PRAMANA
— journal of
physics

Vol. 66, No. 6
June 2006
pp. 1125–1130

On solving energy-dependent partitioned eigenvalue problem by genetic algorithm: The case of real symmetric Hamiltonian matrices

RAHUL SHARMA, SUBBAJIT NANDY* and S P BHATTACHARYYA†
Department of Physical Chemistry, Indian Association for the Cultivation of Science, Jadavpur, Kolkata 700 032, India
*Permanent address: Andrew's High (H.S.) School, Kolkata 700 031, India
†Corresponding author. E-mail: pcspb@mahendra.iacs.res.in

MS received 1 July 2005; revised 10 March 2006; accepted 15 March 2006

Abstract. An energy-dependent partitioning scheme is explored for extracting a small number of eigenvalues of a real symmetric matrix with the help of genetic algorithm. The proposed method is tested with matrices of different sizes (30×30 to 1000×1000). Comparison is made with Löwdin's strategy for solving the problem. The relative advantages and disadvantages of the GA-based method are analyzed.

Keywords. Symmetric eigenvalue problem; genetic algorithm; partitioning techniques; energy-dependent partitioning; Löwdin's method.

PACS Nos 02.60.Pn; 02.70.Hm; 03.65.Fd; 03.65.-w; 31.15.Pf

1. Introduction

Exact solution of the energy eigenvalue equation $\mathbf{H}\psi_n = \mathbf{E}_n\psi_n$ is feasible only for a handful of problems while for a vast majority of the systems, one is compelled to make use of a finite basis $\{\phi_i\}$, preferably an orthonormal one, in terms of which the unknown stationary state wave functions (ψ) are expanded, viz. $\psi = \sum_{i=1}^{N} c_i\phi_i$. A variational ansatz then converts the Schrödinger equation (SE) into a matrix eigenvalue equation $\mathbf{HC} = \mathbf{EC}$, for $\psi = \phi\mathbf{C}$. Since $\hat{H}$ is Hermitian, and the basis used is generally real, the matrix $\mathbf{H}$ is a real symmetric one and the task of solving SE boils down to finding the eigenvalues and vectors of the real symmetric matrix $\mathbf{H}$. There are a variety of algorithms available for solving the matrix eigenvalue problem [1]. Genetic algorithm [2,3] has recently been used in this context with some success [4,5]. We thought it would be worthwhile to explore the feasibility of applying GA to the partitioned matrix eigenvalue problem [6,7]. Traditional deterministic means of solving the energy-dependent partitioning problem have recently been reviewed by Killingbeck and Jolicard [8]. They have concluded that the necessity of inverting large matrices repeatedly undermines the practical usefulness of the

energy-dependent partitioning method. In what follows, we suggest a GA-based approach for handling the problem and compare it's performance with the standard recipe of Loẅdin [6].

2. The method (energy-dependent partitioning)

We start by partitioning the total eigenvector space (N-dimensional) into two subspaces of dimensions n_a and n_b ($N = n_a + n_b$), respectively ($n_a \ll n_b$) where n_a is the number of eigenvalues to be extracted. The Hamiltonian matrix is also similarly partitioned [6–8]:

$$\mathbf{C} = \begin{pmatrix} \mathbf{C}_a \\ \mathbf{C}_b \end{pmatrix}, \quad \mathbf{H} \equiv \begin{pmatrix} \mathbf{H}_{aa} & \mathbf{H}_{ab} \\ \mathbf{H}_{ba} & \mathbf{H}_{bb} \end{pmatrix}. \tag{1}$$

The original eigenvalue equation $\mathbf{HC} = E\mathbf{1C}$ is therefore split into a pair of equations:

$$\mathbf{H}_{aa}\mathbf{C}_a + \mathbf{H}_{ab}\mathbf{C}_b = E\mathbf{1}_a\mathbf{C}_a, \tag{2}$$

$$\mathbf{H}_{ba}\mathbf{C}_a + \mathbf{H}_{bb}\mathbf{C}_b = E\mathbf{1}_b\mathbf{C}_b. \tag{3}$$

Elimination of $\mathbf{C}_b$ from eq. (2) by using eq. (3) leads to the partitioned eigenvalue equation for the a-subspace:

$$\mathbf{H}^a_{\text{eff}}(E)\mathbf{C}_a = \{\mathbf{H}_{aa} + \mathbf{H}_{ab}(E\mathbf{1}_b - \mathbf{H}_{bb})^{-1}\mathbf{H}_{ba}\}\mathbf{C}_a = E\mathbf{1}_a\mathbf{C}_a. \tag{4}$$

An iterative recipe suggested by Löwdin [6,7] for solving the partitioned equation starts by setting $\mathbf{C}_a = 1$ ($n_a = 1, n_b = n - 1$) with an assumed value of $E = E^0$.

In the present approach we focus on eq. (4) and solve the non-linear problem by a stochastic global optimizer like the GA. We start with an assumed value $E^{(0)}$ of E (just as in Löwdin's method), construct the inverse $(E^{(0)} \cdot \mathbf{1}_b - \mathbf{H}_{bb})^{-1}$, form $\mathbf{H}^a_{\text{eff}}(E^{(0)})$, and then refine the targeted eigenvalue of $\mathbf{H}_{\text{eff}}(E^{(0)})$ by suitably invoking genetic algorithm. The refined eigenvalue $E^{(1)}$ is then used as the next input for E to reconstruct $\mathbf{H}^a_{\text{eff}}(E)$ and the process is repeated, till no further improvement in E is possible. GA is used to extract the eigenvalues of $\mathbf{H}^a_{\text{eff}}(E)$ sequentially by first forming the relevant Rayleigh quotient (RQ) $\rho(E)$ [9–11] for the ground state and then maximizing a fitness function constructed with the gradient of $\rho(E)$.

Let ψ_i be the ith member of the evolving population of m trial a-subspace eigenvectors where

$$\psi_i = \sum_{p=1}^{n_a} c_{pi}\phi_p, \quad i = 1, 2, \ldots, n_a, \quad \langle\phi_p|\phi_q\rangle = \delta_{pq}. \tag{5}$$

ψ_i can be represented as a string $\mathbf{S}_i$ of amplitudes c_{pi}:

$$\mathbf{S}_i = (c_{0i}, c_{1i}, \ldots, c_{ki}, \ldots, c_{n_a i}) = \mathbf{C}_i. \tag{6}$$

ρ_i, the RQ for ith wave function string ($\mathbf{C}_i$), its gradient $\nabla\rho_i$ and the fitness f_i are given by

$$\rho_i = \frac{\mathbf{C}_i^\dagger \mathbf{H}_{\text{eff}}^a(E)\mathbf{C}_i}{\mathbf{C}_i^\dagger \mathbf{C}_i}, \tag{7}$$

$$\nabla\rho_i = \frac{2[\mathbf{H}_{\text{eff}}^a(E) - \rho_i \mathbf{1}]\mathbf{C}_i}{\mathbf{C}_i^\dagger \mathbf{C}_i}, \tag{8}$$

$$f_i = \mathrm{e}^{-\lambda(\nabla\rho_i)^\dagger(\nabla\rho_i)}, \tag{9}$$

λ being a user-defined scalar that takes care of exponential overflow or underflow. Clearly, as defined, f_i scales between 0 and 1, and $f_i \to 1$ as $\nabla\rho_i \to 0$, signalling that the wave function string $\mathbf{S}_i = \mathbf{C}_i$ has self-consistently evolved into the ground eigenfunction of $\mathbf{H}_{\text{eff}}^a(E)$. The evolution is brought about by the sequence of genetic operations carried out on the starting population. The operations are (i) selection, (ii) cross-over with a probability p_{c} and (iii) mutation with a probability p_{m}. The selection is fitness proportional and carried out by the standard roulette-wheel procedure [2,3]. Since we are using floating point strings, an arithmetic cross-over operation is used for information exchange. Accordingly, a pair of strings $\mathbf{S}_i$ and $\mathbf{S}_k$ are selected randomly and the cross-over site l is chosen randomly with a probability p_{c}. Two new strings $\mathbf{S}_i'$ and $\mathbf{S}_k'$ (the children strings) are then created by an arithmetic scheme of cross-over [4,5] with probability p_{c} and are allowed to undergo an arithmetic mutation operation with probability ρ_{m} [4,5].

An additional operation called diversification of the population was carried out after every ten generations to maintain enough variability in the population.

The n_a eigenvalues are searched sequentially. First we find out the lowest eigenvalue $\rho_0(E_0)$ and the eigenvector $\mathbf{C}_0^a$ of $\mathbf{H}_{\text{eff}}^s(E_0)$, by applying GA-based Rayleigh quotient search [5]. Our experience so far indicates that only two or three inversions are initially needed to force convergence on the search for the eigenvalue. Once ρ_0 and $\mathbf{C}_0^a$ have been obtained, we find the b-subspace projection of $\mathbf{C}_0$ by evaluating $\mathbf{C}_b^0 = (\rho_0 \mathbf{1}_b - \mathbf{H}_{bb})^{-1}\mathbf{H}_{ba}\mathbf{C}_a^0$ which leads to the full ground eigenvector $\mathbf{C}_0$.

For the next higher eigenvalue ρ_1 of $\mathbf{H}$, we start by forming the projected Hamiltonian $\mathbf{H}_1$ where

$$\mathbf{H}_1 = (\mathbf{1} - \mathbf{P}_0)^\dagger \mathbf{H}(\mathbf{1} - \mathbf{P}_0), \tag{10}$$

where $\mathbf{P}_0 = \mathbf{C}_0\mathbf{C}_0^\dagger$ is the projector for the ground state. $\mathbf{H}_1$ is partitioned in the same way and the GA-based search is carried out similarly. The starting eigenvectors are chosen to be orthogonal to $\mathbf{C}_0$. With successive projections the procedure can be extended to compute all the n_a number of eigenvalues and eigenvectors.

3. Result and discussion

3.1 *General performance pattern*

The algorithm proposed in the previous section has been put to test with a model matrix $\mathbf{H}$ [12] of varying dimensions N (30, 50, 300, 500, 1000) defined as follows:

$$H(i,i) = 2i - 1; \quad H(i,j) = 1, \quad i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, N.$$

For the search of the ground eigenvalue, the initial population of ten trial vectors was chosen by selecting a unit vector $[1, 0, 0, \ldots, 0]^t$ and randomly perturbed unit vectors generated from it. The initial value of $E = E^{(0)}$ was chosen randomly from the range $[-5000, 5000]$. We have used a two-point cross-over scheme with a cross-over probability $p_c = 0.8$ (fixed). The mixing parameter was chosen as $f_c = 0.75 + 0.2r$ where r is a random number in the range $(0, 1)$. Mutation and diversification probabilities (p_m and p_d) were kept fixed at 0.8 and 0.3 respectively.

Figures 1a–e display the evolution of the fitness of the best string in the population as functions of generations when the ground eigenvalue was being searched for different partition sizes. The growth of fitness was nearly exponential with saturation in about 30 generations.

When the dimension of the a-subspace was increased to 10, the evolution profiles of the fitness function was dramatically different for $N = 1000$ (figure 2e). For all other dimensionalities of the $\mathbf{H}$ matrices ($N = 30, 50, 300, 500$), the behaviour noted was more or less similar to what was observed for $n_a = 5$ (figure 2a–d). For $N = 1000$, the fitness profile of the best evolving string shows rapid initial growth to $f > 0.9$ followed by a large scale decrease in the fitness which is overtaken by steep growth to $f = 1.0$. We note, however, that this fluctuation in fitness profile occurs near the saturation point when an explicit inversion of $(E^{(n-1)}.\mathbf{1}_b - \mathbf{H}_{bb})$ is carried out to check if the convergence is stable.

The performance of the method when it comes to computing higher eigenvalues is illustrated graphically in figures 3a–c for the 15×985 partition, as a test case. The overall performance scenario has been captured in table 1.

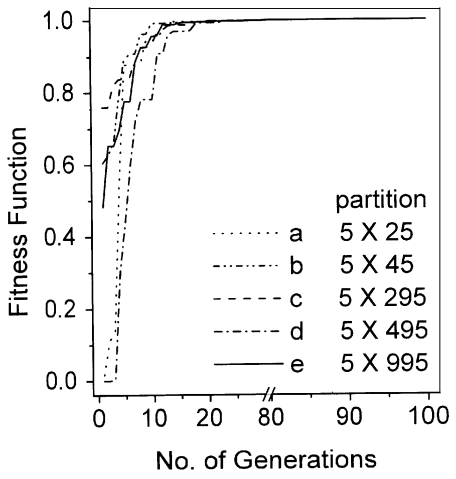

Figure 1. Fitness evolution of the best string during the search for the ground eigenvalue of Coope–Sabo matrix for a partition size of $5 \times (N-5)$: 5×25 (a), 5×45 (b), 5×295 (c), 5×495 (d) and 5×995 (e).

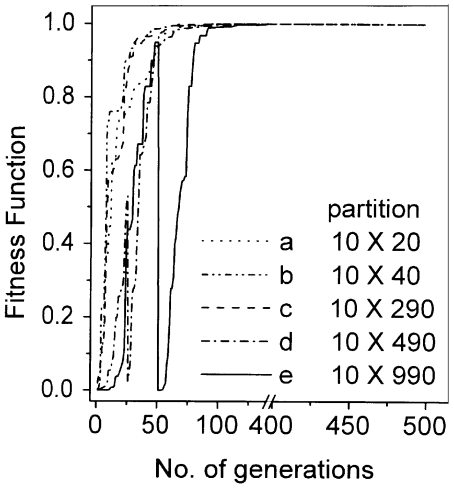

Figure 2. Fitness evolution of the best string during the search for the ground eigenvalue of Coope–Sabo matrix for a partition size of $10 \times (N - 10)$: 10×20 (a), 10×40 (b), 10×290 (c), 10×490 (d) and 10×990 (e).

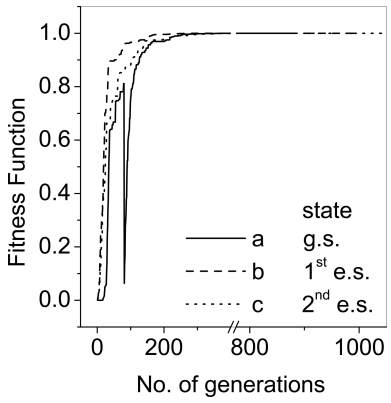

Figure 3. Fitness evolution of the best string during the search for the ground (a), first (b) and second (c) excited state eigenvalue of Coope–Sabo matrix for a partition size of $15 \times (1000 - 15)$.

3.2 *Comparison with Löwdin's strategy*

Table 2 displays the computational time required to compute one eigenvalue of a 500×500 and a 1000×1000 Coope's Hamiltonian matrix by (a) the GA-driven partitioned eigenvalue method and (b) the iterative Löwdin's partitioning method. A relatively small population size of 10 has been used in all the GA-based calculations reported. One can easily see that the GA-driven method performs significantly better in terms of total CPU time consumed in computing one eigenvalue. The relative performance pattern of the two methods is independent of whether the ground or excited eigenvalues are being computed. Parallelization can enhance the performance of the proposed method significantly.

Table 1. Number of generations required to compute the ground, first and the second excited state eigenvalues for the Coope–Sabo matrix of five different dimensions, and for three separate partitions in each case.

Matrix dimension $n_a + n_b = N$		No. of generations spent for obtaining the ground eigenvalue and vector				
Dimension of a-subspace (n_a)	State	30	50	300	500	1000
	g.s.	83	84	76	72	82
5	1st e.s.	70	94	80	75	77
	2nd e.s.	75	69	58	60	71
	g.s.	414	421	423	322	300
10	1st e.s.	441	471	431	365	474
	2nd e.s.	283	426	289	369	370
	g.s.	986	941	927	987	781
15	1st e.s.	943	1142	1064	843	991
	2nd e.s.	905	834	981	1000	1033

Table 2. Comparison of the performance of the GA-driven method and the iterative Löwdin's method for computing one eigenvalue using an energy-dependent partitioning scheme.

Dimension of a-subspace (used in GA)	Time required (in s) to compute the 1st eigenvalue (ground state)		Time required (in s) to compute the 2nd eigenvalue (1st excited state)	
	GA[a]	Löwdin[b]	GA[a]	Löwdin[b]
Dimension of the Hamiltonian matrix **H** is 500×500				
2	10.63	56.4	10.34	56.6
10	10.55		10.41	
20	8.20		9.03	
Dimension of the Hamiltonian matrix **H** is 1000×1000				
2	82.46	460.29	83.10	460.14
10	81.69		81.11	
20	84.31		84.31	

[a]In Löwdin's method a fixed partition size with $n_a = 1$, $n_b = N - 1$ is used throughout.
[b]Population size used in the GA-based scheme is 10.

Acknowledgements

One of us (RS) thanks the CSIR, Government of India, New Delhi, for the award of a Senior Research Fellowship while SPB would like to thank the DST, Government of India, New Delhi, for a research grant.

References

[1] J H Wilkinson, *The algebraic eigenvalue problem* (Oxford University Press, London, 1965)
[2] J H Holland, *Adaptation in natural and artificial systems* (University of Michigan Press, Ann Arbor, MI, 1975)
[3] D E Goldberg, *Genetic algorithms in search, optimisation and machine learning* (Addison Wesley, Reading, MA, 1989)
[4] S Nandy, P Chaudhury and S P Bhattacharyya, *Int. J. Quantum Chem.* **90**, 188 (2002)
[5] S Nandy, P Chaudhury and S P Bhattacharyya, *J. Chem. Sci.* **116**, 285 (2004)
[6] P O Löwdin, *J. Chem. Phys.* **19**, 1396 (1951)
[7] P O Löwdin, *Int. J. Quantum Chem.* **2**, 167 (1968)
[8] J P Killingbeck and G Jolicard, *J. Phys.* **A36**, 12105-R180 (2003)
[9] R K Nesbet, *J. Comp. Phys.* **43**, 311 (1965)
[10] W J Kerner, *J. Comput. Phys.* **85**, 1 (1989)
[11] W Butscher and W E Kammer, *J. Comp. Phys.* **20**, 313 (1976)
[12] J A R Coope and D W Sabo, *J. Comp. Phys.* **23**, 404 (1977)