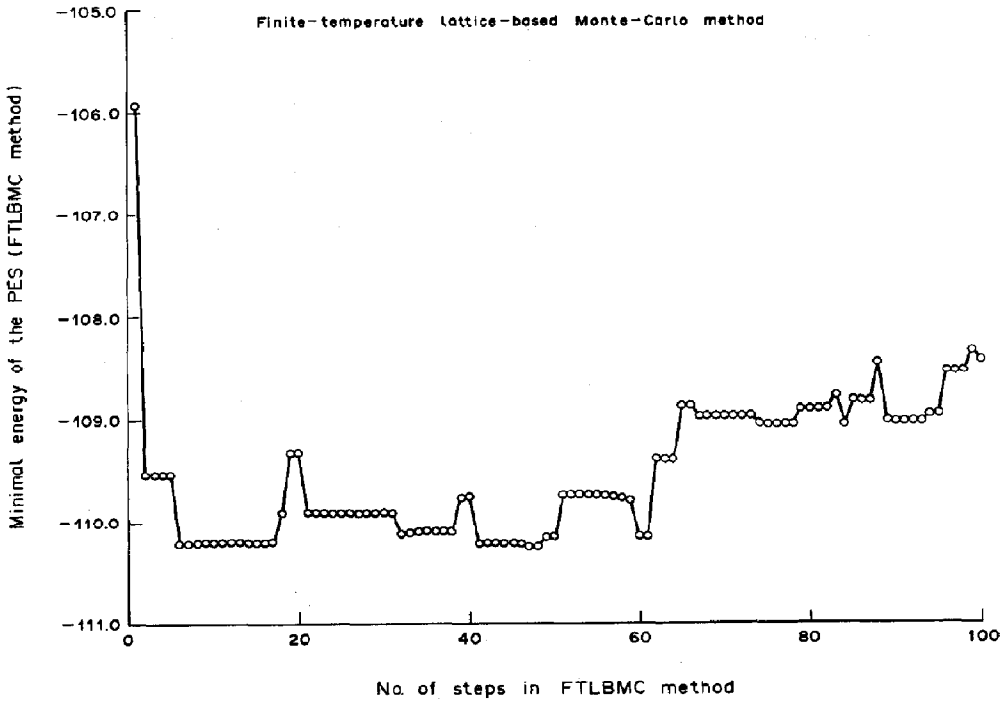


Fig. 3. The FTLBMC method using CONMIN as the local minimizer, displaying the minima of the PES as a function of the number of steps in the FTLBMC method.

due to the fact that for moving specific pairs of atoms in the mixed Ar–Xe clusters it takes many concerted moves of specific atoms to produce the desired interchange in positions of a pair of Ar–Xe atoms. For the typical case, the FTLBMC method outperforms the best SA method by a factor of 25—i.e. taking about 17 CPU s on a CYBER-205 as opposed to 420 s of CPU time for the best scenario SA algorithm accelerated by a local minimizer. As opposed to the SA algorithm with a minimizer, where only about 1% of the time is spent in local minimization, the bulk of the CPU time in the FTLBMC method is consumed in the local minimization, and the FTLBMC method, therefore, provides a vehicle to find the best local minimizer for this and similar problems.

6. COMPARISON OF DIFFERENT LOCAL MINIMIZERS IN THE FTLBMC ALGORITHM

In this test we decided to use three different limited-memory quasi-Newton-like conjugate-gradient methods as our local minimizers. These included the Shanno & Phua (1980) CONMIN code, found by Navon & Legler (1987) to be a very robust and reliable code for large-scale nonlinear minimization, the E04DGF method which is the NAG 12 (1988), update implementation of the PLMA method of Gill & Murray (1979) and, finally, a new code which was made available to us by Dr Jorge Nocedal of Northwestern University (see Liu & Nocedal, 1988).

This code implements a limited-memory Broyden–Fletcher–Goldfarb–Shanno (BFGS) (see Luenberger, 1984) quasi-Newton method called L-BFGS. It uses a number m of BFGS corrections stored as a function of memory availability and uses them in a circular way, i.e. the oldest correction is deleted and the newest one is inserted. During the first m iterations this method is identical to the BFGS method, while for $k > m$, the Hessian is obtained by applying m BFGS updates to H_0—a sparse symmetric positive definite matrix which may be chosen as the identity matrix.

Apart from the CONMIN, E04DDGF and L-BFGS we also tried the quasi-Newton method with BFGS updates from the CONMIN routine of Shanno & Phua (1980). The results are summarized in Table 1.

For the numerical results in Table 1, two basic trends emerge. As far as the number of iterations to convergence is concerned (using the stopping criteria $\| \mathbf{g}_i \| < 10^{-5} \cdot \max(1, \| \mathbf{x}_k \|)$, where $\| \cdot \|$ denotes the Euclidian norm), the Shanno & Phua (1980) CONMIN algorithm emerges as the first. It is followed closely by the L-BFGS method of Liu & Nocedal (1988) (with the number of BFGS updates, $m = 5$) and then by the E04DGF algorithm of Gill & Murray (1979). The performance of the BFGS method for this particular test problem was particularly poor.

Table 1. Comparison of different limited-memory quasi-Newton local minimizers

	Quasi-Newton	CONMIN	E04DGF	L-BFGS
CPU (s)	68.11	19.78	20.78	15.94
Average No. of iterations	70.6	27.2	31.20	29.60

Let us define the iteration time as the time needed to generate the search direction, perform the line-search and test convergence as well as the time to evaluate the function and its gradient. The ranking of the limited-memory quasi-Newton methods by CPU time differs from the ranking by the number of iterations.

The L-BFGS algorithm is the fastest, followed by the CONMIN algorithm and then by the E04DGF algorithm. The quasi-Newton BFGS method is totally outperformed in this case, requiring about a factor of 3 more CPU time. This is a problem-dependent result and more research is required into the structure of the Hessian of our problem to determine the reasons of the poor results with the full quasi-Newton BFGS method. In the ranking of L-BFGS, CONMIN and E04DGF algorithms, our results agree with the recent results of Liu & Nocedal (1988).

7. LIMITED-MEMORY QUASI-NEWTON METHODS

Limited-memory quasi-Newton methods can be viewed as an extension of the conjugate-gradient methods in which addition of modest storage serves to accelerate convergence of large-scale unconstrained local minimization. They were first proposed by Perry (1976, 1978) and generalized by Shanno (1978a, b). The memoryless (or limited-memory) quasi-Newton method is based upon the idea of computing the direction of search

$$\mathbf{p}_k = \alpha_k \mathbf{d}_k, \tag{5}$$

where α_k is the step-size, as

$$-H_k \mathbf{g}_k, \tag{6}$$

where H_k is an approximation to the inverse Hessian of the functional $f(\mathbf{x}_k)$ we wish to mimimize, while $\mathbf{g}_k$ is the gradient of $f(\mathbf{x}_k)$. For limited-memory quasi-Newton methods there is a symmetric matrix obtained by updating the identity matrix I with a limited number of quasi-Newton updates all consisting of rank-two matrices.

The matrix H_k is never computed explicitly, nor stored, and the only additional storage consists of vectors defining the updates. Various limited-memory quasi-Newton methods may be defined, depending on the number of updating vectors stored (see Nazareth, 1979). Shanno (1978a, b) developed his limited-memory quasi-Newton method by using an observation of Perry (1976, 1978) that if

$$\beta_k = \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{y}_k^T \mathbf{d}_k}, \tag{7}$$

where $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, then the new direction of search, $\mathbf{d}_{k+1}$, is given by

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{d}_k = -\left(I + \frac{\mathbf{d}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{d}_k}\right)\mathbf{g}_{k+1}. \quad (8)$$

If one denotes

$$\mathbf{p}_k = \alpha_k \mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \quad (9)$$

the new search direction using

$$\mathbf{d}_k = -H_k \mathbf{g}_k \quad \text{(quasi-Newton conditions)} \quad (10)$$

as

$$\mathbf{d}_{k+1} = -\mathbf{g}_{k+1}\left[\left(1 + \frac{\mathbf{y}_k^T \mathbf{y}_k}{\mathbf{p}_k^T \mathbf{y}_k}\right)\frac{\mathbf{p}_k^T \mathbf{g}_{k+1}}{\mathbf{p}_k^T \mathbf{y}_k} - \frac{\mathbf{y}_k^T \mathbf{g}_{k+1}}{\mathbf{p}_k^T \mathbf{y}_k}\right]\mathbf{p}_k + \frac{\mathbf{p}_k^T \mathbf{g}_{k+1}}{\mathbf{p}_k^T \mathbf{y}_k}\mathbf{y}_k, \quad (11)$$

one can show that for exact line-searches this formula [i.e. the limited-memory quasi-Newton method of Shanno (1978a)] generates mutually conjugate directions.

Shanno & Phua (1980) proposed an implementation of the above algorithm, i.e. the CONMIN algorithm, which required the storage of seven vectors of length N for $\mathbf{x}_k$, $\mathbf{x}_{k+1}$, $\mathbf{g}_k$, $\mathbf{g}_{k+1}$, $\mathbf{d}_k$, $\mathbf{d}_t$ and $\mathbf{y}_t$. Here $\mathbf{d}_t$ and $\mathbf{y}_t$ are two vectors storing the Beale (1972) restart method designed to accelerate convergence for general functions where there is loss of conjugacy.

Gill & Murray (1979) proposed a two-step BFGS (see Luenberger, 1984) limited-memory quasi-Newton method with preconditioning which was implemented in the NAG-12 (1988) update software library in the code E04DFG. In their preconditioning they use a scaling based on recurring the diagonal of the Hessian approximation produced by the direct BFGS formula.

Nocedal (1980) and Liu & Nocedal (1988) proposed a limited-memory BFGS quasi-Newton method which uses a variable amount of storage. In their approach, the quasi-Newton matrix (the approximation to the inverse Hessian) is updated at every iteration in a circular way by dropping the oldest information and replacing it by the newest information. Finally, Buckley & Lenir (1983, 1985) and Buckley (1989) proposed a quasi-Newton limited-memory method with variable storage, which falls back on the usual limited-memory Shanno & Phua (1980) method when the available storage has been exhausted resulting in a combined conjugate-gradient and quasi-Newton method. All of these methods have been found to be extremely efficient for large-scale unconstrained minimization since they allow the user to exploit his available storage in the most optimal way, by using only a small number of additional vectors of length N, where N is the number of variables of the functional to be mimimized. In addition, they perform very well with inaccurate line-searches.

8. CONCLUSIONS

An investigation was conducted to determine the energy-optimized structures of mixed RGCs containing both Ar and Xe atoms. An SA procedure was improved and speeded up by using a highly efficient conjugate-gradient as a local minimizer.

The FTLBMC procedure related to the discretized SA was developed to study clusters which have the underlying structure of pure RGCs. This method uses heavily the local minimizer to sample the lowest energy minima on the PES and is significantly faster than the SA method combined with a local minimizer.

In general we achieved speed-ups of a factor of 20 or more. For this method where the bulk of CPU time is spent in local minimizations we compared the quasi-Newton method with BFGS updates and three limited-memory quasi-Newton-like conjugate-gradient methods, namely CONMIN, E04DGF and a new method L-BFGS, proposed by Liu & Nocedal (1988), which will be included in the Harwell numerical optimization library. The L-BFGS emerged as the best performer for this particular problem.

Acknowledgements—This work was supported in part by the Florida State University Supercomputer Computations Research Institute, which is partially funded by the U.S. Department of Energy through Contract No. DE-FC05-85ER250000.

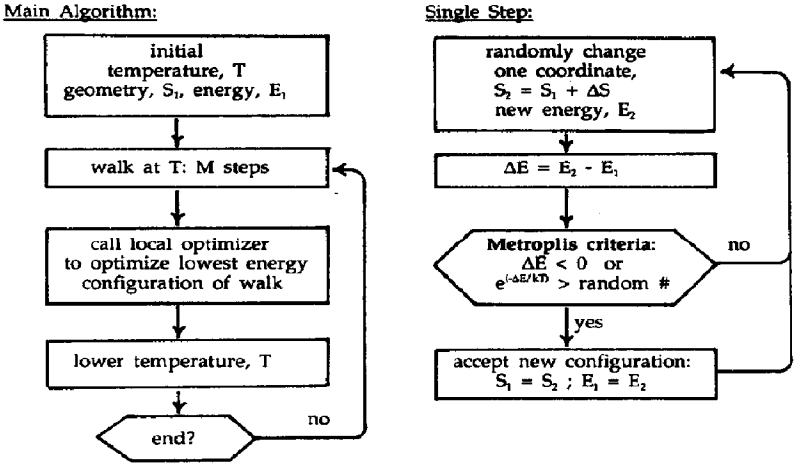
REFERENCES

Beale A. M. (1972) In *Numerical Methods for Nonlinear Optimization* (Edited by Lootsma F. A.), p. 39. Academic Press, New York.
Buckley A. G. (1989) *ACM Trans. Math. Software*, **15**(3), 262.
Buckley A. G. & Lenir A. (1983) *Math. Program.* **27**, 155.
Buckley A. G. & Lenir A. (1985) *ACM Trans. Math. Software* **11**(2), 103.
Germer L. H. (1939) *Phys. Rev.* **56**, 58.
Gill P. E. & Murray W. (1979) Technical Report SOL 79-15, Systems Optimization Lab., Dept of Operations Research, Stanford, Calif.
Hoare M. R. & McInnes J. A. (1983) *Adv. Phys.* **32**, 791.
Hoare M. R. & Pal P. (1971) *Adv. Phys.* **20**, 161.
Kirkpatrick S., Gelatt C. D. Jr & Vecchi M. P. (1983) *Science* **220**, 671.
Li Z. & Scheraga H. A. (1987) *Proc. Natl. Acad. Sci. U.S.A.* **84**, 6611.
Liu D. C. & Nocedal J. (1988) Technical Report NAM 03, Dept. of Electrical Engineering and Computer Science, Northwestern Univ., Chicago, Ill.
Luenberger D. G. (1984) *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Mass.
Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H. & Teller E. (1953) *J. Chem. Phys.* **21**, 1087.
NAG-12 (1988) FORTRAN Library Reference Manual, Numerical Algorithms Group, Mark 12.
Navon I. M. & Legler D. (1987) *Mon. Weather Rev.* **115**, 1479.
Nazareth L. (1979) *SIAM J. Numer. Anal.* **16**, 794.

Nocedal J. (1980) *Math. Comput.* **35,** 773.
Northby J. A. (1987) *J. Chem. Phys.* **87,** 6166.
Perry A. (1976) Discussion Paper No. 229, Center for Mathematical Studies in Economics and Management Sciences, Northwestern Univ., Chicago, Ill.
Perry A. (1978) *Oper. Res.* **26,** 1073.
Robertson D., Brown F. B. & Navon I. M. (1989) *J. Chem. Phys.* **90,** 3221.
Saunders M. (1987) *J. Am. Chem. Soc.* **109,** 3150.
Shah B., Buehler R. & Kempthorne O. (1964) *SIAM J. Appl. Math.* **12,** 74.
Shanno D. F. (1978a) *Math. Oper. Res.* **3,** 244.
Shanno D. F. (1978b) *SIAM J. Numer. Anal.* **15,** 1247.
Shanno D. F. & Phua K. H. (1980) *ACM Trans. Math. Software* **6,** 618.
Vanderbilt D. & Louie S. G. (1984) *J. Comput. Phys.* **56,** 259.
Wille L. T. (1987) *Chem. Phys. Lett.* **133,** 405.

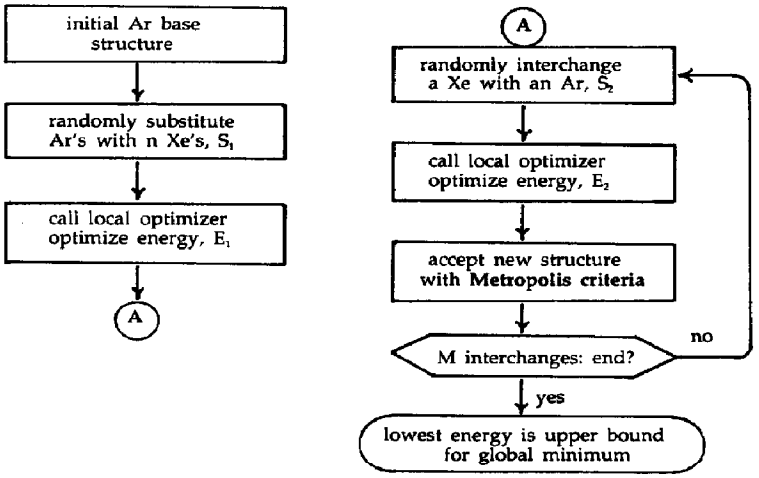
APPENDIX

SA Algorithm



Scheme A1

FTLBMC Method



Scheme A2