Study of the Ground-State Geometry of Silicon Clusters
Using Artificial Neural Networks

M.R. Lemes*, L.R. Marim and A. Dal Pino Jr.

Department of Physics - ITA
12228-900 São José dos Campos - SP, Brazil

Theoretical determination of the ground-state geometry of Si clusters is a difficult task. As the number of local minima grows exponentially with the number of atoms, to find the global minimum is a real challenge. One may start the search procedure from a random distribution of atoms but it is probably wiser to make use of any available information to restrict the search space. Here, we introduce a new approach, the Assisted Genetic Optimization (AGO) that couples an Artificial Neural Network (ANN) to a Genetic Algorithm (GA). Using available information on small Silicon clusters, we trained an ANN to predict good starting points (initial population) for the GA. AGO is applied to Si_{10} and Si_{20} and compared to pure GA. Our results indicate: i) AGO is, at least, 5 times faster than pure GA in our test case; ii) ANN training can be made very fast and successfully plays the role of an experienced investigator; iii) AGO can easily be adapted to other optimization problems.

1. Introduction

Artificial Neural Networks (ANN) and other artificial intelligence algorithms have proved to be very useful tools in theoretical and experimental Chemistry. Recently, Gasteiger and Zupan[1] have compiled some of the most important applications of ANN in Chemistry. Some interesting examples include automatic identification of groups of molecular spectrum and the determination of the sequence[1] of amino-acids in a protein. Other important applications are: i) comparison of ANN with quantum mechanical techniques for the prediction of molecular properties for inorganic systems[2]. ii) predictions, made by Sigman and Rives[3], of atomic ionization potentials using shell model parameters as input data for the ANN. These applications encourage us to explore the potential of ANN in yet another field: the prediction of the ground-state geometry of clusters.

Due to the problems in experimental[4,5] production and selection of silicon clusters, traditional methods fail to establish their ground-state geometry. Therefore, one must infer the structure of these clusters, either from indirect evidence or from theoretical calculations. On the other hand,

theoretical calculation of the geometry of the ground-state of a large collection of atoms is an extremely complicated task due to the following reasons: i) most of these problems requires quantum mechanical methods to produce a realistic total energy. These calculations are very demanding of computer resources[6], and ii) the energy hyper-surface depends on a large number of variables and has countless local *minima*. For instance, a cluster composed of ~150 noble-gas atoms[7] has an estimated number of 10^{60} *minima*! An even larger number of local *minima* is expected for covalent materials. Obviously to select the global *minimum* among so many local *minima* is a very difficult task.

ANN atractive features make them useful for modelling, simulation, control and prediction[8-10] in many fields of sciences. In most of these applications, ANN are trained with data collected during operations or experiments. After training, ANN are able to deliver the desired predictions thanks to their natural generalization capability.

Traditionally, practical problems in Chemistry and Physics are transformed into optimization problems. Many algorithms exist to solve these problems and they may be split in two groups: i) gradient based methods[11,12]. For instance,

---

*e-mail: ruv@uol.com.br

the conjugate gradient[12] method is a procedure based on the use of derivatives. These methods are not designed to avoid being trapped by local *minima*. Thus, it must be repeated several times starting from different initial points. The best result of a series of iteractive procedures is assumed as the sought solution; and ii) methods that do not use derivatives[13]. Genetic algorithms[14-16] and simulated annealing[13,17] are optimization methods that do not depend on the calculation of gradients. They imitate natural processes and they are able to overcome barriers to avoid local *minima*. No matter which optimization method, the choice of efficient starting point (or points) is of vital importance for a successful search.

Recently, Cundari and Moody[18] used ANN to predict molecular properties of a series of diatomic molecules. They showed that, after proper training, ANN can predict chemical quantities such as vibration frequency, binding energy and equilibrium distance as accurately as "ab-initio" calculations.

Here, we want to associate ANN to a quantum chemistry method to search for the geometry of the ground-state of silicon clusters. We used ANN to select good starting points for an iterative optimization method. Specifically, an ANN will provide candidate structures for the genetic algorithm. Differently from ref. 18 that used ANN to compare with first-principles calculations, we used the ANN to accelerate the quantum chemistry calculation instead of replacing it. We chose to test the method under stringent conditions. The miniaturization of devices stimulates the interest in the properties of Silicon clusters[19] because silicon remains as the most important element for the development of electronic devices. Therefore, the search for structural models of silicon cluster is technologically motivated because structure determines, in good part, the electrical and mechanical properties of the material[20].

Previous works tried to predict three-dimensional geometry of silicon clusters. First-Principle calculations[21-23] are limited to few atoms. Only small clusters (up to ~ 10 atoms) can be completely investigated[21]. For larger clusters, first-principle calculations are not feasable. For clusters with more than 10 atoms, searches are artificially restricted to models[24,25].Previous attempts were limited to high symmetries or were based on the geometry of the crystal[26], or yet, on the reconstruction surfaces of Silicon[27].

In this work, we used ANN to distinguish the affinity among different atomic layers. Starting with information obtained from small clusters whose energies were previously calculated, we wanted ANN to identify which layers tend to attract each other more strongly. Avoiding sequences of layers that ANN predicts as unfavorable we can keep the search algorithm from wasting valuable time. In this case, ANN learning power plays the role of an experienced investigator.

A small set of training data is used to train the ANN. Obviously. one should not expect that an ANN trained with such information could accurately predict energy values for new clusters. However, it is able to select structures efficiently for the subsequent global optimization algorithm. In our test case, we used the genetic algorithm . Our results show that the ANN significantly increase the efficiency of the algorithm optimization.

Next, we will present how we transformed the chemical problem into a classification problem. Then we discuss, the architecture of the ANN and the results obtained by the combination of the classifier–ANN with the genetic algorithm.

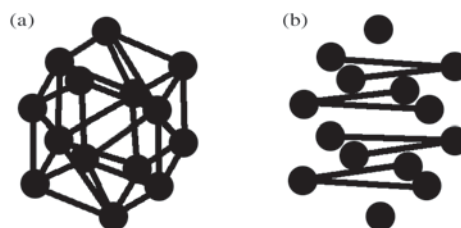## 2. Artificial Neural Network Coupled to Genetic Algorithm

In order to insert geometric information into ANN, we have described the structure of a cluster as a piling up of plane layers of atoms. Such treatment resembles the one presented by Grossman and Mitas[24]. They suggested a geometric description of the silicon clusters as a stacking of triangular elements, with some atoms in the ends, according to Fig. 1.

Figure 1 shows three-dimensional structures described as a series of layers, each one containing three atoms. Here, we described each cluster as a piling up of planar polyatomic layers with up to five atoms. This choice restricts the number of different descriptions of a cluster with **N** atoms.
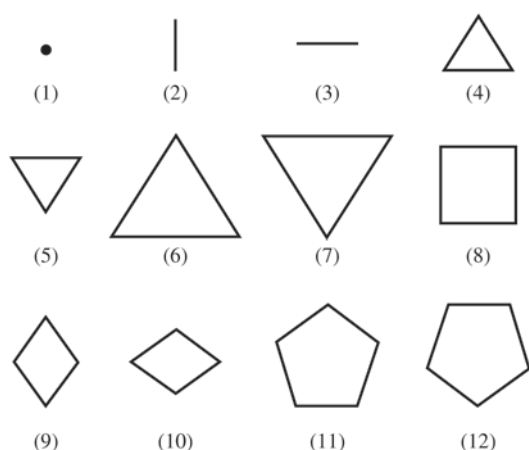
We selected a group of possible structures for the description of the layers. (Fig. 2).

As an example, Fig. 3 shows Si_6 cluster represented by 5 different descriptions, based on the elements of Fig. 1. It is important to point out that the geometric elements used in each description and its associated energy will be used as input data for ANN.
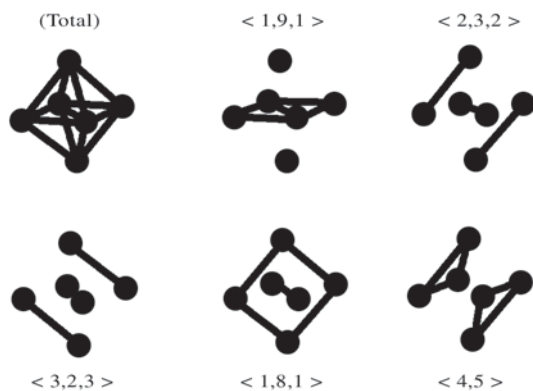
The neural classifier was built to filter the configurations that would be supplied to the genetic algorithm as possible candidates. ANN distinguishes which piling up of atomic layers would probably have high binding energy, i.e., the more stable structures. It also should be possible to

Figure 1. Si_{14} cluster. a) global view of the cluster; b) one can see the sequence of atomic layers.

Figure 2. Layers that are piled up to form the clusters. Notice that each corner corresponds to the position of an atom.

Figure 3. Configurations of Si_6, with their respective description.

Table 1. It shows the structure and the corresponding TB energy, for three of the 110 cases used. The first example describes the structure shown in Fig 3. The second example describes the structure shown in Fig 2.

Structures	binding energy per atom (eV)
3 - 2 - 3	3.0710
4 - 9	3.2670

the ANN. We used the training method known as back-propagation[31]. We have trained ANN to discern the structures appropriate for global minimization from the inadequate ones. Thus, based on the previous knowledge of smaller clusters, ANN distinguishes high binding energy structures and send them to the Genetic Algorithm (GA). Table 1 shows that the binding energy per atom for Si_6 is larger than 3 eV. It is expected that binding energy per atom increases with the number of atoms, we chose 2,8 eV as a reference value. This choice takes into consideration that ANN was trained with very few input data and therefore, it is not expected that ANN predictions should be of quantitative quality. It is fast and simple to expand the training set to use the same approach to other clusters. Training does not need to be very long to yield reliable results, even fast training improves the performance of the Genetic Algorithm. We'll make it clear in the results section.

Finally, the ANN make their predictions. Select the size of the cluster Si_N (N > 9) whose ground-state geometry we want to predict. From every possible combination of layers, the preditor selects those classified as appropriate and it eliminates the others. The "good" structures are sent to the GA in two ways: (i) a certain number of them is used as the GA's initial population; (ii) the remaining ones are introduced in the population of the algorithm through mutations. In other words, every n generations a new structure is introduced into the population. Cluster's binding energy are calculated by the TB approach and this is the quantity that is maximized by the genetic algorithm. In order to test the method just presented, we chose to determine the ground-state geometry of the Si_{10} cluster. This an interesting test because this system possesses many local minima and its energy can be calculated rather quickly in the TB approach.

3. Application and Results

We defined the architecture of our 3 ANN in the following way: an input layer with 11 elements, an intermediate layer and an output layer with 2 elements. In the intermediate layer, we used 12 (ANN_{12}), 6 (ANN_6) and 3 (ANN_3) neurons respectively, whose results will be presented in this section. We have tested sigmoid, hyperbolic tangent and

train it with a quite restricted number of elements in the training group. We divided its preparation in three steps: generation of input data; b) training and; c) prediction.

The first step consists of obtaining information and the necessary elements to the description and characterization of the system. We supplied, as input data for the ANN, the binding energy of 110 clusters. The training group comprises structures of clusters with 9 or less atoms. It is important to point out that the application of the neural classifier is combined to a method of total energy calculation. Any method that we chose would be equally convenient. In this work, we used a Tight Binding (TB) semi-empirical method whose detailed description can be found in the references[28-30].

Next we trained the ANN, adopting as input data the 110 structures and as output data their respective energies. This is an extremely important step because it will determine the quality of the predictions to be accomplished by

gaussian activation functions. Results were not very sensitive to the activation function chosen. The Figs. 4 - 7 correspond to the gaussian activation function.

A fast training is capable of identifying a high percentage of inadequate structures. We decided to stop the training procedure when 60% of the structures of all possible geometries were recognized as inadequate.

The following procedure was performed for each one of the nets. i) the ANN indicates a group of n_p = 10 structures chosen randomly to generate the initial population for a genetic algorithm calculation; ii) cross-over is performed as described in reference 16; iii) every n_m = 10 generations, two new structures, chosen among those considered appropriate by the ANN, replace the "less-fit" elements of the population. This is a special kind of mutation.

Genetic Algorithm, pre-conditioned by each one of the ANN, was executed for 3000 generations and their results compared to those of Pure GA. We represented a N-atom silicon cluster by a list of 3N atomic cartesian coordinates, that is, a chromosome constituted by N genes, each one composed by three coordinates, representing the position of an atom. We use this codification because a bit string, as commonly used, is not very efficient to optimize the geometry of atomic clusters[32]. Crossover probability was defined according to rank selection. Greedy overselection is a procedure designed to improve the population used in GA. Unfortunately, it can only be used if the number of individuals in the population is rather large (> 1000). Since in our case, no more than 10 elements form the initial population greedy overselection cannot help us.

Since genetic algorithm uses random numbers, 10 different calculations for each ANN was considered. We believe that the average of the 10 calculations reliably demonstrates the characteristics of this new procedure. Figures 4, 5 and 6 show a comparison between the best calculation and their average, with a pure genetic algorithm, i.e., genetic algorithm without ANN. Notice that these graphs present the evolution of the opposite of the binding energy per atom as function of the number of generations. Thus, the most stable structures corresponds to the smallest values of energy.

Figure 4 shows the performance of GA coupled to ANN_{12}. One notices that while the pure genetic calculation takes about 4500 generations to find structures with binding energy per atom larger than 3 eV, ANN_{12} best calculation reached the same mark with only 500 generations! The average of 10 runs, reached 3 eV after just 1600 generations.
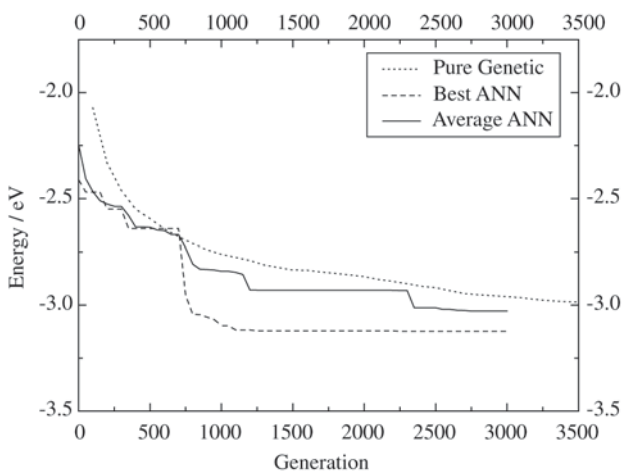
Figure 5 shows the performance of GA coupled to ANN_6. One notices that while the pure genetic calculation takes about 4500 generations to find structures with binding energy per atom larger than 3 eV, ANN_6 best calculation reached the same mark with only 300 generations! The av-
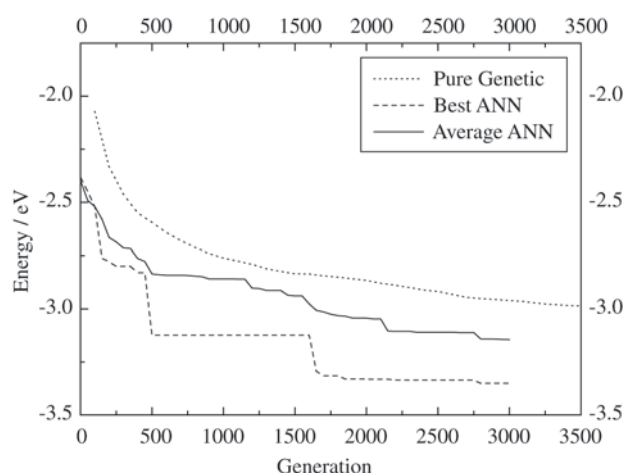
Figure 4. The comparative evolution for hidden layer with 12 neurons.
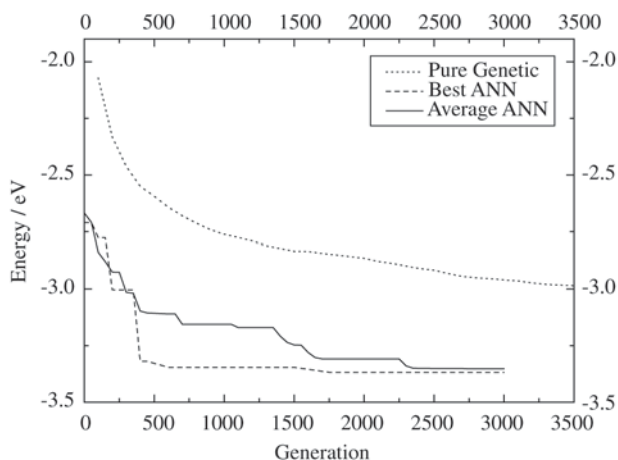
**Figure 5.** The comparative evolution for hidden layer with 6 neurons.

**Figure 6.** The comparative evolution for hidden layer with 3 neurons.

erage of 10 runs, reached 3 eV after just 300 generations.

Figure 6 shows the performance of GA coupled to ANN_3. One notices that while the pure genetic calculation takes about 4500 generations to find structures with binding energy per atom larger than 3 eV, ANN_3 best calculation reached the same mark with only 800 generations! The average of 10 runs, reached 3 eV after about 2000 generations.

Figures 4-6 show that ANN dramatically reduce the total number of generations that a genetic algorithm needs to
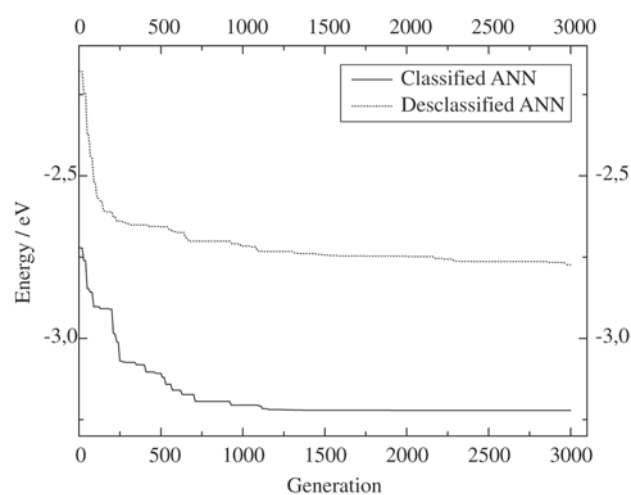
Figure 7. Structures appropriate x Structures considered inappropriate.
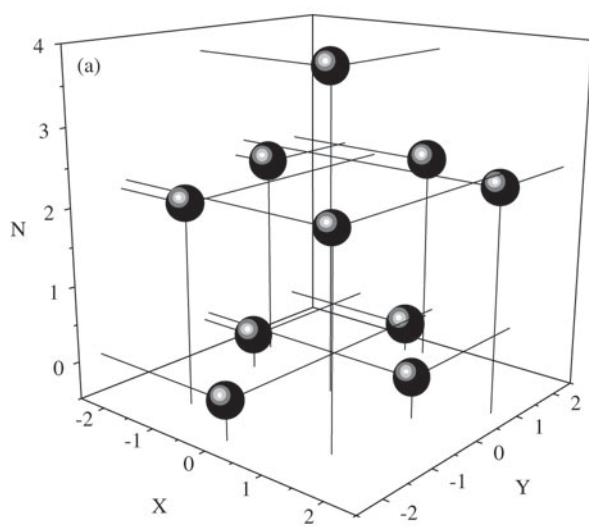
reach our goal, i.e., to find the global minimum. Another interesting observation is that ANN_3 faces more difficulties to get satisfactory results then ANN_6, ANN_{12}. It means that too small an ANN may not be efficient to perform generalization. On the other hand, as ANN_6 and ANN_{12} present comparable performances, it means that one does not need a large ANN to make our approach to work properly.

As a further test we decided to analyze the performance of a GA that used the combined result of the three ANN presented. Only those structures that were simultaneously considered appropriate by all ANN were used to generate the population for the GA. Another calculation was performed using only those structures considered inappropriate by all ANN. These results are shown in Fig. 7.

Figure 8a shows the best geometry obtained after applying AGO for 3000 generations. Figure 8b shows the best geometry obtained by pure GA after the same number of generations. One can easily notice that GA's "predict" structure still has a long way to go before finding the ground state geometry.

Next, we used ANN6 to select reasonable candidate geometries for Si_{20}. Figure 9 compares AGO with pure GA's performances. One notices that it takes more than 100 generations of pure GA to reach the starting value of AGO!

4. Conclusions

We used total energy information for small silicon (Si_n, $n \leq 9$) clusters to train ANN. The training followed standard back-propagation procedure and our only concern was to keep it fast. Next, we took advantage of ANN natural ability to recognize affinity between layers of silicon
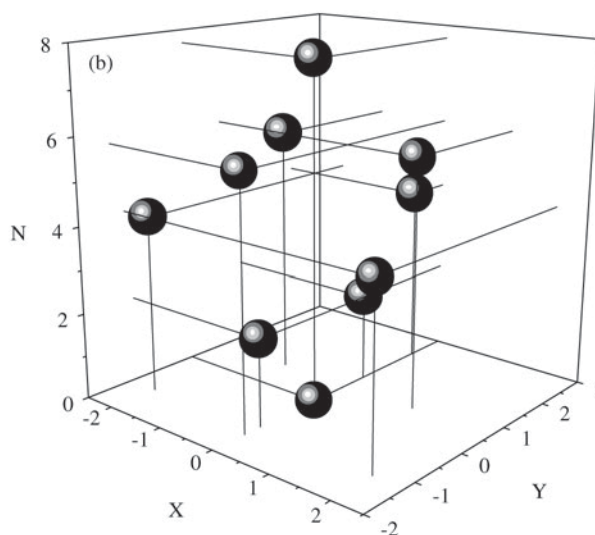
Figure 8. The structure Si_{10} with AGO a) and without AGO b).
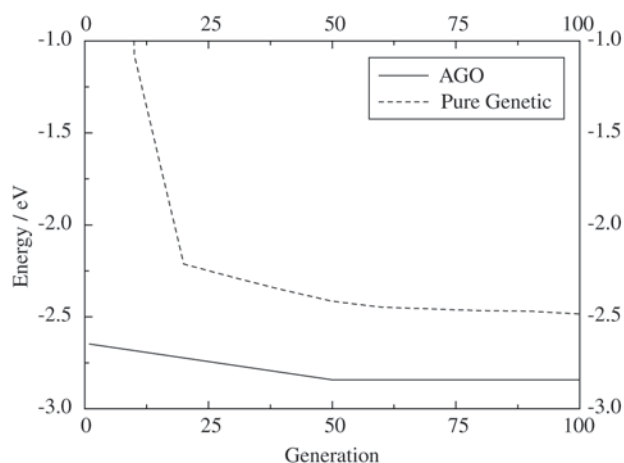
Figure 9. The comparative evolution for Si20.

atoms. Thus, it yields candidates solutions to the Genetic Algorithm. This kept the search algorithm from wasting time.

Our results showed that artificial neural networks can be trained to incorporate information from quantum mechanics and to accelerate total energy calculations of polyatomic systems. All three different ANN (ANN3, ANN6, ANN12) could improve GA's performance if compared to Pure GA. After a fast training procedure, ANN's select efficient starting points for methods of global optimization. If one generation is taken as time unit (tu), training takes about 70 tu. Thus AGO saves at least 2000 tu to reach the ground state geometry for Si_{10}! We consider this method very promising to be adapted for larger cluster (Si_n n> 10) because each generation would take more time but the training time would remain the same.

Finally, our algorithm can be easily adapted for other materials, for other methods of total energy calculation and yet for other optimization problems.

References

1. Gasteiger, J.; Zupan, J. Neural Networks for Chemists; VCH: Weinheim, 1993.
2. Cundari, T.R.; Moody, E.W. *J. Chem. Inf. Comp. Sci.*, v. 37,p. 871-875, 1997.
3. Sigman, M.E.; Rives, S.S. *J. Chem. Inf. Comput. Sci.*, v. 34, p. 617, 1994.
4. Bloomfield, L.A.; Freeman, R.R.; Brown, W.L. *Phys. Rev. Lett.*, v. 54, n. 20, p. 2246, 1985.
5. Jarrold, M.F.; Constant, V.A. *Phys. Rev. Lett.*, v. 67, n. 21, p. 2994, 1991.
6. Remler, D.K.; Madden, P.A. *Mol. Phys.*, v. 70, p. 921, 1990.
7. Wales, D.J.; Doye, J.P.K. *J. Phys. Chem. A*, v. 101, p. 511, 1997.
8. He, F.; Sung, A.H. IASTED International Conference on Control, New Mexico, 1997.
9. Narendra, K.S.; Parthasarathy, K. IEEE Trans. on Neural Networks, v. 1, n. 1,p. 4, 1990.
10. Drossu, R.; Obradovic, Z. IEEE Computational Sciences and Engineering, v. 3, p. 78, 1996.
11. Polak, E. Computational Methods in Optimization: A Unified Approach. New York and London, Academic Press, 1971.
12. Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flanery B.P. Numerical Recipes: The Art of Scientific Computing; 2nd ed., Cambridge University Press, Cambridge, 1992.
13. Lemes, M.R.; Zacharias, C.R.; Dal Pino, Jr. A. *Phys. Rev. B*, v. 56, n. 15, p. 9279, 1997.
14. Holland, J.H. Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1989.
15. Holland, J.H. *Sci. Am.*, July, p. 44, 1992.
16. Zacharias, C.R.; Lemes, M.R.; Dal Pino, Jr. A. *J. Mol. Struc. (Theochem)*, v. 430, p. 29, 1998.
17. Kirkpatrick, S.; Gellar, D.C.; Vechi, M.P. *Science*, v. 220, n. 4598, p. 671, 1983.
18. Cundari, T.R.; Moody, E.W. *J. Chem. Inf. Comput. Sci*, v. 37, p. 871, 1997.
19. Ho, K.M. *Nature*, v. 392, p. 582, 1998.
20. Somorjai, G.A. Chemistry in Two Dimensions: Surfaces. Ithaca: Cornell Univ. Press, 1981.
21. Raghavachari, K.; Rohlfing, C.M. *J. Chem. Phys.*, v. 89, n. 4, p. 2219, 1988.
22. Car, R.; Parrinello, M. *Phys. Rev. Lett.*, v. 55, n. 22, p. 2471, 1985
23. Grossman, J.C.; Mitas, L. *Phys. Rev. Lett.*, v. 74, n. 8, p. 1323, 1995
24. Grossman, J.C.; Mitas, L. *Phys. Rev B*, v. 52, n. 23, p. 16735, 1995.
25. Kaxiras, E.; Jakson, K.A. *Z. Phys. D.*, v. 26, p. 346, 1993.
26. Chadi, D.J.; Cohen, M.L. *Phys. Stat. Sol. B*, v. 68, p. 405, 1975.
27. Chadi, D.J. *Phys. Rev. B*, v. 29, n. 2, p. 785, 1984.
28. Laasonen, K.; Nieminen, R.M. *J. Phys. B:Condens. Matter*, v. 2, p. 1509, 1990.
29. Menon, M.; Subbaswamy, K.R. *Phys. Rev. B*, v. 47, p. 12754, 1993
30. Wang, C.Z.; Chan, C.T.; Ho, K.M. *Phys. Rev. Lett.*, v. 66, n. 2, p. 189, 1991.
31. Hertz, J.; Krog, A.; Palmer, R.G. Introduction to the Theory of Neural Computation, Addison-Wesley Publishing Company, 1991.
32. Deaven, D.M.; Ho, K.M., *Phys. Rev. Lett.*, v. 75, p. 288, 1995.