Finding Multiple First Order Saddle Points Using a Valley Adaptive Clearing Genetic Algorithm

Mostafa M. H. Ellabaan
School of Computer Engineering, Nanyang Technological University, Singapore
most0001@ntu.edu.sg

Yew Soon Ong
School of Computer Engineering, Nanyang Technological University, Singapore
asysong@ntu.edu.sg

Meng Hiot Lim
School of Electrical & Electronic Engineering, Nanyang Technological University, Singapore
emhlim@ntu.edu.sg

Kuo Jer-Lai
School of Physical & Mathematical Sciences, Nanyang Technological University, Singapore
jlkuo@ntu.edu.sg

***Abstract*-First order saddle points have important applications in different fields of science and engineering. Some of their interesting applications include estimation of chemical reaction rate, image segmentation, path-planning and robotics navigation. Finding such points using evolutionary algorithms is a field that remains yet to be well investigated. In this paper, we present an evolutionary algorithm that is designed for finding multiple saddle points. In contrast to earlier work [1], we propose a new fitness function that favors 1st order saddle points or transition states. In particular, a valley adaptive clearing multi-modal evolutionary optimization approach is proposed to locate and archive multiple solutions by directing the search towards unexplored regions of the search space [2]. Experimental results on benchmark functions and the Lennard Jones Potential are presented to demonstrate the efficacy of the proposed algorithm in locating multiple 1st order saddle points.**

Key Words: First orders saddle points, Evolutionary Optimization, Robotics navagation, and Transition states

I. INTRODUCTION

Finding First order saddle points are important for robotics scholars and professionals. First order saddle points, for example, plays an important role in robotic navigation. In [3], first order saddle points has been used as a high accurate approach for identifying x-junction from images captured by robotic visual sensors. Building roadmaps in visual models, also, represent another important general-purpose application of first order saddle points in computer science and robotics [4]. The roadmaps could be used, for example, in estimating human pose captured from monocular images and identifying the minimum energy path which is important for path-planning and robotic navigation [5].

Finding first order saddle points is also important for many other areas in science such as economy [6], chemistry [7], and biology [8]. In theoretical chemistry, for example, geometry optimization of chemical structures, especially transition structures, is important. Transition structures or 1st order saddle points, unlike other stable structures, are energy maxima along the minimum energy path connecting two stable isomers of a given cluster or reactant and product for a chemical reaction, posing fundamental difficulties in finding such structures. For their fleeting nature, transition structures are impossible to be isolated experimentally. Therefore, optimizing such structures computationally is unavoidable [1].

Evolutionary algorithm or EA is a population-based stochastic method, where a population of individuals probabilistically roams the problem's search space**.** By analogy of natural evolution of species, individuals evolve through various genetic operators to better solutions. These operators combine and mix the genetic information of the parents to generate offspring. The goal is to create offspring which are fitter than the parents. The fitness of an individual is often defined numerically or otherwise. The process continues for a number of generations until the specified convergence criteria are satisfied. Evolutionary algorithms have been used extensively for optimizing complex systems. For instance, a motivating example for us is the optimization of optimal chemical structures [9-16].

EA possesses many advantages in finding transition states over other classical deterministic numerical methods. Firstly, EA is well established for its efficiency in sampling search space to locate the global optimum. Secondly, it does not require any prior knowledge about the landscape. In contrast to classical numerical methods where derivatives information is mandatory, only the fitness of a given potential input vector is necessary to guide evolution search. Thirdly, a well-known strength of EAs is their ability to partition the population of individuals among multiple computing nodes. Doing so allows sublinear speedups in computation and even super-linear speedups [12, 16]. Further, unlike classical numerical methods which converge to first order saddle points or transition structures only if the initial guesses were very close to the saddle points - which are unlikely in most cases due to the effect of curse of dimensionality. EA, on the other hand, can converge to saddle points even if started far apart from the solutions. However, the canonical and typical EAs are designed to bias toward single solution. To locate multiple optimal solutions in multi-modal problems using EA, diversity-assuring and multiple-solution maintaining schemes have been introduced [2].

In this paper, we introduce a new fitness function that favors 1st order saddle points or transition states. We also present a valley adaptive clearing multi-modal genetic algorithm for finding multiple 1st order saddle points. The essential backbone of our algorithm is a valley adaptive clearing scheme [2] for multi-modal optimization. In the algorithm, the initial population of individuals is sampled to sit in unique valley or basin of attraction. The reproduction operators are employed on the population individuals, leading to offspring. Yielding offspring sharing a common valley are subsequently grouped together and categorized into elites and inferiors according to their fitness. Elites are enforced to survive to the next generation, while the inferiors are relocated to unexplored area of the search space. The next evolutionary iteration proceeds until the maximum number of generations is exceeded, or pre-specified stopping criteria are satisfied.

The paper is organized as follows: Section II provides a brief definition of the non-linear programming problem. Related work is presented in Section III. The details of the proposed method are presented in section IV while section V reports the results obtained from our computational study. A brief conclusion and future work are then stated in Section VI.

II. Problem Statement

A transition state or a 1st saddle point is a stationary point with a vanished gradient and one and only one negative eigenvalue in the hessian matrix [7]. These points can be mathematically expressed as:

$$\mathbf{X_T} = \left\{ \mathbf{x_i} \middle| \left(\frac{\delta f(\mathbf{x_i})}{\delta \mathbf{x_i}} = 0 \right) \,\&\, (e_{ik} < 0) \right\} \tag{1}$$

Where $\mathbf{x_i} \in \mathbf{R^d}$, d is the dimensional size, $f(\mathbf{x_i}) \in \mathbf{R}$, $\mathbf{X_T}$ is the set that includes all possible 1st order saddle points, and e_{ik} is the only negative eigenvalue of the hessian matrix $\mathbf{H_i}$, respectively. '&' denotes a logical AND operator.

III. Related Work

In the last decades, many methods have been proposed to tackle the problem of finding first order saddle points or transition states. These methods can be classified into two categories: deterministic and stochastic methods.

Deterministic methods are the ones in which the gradient and/or hessian information is used to find the 1st order saddle points. These methods can be further classified into doubly-ended and single-ended methods [17-21]. In doubly-ended methods, two minima are required in order to find saddle points in between. The accuracy of the saddle points remains unresolved on most of doubly ended methods. The main advantage of such methods is the ability to converge quickly to the area near first order saddle points. However, it hardly converges to first order saddle points precisely. In contrast to doubly-ended methods, single-ended methods require no information about the location of the transition [20, 21]. However, it requires a good initial guess to find 1st saddle points.

In stochastic search methods, instead of utilizing only one or two solution(s) to find saddle points, a population of individuals is used instead to explore the search space for saddle points [1]. To date, there have been few stochastic methods proposed for finding first order saddle point, making it a fertile area for further research investigation. Chaudhury et al. suggested a simulated annealing method to locate saddle points [22]. Bungay et al. [1] also suggested a GA method that utilized the number of negative eigenvalue to bias the evolutionary search towards 1st order saddle points. Both previously mentioned methods utilized second derivative information (hessian matrix). Chaudhury et al. [23] also described a GA that estimates the eigen values from the gradient information, with the assumption that these points lie on the minimum energy path. For such assumption to hold, the GA must start its search from the global minimum configuration [23]. In addition, its trajectory search nature limits parallelism of GA.

IV. Valley Adaptive Clearing Genetic Algorithm for Finding Mutliple Saddle Points

Here, a real-coded valley-adaptive clearing genetic algorithm is proposed to locate multiple saddle points. In this algorithm, the population of individuals is initiated with each individual falling in a different basin of attraction, using a *hill-valley detection scheme*[1]. Each individual $\mathbf{x}$ is evaluated using Eqn. (2)

$$f(\mathbf{x}) = \begin{cases} \frac{-1}{(\|\mathbf{g}\|+\delta)^d} & if\ n = 1 \\ l & if\ n \neq 1 \end{cases} \tag{2}$$

where $\| \mathbf{g} \|$ is the L_2-norm of the gradient vector; n is the number of negative eigen values, ι is a real number greater zero, and δ denotes a small numerical value to avoid any division by zero error. l is a small positive number.

Individuals then undergo selection, mutation and crossover. Thereafter, the yielding offspring undergoes the valley adaptive clearing scheme [2] which involves the *identification*, *clearing* and *valley replacement* phases.

A. Hill-Valley Detection

The hill-valley detection procedure [24] begins by generating a line connecting two given points in the Euclidean space. Subsequently, a number of intermediate points are sampled within the line. The fitness values of these points are then calculated. A valley existence, on one hand, is identified, if the fitness of any sampled points represents an improvement over that of the given points. Otherwise, a hill is established.

[1] A minimization of the fitness function Eqn. (2) is assumed.

B. Valley-Adaptive Clearing Scheme

The valley-adaptive clearing scheme is proposed to adapt to non-uniform width of valleys in the fitness landscape. The main idea of the valley-adaptive clearing scheme is to group the population individuals into niches based on whether they share a common valley or not. Subsequently, the lowest fit individuals of a common valley, i.e., from the same niche, are relocated to unexplored area of the landscape [2].

The valley-adaptive clearing scheme is composed of three core phases. The valley identification phase categorizes the population of individuals into groups of individuals sharing the same valley, denoted as $\mathbf{V}_{\text{groups}}$. Subsequently, the dominant individual (i.e., in terms of fitness value) of a valley group or $\mathbf{V}_{\text{id}}$ is archived if it represents a unique first order saddle solution, while all other members of the same group undergo the valley replacement phase where relocation of these individuals to new basins of attractions or valleys are made so that first order saddle solution elsewhere may be uncovered. In the event that no first order saddle solution exists in a valley group, all individuals of the group will undergo the valley clearing stage where elite individuals are ensured to survive across the search generation while all others are relocated to new basin of the attractions.

1) Valley Identification Phase

The procedure of valley identification begins with the sorting of population individuals in ascending order according to fitness. Individuals are then grouped together if they share a common valley. Individuals belonging to the same valley group are then categorized according to their fitness into *elites* and *inferiors*. Elites are the fittest *k* individuals in a group, while the remaining individuals are the inferiors.

2) Valley Clearing Phase

Valley clearing is a process in which less fit individuals (or inferiors) are relocated out of the same basin of attraction, leaving valleys to be further exploited by the fittest individuals (elites). In the valley clearing process, each inferior member (**x**) of the valley group ($\mathbf{V}_{\mathbf{id}}$) is relocated randomly in the range of $\mathrm{Q}_{\mathrm{clear}}$, i.e., the clearing niche radius, to 3 * $\mathrm{Q}_{\mathrm{clear}}$, whereas other individuals (or elites) are left unchanged for the purpose of exploiting the basin of attraction.

3) Valley Replacement Phase

The motivation behind valley replacement process is to reduce any computational resources wasted on rediscovering of valleys where the 1st order saddle point solutions have already been uncovered. Individuals of the populations falling in previously encountered valleys are replaced with individuals in new basins of attractions, so as to bias the search towards previously unexplored region of the landscape.

C. Archiving Procedure

All first order saddle point solutions found throughout the search are archived using an indexed database. Two data structures, the first is the array(s) or list(s) of discovered solutions, while the second is a hierarchical index or tree. Its nodes represent all cluster centers of the solutions found throughout the search, organized in a hierarchical manner according to the spatial order between solutions. The lists of solutions lie at the leave nodes of the index tree.

To keep the archive free of duplicates, we proposed a hybrid archiving procedure that combines a distance metric with hill-valley detection procedure to detect duplicates in the archive. In this procedure, the hill-valley detection procedure is employed only on selected archived saddle points that fall within a predefined distance of a saddle point.

V. Empirical Study

In this section, we begin first with a study on the efficacy of the valley adaptive clearing genetic algorithm (AVAC), taking Bungay's GA (BGA), other archiving clearing GAs (AC – Archiving clearing GA, AMC – archiving modified clearing GA) [2] and the stochastic multi-start Dimer local search (SMLS) (a classical method for finding transition) [20] as the baseline for comparison, using several multi-modal benchmark test problems. The test problems considered in the study and computational results obtained are presented in Sections A and B, respectively. The landscape of all problems considered in the present study contains significant number of critical points including maxima, minima and saddle points.

A. Benchmark Test Problems

Problem 1. The 2D Rastrigin Function Test Case

The 2D Rastrigin function Eqn. (3) is a typical multi-modal benchmark test problem used in evolutionary computation research. For the range of [-5,5], the landscape contains approximately 223 1st order saddle points[2].

$$f(x,y) = 20 + x^2 + y^2 - 10(cos\,2\pi\,x + cos\,2\pi\,y) \quad \textbf{(3)}$$

Problem 2. 2D Multi-function test Case

The 2D Multi-function Eqn. (4) is a challenging test case for saddle point algorithms. It has approximately 144 saddle points[2] within the range of [-2, 2].

$$f(x,y) = -1 - x\sin(4\pi\,x) + y\sin(4\,\pi\,y + \pi) \quad (4)$$

Problem 3. The Sines Function Test Case

[2] The number of 1st order saddle points is approximated by a mean of visualization.

The 2D Sines function Eqn. (5) is multi-modal test function with approximately 84 1st order saddle points[2] within the range of [-10, 10].

$$f(x,y) = 1 + sin^2(x) + sin^2(y) - 0.1e^{-x^2-y^2} \quad (5)$$

Problem 4. The Three-Atom Lennard–Jones cluster

One of the important molecular systems is the Lennard Jones clusters. Lennard Jones potential describes the interaction between atoms in molecular dynamics:

$$f(\mathbf{x}) = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} \varepsilon \left[\left(\frac{r_0}{r_{ij}}\right)^{12} - 2\left(\frac{r_0}{r_{ij}}\right)^{6}\right] \quad (6)$$

where **x** represents the vector of atomic configurations, while ε, r_0 and N denoting the depth of the potential well, (finite) distance at which the interparticle potential is zero and atom size, are configured as 1, 1 and 3, respectively. Finally, r_{ij} denotes the Euclidean distance between atoms *i* and *j*.

B. Numerical Results

In this section, the computational settings and results are reported. All methods used here to benchmark against and the proposed method are implemented in MATLAB development environment and investigation executed on a PC with 2.66 GHz Intel Duo Core CPU and 3 GB RAM. The following algorithmic configurations are also considered: uniform mutation probability of 20%; scatter crossover probability of 60%, 20% of the population represents the elites and stochastic uniform sampling based selection. The number of sample points (m) in the hill-valley detection, the number of elites (k), the archiving-distance threshold (τ) and the clearing niche radius (Q_{clear}) are configured at 5, 2, 0.5 and 0.25, respectively. A population size of 100 is considered for all benchmark problems, and the search terminates at a maximum generations of 100.

Performance of different GAs is compared in terms of the number of optima and the execution time. The fitness of the individuals is evaluated based on Eqn. (2). The numerical results are summarized in Figures 1-2.

In Figure 1, percentages of 1st order saddle points uncovered by different archiving clearing GAs and Bungay GA are presented. The plotted results show that the archiving clearing GAs maintains the highest percentages of uncovered 1st order saddle points on each of the different benchmark problems. The percentages of saddle points discovered by the archiving clearing GAs vary in the range of 10-100%, whereas those uncovered by BGA and SMSL vary in the range of 1-8%.

Among the clearing methods, the valley adaptive clearing GA or AVAC maintained the highest percentage of 1st order saddle points, varying in the ranges of 94-100%. On the other hand, those uncovered by the other clearing compeers vary largely in the range of 25-55%, respectively.

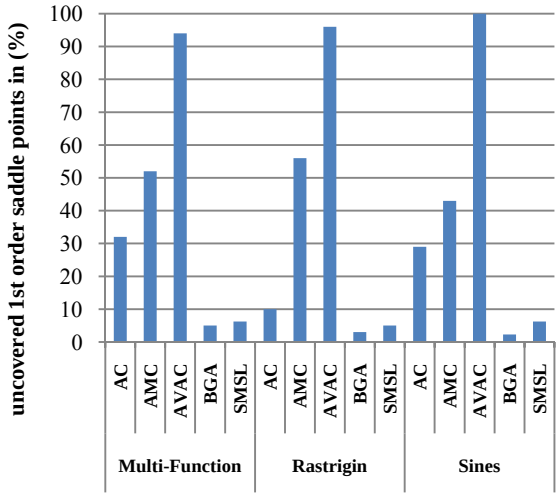

Figure 1. Percentages of uncovered 1st order saddle points by different GAs, on the three benchmark problems.

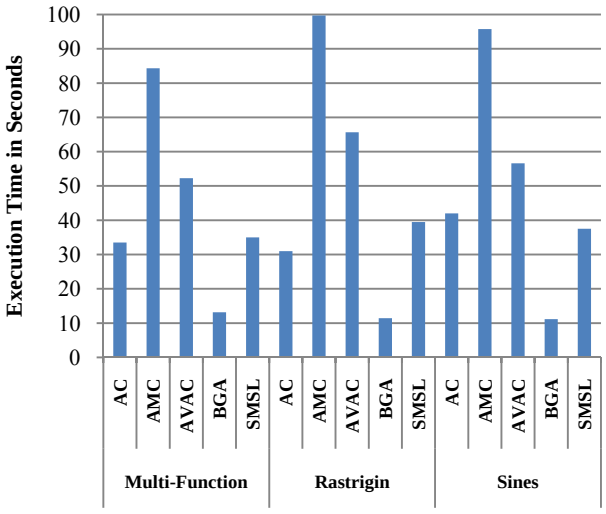

Figure 2. Execution time of different methods, on the three benchmark problems.

Figure 2 shows the execution time of different GAs on three benchmark problems. The clearing GAs execution time tends to be reasonably higher than BGA. However, in complex optimization where function evaluations are computationally expensive [14], the archiving overhead may be regarded as negligible.

We have also employed the algorithms on the realistic three-atom Lennard-Jones cluster problem. The proposed valley adaptive clearing genetic algorithm (AVAC) uncovered the highest average of 355.5 first order saddle points, whereas the other methods, namely, AC, AMC, BGA, and SMLS uncovered averages of 35.5, 168.2, 3.9, and 45.4, respectively. The average execution times of the algorithms were 91.652, 31.594, 58.649, 13.251 and 34.089 seconds for AVAC, AC, AMC, BGA, and SMLS,

respectively. Although the execution time of AVAC is reasonably longer than the others, the number of transition states uncovered by AVAC per second, which is 3.87, is the highest among all algorithms.

Next, we also study the efficacy of our fitness function proposed in Eqn. (2) against that in Eqn. (7), introduced previously in [1].

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \varepsilon \left[\left(\frac{r_0}{r_{ij}} \right)^{12} - 2 \left(\frac{r_0}{r_{ij}} \right)^{6} \right] \qquad (7)$$

Maintaining the same computational setting for AVAC, the percentages of true and false 1st saddle points uncovered and execution time are reported in Table 1. Note that the false 1st order saddle points donate non 1st order saddle points, critical points.

The results, summarized in Table 1, show that the fitness function proposed in Eqn. (2) maintained the highest percentages of true 1st order saddle points and 0% false 1st order saddle points, while the fitness function proposed in [1] Eqn. (7) lead to 36-39% false saddle points. Extra computation is, thus, unnecessary incurred to distinguish between true and false saddle points. Hence, the proposed fitness function lead to a ~1.45 factor of improvements in the execution time over that proposed in [1].

TABLE 1. FITNESS FUNCTIONS PERFORMANCE COMPARISON, ON THE THREE BENCHMARK PROBLEMS.

Benchmark Problem	Fitness functions	Total uncovered Solutions (%)	True Saddle Points (%)	False Saddle Points (%)	Execution Time (Seconds)
Multi-Function	**Proposed Eqn. (2)**	**94%**	**94%**	**0**	**52.094**
	Eqn. (7) [1]	125%	87%	38%	75.890
Rastrigin	**Proposed Eqn. (2)**	**96%**	**96%**	**0**	**65.251**
	Eqn. (7) [1]	133%	94%	39%	102.351
Sines	**Proposed Eqn. (2)**	**100%**	**100%**	**0**	**56.775**
	Eqn. (7) [1]	136%	100%	36%	81.13

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a novel genetic algorithm for finding multiple first order saddle points. The algorithm includes a new fitness function, and valley adaptive clearing scheme which involves hill-valley initialization, valley-adaptive clearing and archiving. The experimental results showed that the archiving clearing GAs located much more 1st order saddle points than previously proposed genetic and classic algorithms.

Among the proposed archiving clearing GAs, numerical results showed that the proposed valley-adaptive clearing GA maintained the highest percentages of uncovered 1st order saddle points on all benchmark problems.

We have also introduced a new formulation of fitness function for locating 1st order saddle points. Investigation against the fitness function used in [1] highlighted the efficacy of the new fitness function in converging to true 1st order saddle points.

For future work, the authors are going to study more complex Lennard Jones clusters using a memetic algorithm version of the proposed method. As memetic algorithms combine the evolutionary algorithm with individual learning procedures capable of performing local refinements, they efficiently explore and exploit the search landscape much better than GA. They have also been successfully applied to many complex optimization problems across a wide variety of application domains [11-14] including water cluster optimization [25].

VII. ACKNOWLEDGEMENTS

The authors acknowledge the contributions of Intellisys, a centre for research collaboration between NTU and ST Engineering.

VIII. REFERENCES

[1] S.D. Bungay, R.A. Poirier, and R.J. Charron, "Optimization of transition state structures using genetic algorithms", Journal of Mathematical Chemistry, 2004. 28(4): p. 389-401.

[2] M. M. H. Ellabaan, Y. S. Ong, "Valley-Adaptive Clearing Scheme for Multimodal Optimization Evolutionary Search", The International Conference on Intelligent Systems Design and Applications, ISDA09, 30th Nov – 2nd Dec. Pisa, Italy

[3] L. Lucchese and S.K. Mitra. Using saddle points for subpixel feature detection in camera calibration targets. In Proceedings of the 2002, Asia Pacific Conference on Circuits and Systems, volume 2, pages 191– 195, 2002.

[4] C. Sminchisescu, B. Triggs, Building roadmaps of minima and transitions in visual models, International Journal of Computer Vision 61 (1), pp. 81-101, 2005.

[5] K. K. Lim, Y. S. Ong, M. H. Lim, X. Chen and A. Agarwal, 'Hybrid Ant Colony Algorithm for Path Planning in Sparse Graphs', Soft Computing Journal, pp. 981-994, 2007

[6] M. Lassonde, "Approximation, optimization and mathematical economics" Springer, 2001.

[7] G. M. Keserü, I. Kolossváry "Molecular mechanics and conformational analysis in drug design", Oxford; Malden, Mass.: Blackwell Science, 1999.

[8] D.S. Jones, B.D. Sleeman, "Differential Equations and Mathematical Biology" Chapman & Hall/CRC; 2003.

[9] Y. S. Ong, P.B. Nair and A.J. Keane, "Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling", American Institute of Aeronautics and Astronautics Journal, 2003, Vol. 41, No. 4, pp. 687-696.

[10] Y. S. Ong, P. B. Nair and K. Y. Lum, “Max-Min Surrogate-Assisted Evolutionary Algorithm for Robust *Design”*, IEEE Transactions on Evolutionary Computation, Vol. 10, No. 4, pp. 392-404, August 2006.

[11] F. Neri, J. Toivanen, G. L. Cascella and Y. S. Ong, "An Adaptive Multimeme Algorithm for Designing HIV Multidrug Therapies”, IEEE/ACM Transactions on Computational Biology and Bioinformatics, Special Issue on Computational Intelligence Approaches in Computational Biology and Bioinformatics, 2007, Vol. 4, No. 2, pp. 264-278.

[12] J. Tang, M. H. Lim and Y. S. Ong, "Diversity-Adaptive Parallel Memetic Algorithm for Solving Large Scale Combinatorial Optimization Problems", Soft Computing Journal, 2007, Vol. 11, No. 9, pp. 873-888.

[13] Q. C. Nguyen, Y. S. Ong, H. Soh and Jer-Lai Kuo, "Multiscale Approach to Explore the Potential Energy Surface of Water Clusters (H2O)8 n<=8", Journal of Phys. Chem. A, Vol. 112, No. 28, pp. 6257 - 6261, 2008.

[14] Z. Z. Zhou, Y. S. Ong, M. H. Lim and B. S. Lee, "Memetic Algorithm using Multi-Surrogates for Computationally Expensive Optimization Problems", Soft Computing Journal, Vol. 11, No. 10, pp. 957- 971, August 2007.

[15] R. Judson, Genetic algorithms and their use in chemistry, Reviews in Computational Chemistry 10 (1997) 1–73.

[16] D. Lim, Y. S. Ong, Y. Jin, B. Sendhoff and B. S. Lee, “Efficient Hierarchical Parallel Genetic Algorithms Using Grid Computing”, Future Generation Computer Systems: The International Journal of Grid Computing: Theory, Methods and Applications, 2007, Vol. 23 , No. 4 , pp. 658-670.

[17] S. Trygubenko and D. Wales, “A Doubly Nudged Elastic Band Method for Finding Transition States”. Chem. Phy., 2004. 120(5): p. 2082-2094.

[18] J.M.d. Campo, and A.M. Koster, “A hierarchical transition state search algorithm”, Chem. Phy., 2008. 129: p. 024107-12.

[19] REDDY, C.K. and H.-D. CHIANG, “A Stability Boundary Based Method for Finding Saddle Points on Potential Energy Surfaces”. JOURNAL OF COMPUTATIONAL BIOLOGY, 2006. 13: p. 745–766.

[20] G. Henkelman and H. Jonsson, “A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives”. Chem. Phy., 1999. 15(111): p. 7010-7022.

[21] R.A. Olsen, “Comparison of methods for finding saddle points without knowledge of the final states”. Chem. Phy., 2004. 121(20): p. 9776-9792.

[22] P. Chaudhury, and S.P. Bhattacharyya, “A simulated annealing based technique for locating first-order saddle points on multidimensional surface and constructing reaction paths: several model studies”. Theo. Chem., 1998. 429: p. 175-186.

[23] P. Chaudhury, S.P. Bhattacharyya, and W. Quapp, “A genetic algorithm based technique for locating first-order saddle point using only a gradient dominated recipe”. Chemical Physics 2000. 253: p. 295-303.

[24] J. Branke, “Evolutionary Optimization in Dynamic Environments”. 2002: Kluwer Academic Publishers.

[25] H. Soh, Y. S. Ong, Q. C. Nguyen, Q. H. Nguyen, M. S. Habibullah, T. Hung and J.-L. Kuo, "Discovering Unique, Low-Energy Pure Water Isomers: Memetic Exploration, Optimization and Landscape Analysis", IEEE Transactions on Evolutionary Computation, Accepted August 2009.