MULTIPLE PHASE TRANSITION PATH AND SADDLE POINT SEARCH IN COMPUTER AIDED NANO DESIGN

A PhD Thesis
Presented to
The Academic Faculty

by

Lijuan He

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology
August 2015

MULTIPLE PHASE TRANSITION PATH AND SADDLE POINT SEARCH IN COMPUTER AIDED NANO DESIGN

Approved by:

Dr. Yan Wang, Chair
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Graeme Henkelman
Department of Chemistry and
Biochemistry
University of Texas at Austin

Dr. Seung Soon Jang
School of Materials Science and
Engineering
Georgia Institute of Technology

Dr. David L. McDowell
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Dr. Ting Zhu
George W. Woodruff School of
Mechanical Engineering
Georgia Institute of Technology

Date Approved: June 15th, 2015

ACKNOWLEDGEMENTS

I would like to express my deep appreciation and gratitude to my advisor, Dr. Yan Wang, for the patient guidance and mentorship as well as financial support in the past five years. My research is always inspired by talking with him. It is his encouragement and feedback in my efforts that make this dissertation possible. He provides me not only the training to be a good researcher, but also the guidance on choosing my career path by sharing his own experiences and thoughts.

I would also like to thank my committee members, Dr. Graeme Henkelman, Dr. Seung Soon Jang, Dr. David L. McDowell and Dr. Ting Zhu for providing their insightful feedbacks on my PhD proposal which spawned some ideas in this dissertation. Dr. Henkelman provided me valuable suggestions on the potential uncertainties which should be addressed for my third topic as well as pointed out the potential sampling problems in the second algorithm. Dr. Jang's suggestion on the collective potential model inspired me to explore different models to serve our purpose. Dr. McDowell's insight into the uncertainties due to the dynamical interaction among atoms on the fly during the search process gave me a new angle on the uncertainties I need to address. His valuable suggestion on considering physical meanings when construct the collective potential model makes a successful collective model in this dissertation. Dr. Ting Zhu provided valuable perspective on how to demonstrate the algorithms in a systematic way. He also provided me precious guidance on the physical material examples when I had problems on those examples.

I would like to acknowledge the support of the National Science Foundation grant CMMI-1001040 and 1306996, which provided me with the financial support for the research done in this dissertation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Functional materials with controllable phase transitions have been widely used in devices for information storage (e.g. hard-disk, CD-ROM, memory) and energy storage (e.g. battery, shape memory alloy). One of the important issues to design such materials is to realize the desirable phase transition processes, in which atomistic simulation can be used for the prediction of materials properties. The accuracy of the prediction is largely dependent on searching the true value of the transition rate, which is determined by the minimum energy barrier between stable states, i.e. the saddle point on a potential energy surface (PES). Although a number of methods that search for saddle points on a PES have been developed, they intend to locate only one saddle point with the maximum energy along the transition path at a time. In addition, they do not consider the input uncertainty associated with the calculation of potential energy. To overcome the limitations, in this dissertation, new saddle point search methods are developed to provide a global view of energy landscape with improved efficiency and robustness. First, a concurrent search algorithm for multiple phase transition pathways is developed. The algorithm is able to search multiple local minima and saddle points simultaneously without prior knowledge of initial and final stable configurations. A new representation of transition paths based on parametric Bézier curves is introduced. A curve subdivision scheme is developed to dynamically locate all the intermediate local minima and saddle points along the transition path. Second, a curve swarm search algorithm is developed to exhaustively locate the local minima and saddle points within a region concurrently. The algorithm is based on the flocking of multiple groups of curves. A collective potential model is built to simulate the communication activities among curves. Third, a hybrid saddle-point search method using stochastic kriging models is developed to improve the efficiency of the search algorithm as well as to incorporate model-form uncertainty and numerical errors associated with density functional theory calculation. These algorithms are demonstrated by predicting the hydrogen diffusion process in FeTiH and body-centered iron Fe_8H systems.

CHAPTER I
INTROUCTION

To accelerate the development of new materials, we should adopt a "solution-oriented" approach to create new materials by systematic design. In the traditional "discovery-based" approach, in which new materials are discovered first and then their potential applications are explored, heavily relies on trial-and-error laboratory experiments to turn them into commercial products for potential applications. The process is time-consuming and not cost-effective for developing new materials that satisfy increasing demand for them. Materials by design [1, 2] within the scope of the Materials Genome Initiative (MGI) [3] will be the focus of materials science and technology study in the next several decades. A materials innovation infrastructure that integrates computational tools, experimental tools, and digital data is the focus of MGI, as illustrated in Figure 1. Creating new materials by systematic design based on a problem-solving paradigm will be a major mission for engineers in the future. The availability of computational design tools is the key to improve the efficiency of the materials design process. The research presented in this thesis is to provide such kind of design tools, specifically the ones for simulation-based phase change materials design. In the rest of this chapter, the motivation of our research and the problem we try to solve are first described. A general description of the proposed method to solve the problem is given, followed by a summary of the contributions of this dissertation.

1.1 Motivation

Nanotechnology of today mainly focuses on the discovery of nanoscale materials with new properties and applications. So far it heavily relies on trial-and-error laboratory experiments to turn them into commercial products for potential applications. This discovery-based development approach is a time-consuming process and will not be able to satisfy the increasing demand on new materials in the future. A solution-oriented approach should be adopted instead. That is, engineers start with the analysis of the existing needs in solving engineering problems, then design and produce new materials that can fulfill those needs in order to solve the specific problems. Engineers need enabling technologies to create novel materials by designing the microstructure of the materials systematically through a bottom-up approach. In order to achieve this goal, computational design tools are essential. For example, modeling and simulation software allows engineers to accurately predict the properties and functions of different material structures to improve the efficiency of the materials design process.

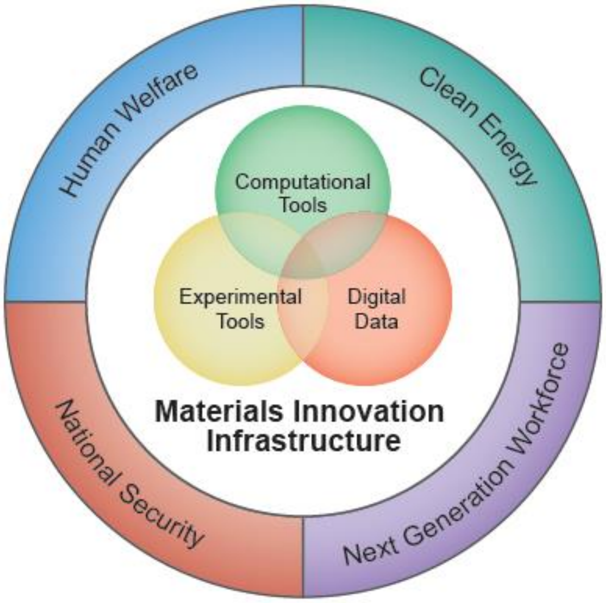


Figure 1: Materials innovation infrastructure in Materials Genome Initiative [3]

One category of the most important and widely-used materials in engineering are phase-change materials (PCMs). Each phase in materials has different physical properties such as thermal conductivity, electrical conductivity, and optical reflectivity. PCMs realize unique functionalities of products based upon the change of phases, which have attracted considerable attentions in industries [4, 5], including energy conservation (e.g. calcium chloride hexahydrate [6-8] and polyethylene glycol [9, 10] in construction), data storage (e.g. doped Sb_2Te alloys in information and electronics [11, 12]), energy storage (e.g. paraffin waxes for battery in automobile [13]), logistics (e.g. nano-structured calcium silicate for food package [14]), bioengineering (e.g. shape memory alloys for implants [15]), and apparel (e.g. encapsulated paraffin wax for adjustable temperature in cloth [16, 17]), as illustrated in Figure 2.

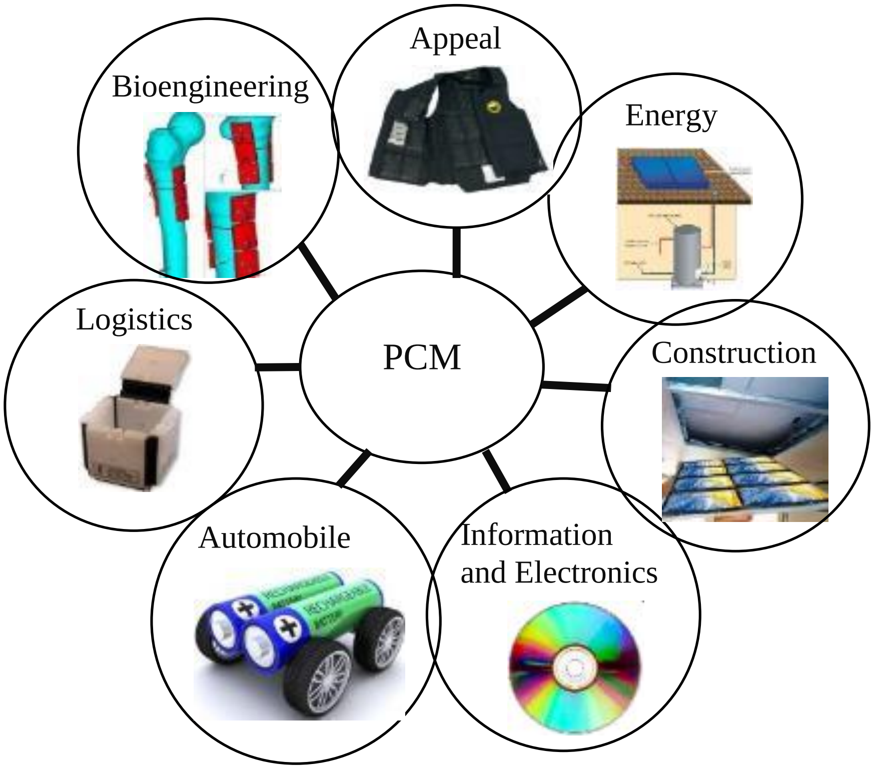


Figure 2: Various applications of PCMs

One of the most important design issues for PCMs is to engineer the phase transition processes. Simulation that predicts transition processes efficiently and accurately under desirable conditions is critical in the design of PCMs. The goal of designing PCMs is to find desirable stable phases and transition conditions to meet the performance requirements. For example, for PCMs in data storage that requires high-speed write and erase and small energy consumption, the design for such materials is to identify and realize such structures using structure-property relationships. Simulation can be used in the iterative design process.

In thermodynamics, a phase transition is described from a top-down viewpoint as the transformation of a thermodynamic system from one phase to another [18, 19]. A phase is one of the states of matter that have uniform physical properties, and the materials system has a particular level of free energy. When external conditions such as temperature or pressure vary, one or more properties of the material change and the phase transition occurs. As a result of these external influences, the system shifts from one free energy level to another. External conditions and the amount of required input energy, which can be quantitatively estimated, define a phase transition. From the atomistic point of view, a phase transition is a geometric and topological transformation process of materials from one phase to another, each with a unique and homogeneous physical property. This phenomenon is observed in many processes. For example, the diffusion of adatoms on a solid as a fundamental problem in surface science can be regarded as a phase transition process. Chemical reactions and protein folding can also be generally treated as phase transitions. Using the description at atomic level, we are interested in structural and topological changes at the atomistic scales. Understanding and controlling the transition at the microscopic level can provide the fundamental understanding for the purpose of design.

Simulating phase transitions can provide the insights of the detailed physical and chemical processes at the atomistic scales. In simulation-based materials design, the challenge of accurately predicting a phase transition is the knowledge of the transition rate. Within the harmonic transition state theory (hTST) [20], the transition rates that characterize the probability of an event occurring is determined by the energy difference between the initial state and the corresponding transition state. Mathematically, the energy landscape is characterized by a potential energy surface (PES). The energy change during the transition process is determined by the initial state (a local minimum on the PES) and the transition state (a saddle point on the PES). The general process to simulate the phase transition process at atomistic scales is as follows. First, a PES is generated. Second, a minimum energy path (MEP) which is the most probable physical pathway of transition among all possible ones is located. Third, the activation energy is obtained by finding the maximum energy on the MEP and the transition rate is calculated using hTST.

The key of phase transition simulation is searching the saddle points on the PES. Numerous efficient optimization algorithms are available to accurately locate the local minima on a PES. However, the determination of saddle points on a PES is still a challenging problem. First, a PES with high dimensionality is usually not explicitly known. In first principles approaches, the interaction between atoms can be obtained by solving the Schrödinger equation using *ab initio* calculation. However, an analytical description of the high dimensional PES is still infeasible. Second, a good approximation of reaction coordinate is usually unavailable. On a PES, a saddle point is a stationary point with zero forces in all directions. The Hessian matrix of a saddle point has only one negative eigenvalue. The corresponding eigenvector defines the reaction coordinate. Hence, saddle point search usually involves maximization of the energy in one direction and minimization in all other directions. The challenge is to find the reaction coordinate which is unknown in most cases.

To search saddle points on a PES, a number of algorithms [21-29] are proposed. These methods can be generally categorized into two groups: single-ended methods and double-ended methods, which are reviewed in Section 2.1. The single-ended search methods usually identify the saddle point without locating the transition path that connects the identified saddle point and corresponding local minima. The methods usually identify one saddle point at a time. In addition, the search in these methods is likely to be trapped at the intermediate saddle point without the highest energy if the transition consists multiple phases. For the double-ended search methods, they usually require the knowledge of initial and final states to locate the saddle point. In addition, they usually locate some but not all the intermediate saddle points and local minima between two stable states.

In addition to the development of search algorithms, another key issue in phase transition simulation is the quantification of uncertainties during the simulation. For a phase transition simulation, the inaccuracy of the simulation results arises from two basic resources. First, hTST is a source of uncertainty. The calculation of transition rates based on hTST provides an upper bound of the transition rates. The theory has three assumptions: 1) the Born-Oppenheimer approximation is reasonable, 2) a Boltzmann distribution is accurate enough to describe the rate, and 3) an $n-1$dimensional dividing surface exists between the initial state and final state, and the transition trajectory only crosses the dividing surface once. Second, the local minima and saddle point search algorithms also introduce uncertainties. Since all the algorithms are conducted on the computer, numerical errors are inevitable. The PES is another source of uncertainty. A PES is constructed by fitting or interpolating available energy data points either from experiments or *ab initio* calculations. Since fitting and interpolation methods employ limited number of data points to build the whole surface, error is unavoidable. The details of uncertainties are reviewed in Section 2.4.

The objective of this doctoral dissertation is to develop three new saddle point search methods to provide a global view of energy landscape efficiently and robust estimation of activation energy. Specifically, a *concurrent search algorithm* for multiple phase transition pathways is developed to locate all saddle points and local minima on a multiple transition pathway. As an extension to this algorithm, a *curve swarm search* algorithm is developed to further improve the efficiency of energy landscape survey by a swarm of curves. The curve swarm search algorithm will exhaustively locate the local minima and saddle points within the searching area on a PES. In addition, an efficient and robust saddle point search method using stochastic kriging metamodels is developed to improve the efficiency of the searching process and to incorporate the model-form uncertainty and numerical errors associated to DFT calculation and search algorithms on-the-fly during the searching process, which is to improve the robustness of activation energy estimation. The detailed description of the three algorithms is presented in CHAPTER III, CHAPTER IV and CHAPTER V.

1.2 An Overview of the Developed Algorithms

This section gives a general description for the concurrent search algorithm, the curve swarm search algorithm, and the efficient and robust saddle point search algorithm.

1.2.1 Concurrent Search Algorithm

The concurrent search algorithm is to search the MEP between two stable states. The new algorithm is able to locate both the saddle point and local minima simultaneously. Therefore no prior knowledge of the precise positions for the reactant and product on the PES is needed. Unlike existing pathway search methods, the algorithm is able to locate multiple local minima and saddle points along a multiple phase transition pathway, which improves the accuracy of the activation energy estimation.

Here the Bézier curve is used to represent the transition pathway. A Bézier curve is a parametric curve defined by control points which can be used to manipulate the shape of the curve. In the algorithm, each control point of the curve represents one state, also called image, on the transition path. During the searching process, reactant and product states are located by minimizing the two end control points of the Bézier curve using the conjugate gradient method, while the shape of the transition pathway is refined by moving the intermediate control points of the curve in the conjugate directions. In each iteration, a set of conjugate directions are determined and then several line minimization steps along those conjugate directions are applied to each intermediate control points. As a result, the Bézier curve will gradually converge to the MEP.

To keep the control points evenly distributed, we developed a constrained and local Bézier curve degree elevation and reduction scheme to redistribute the control points in each iteration. Since there could be more than one saddle point with extra local minima between the two stable states, one curve could be broken into two to represent two stages of transitions recursively. We also developed a curve subdivision scheme to check whether there is more than one saddle point with extra local minima between the two end points of the optimized curve. If there is, we break this curve into two which are regarded as the initial guess of the transition path for the two stages of transitions. Then those two curves are optimized by the same procedure as we did to the original curve. This ‘check-and-break’ process continues until each of the curves only passes through two adjacent local minima with their end points located at those local minima. Since in real applications, people are more concerned about the exact positions of the saddle points instead of the MEP in most cases, a climbing up scheme is also introduced to locate the saddle points on those curves that are close enough to the MEP.

1.2.2 Curve Swarm Search Algorithm

The curve swarm search algorithm is to exhaustively locate the local minima and saddle points on a PES within a searching area. Thus it provides a global view of the energy landscape. Unlike the concurrent search algorithm, the curve swarm search algorithm uses multiple groups of curves to locate multiple multi-stage transition paths within a search area, instead of one multi-stage transition path.

The algorithm represents a transition path by a parametric Bézier curve with control points. Instead of using one initial curve as in concurrent search algorithm, the curve swarm search algorithm uses a number of initial curves to thoroughly explore a PES. The method uses the concept of flocking (i.e. swarming or schooling of multi-agent systems) that describes the collective behavior of many interacting particles with a common objective [30, 31]. The algorithm treats all newly created curves (called descendants of one initial curve) as one group, with their end points connected together. For each curve in one group, its two end control points are minimized to locate two local minima, while intermediate control points is updated to refine the shape of the curve. If extra minima are found along the curve, it is broken into multiple curve segments to form a multi-stage transition path. During the searching process, each group will communicate with their neighbors to avoid getting too close or far away.

A major issue in the curve swarm search algorithm is to maintain cohesion and to avoid collision with other groups at the same time during the searching process. Cohesion means that particles should stay relatively close to each other to explore a PES thoroughly, thus, locate all the local minima and saddle points. Collision means that more than one groups search the same area on a PES. The algorithm should avoid collision to prevent repetitive exploration, thus, maintains global search and reduces computational cost.

To maintain cohesion and to avoid collision, we introduce a collective potential model which is a function of distance between two curves to simulation the forces among curves. During the searching process, the algorithm will calculate the collective force of one curve from its neighbors based on the collective potential model. The collective force is then applied to each intermediate control point during searching process. Thus, each intermediate control point is driven by a weighted sum of the collective force and a parallel component of the true potential force along the conjugate direction. The collective force is defined as a weighted sum of the pairwise forces between the curve and its neighbor curves within an area.

1.2.3 Efficient and Robust Saddle Point Search Algorithm

The efficient and robust saddle point search algorithm is to improve the efficiency of the saddle point searching using surrogate model and to incorporate the PES model error on the fly during the searching process, which is to improve the robustness of activation energy estimation. In the existing first-principle saddle point search methods, the requirement of a large number of expensive evaluations of potential energy, e.g. using density functional theory (DFT), limits the application of such algorithms to large systems. Thus, it is meaningful to minimize the number of functional evaluations as DFT simulations during the search process. Furthermore, model-form uncertainty and numerical errors are inherent in DFT and search algorithms. Robustness of the search results should be considered. The efficient and robust saddle point search algorithm aims to handle this two aspects using kriging models with hidden Gaussian process. Different from existing searching methods, the algorithm keeps a memory of searching history by constructing surrogate models and uses the search results on the surrogate models to provide the guidance of future search on the PES. The surrogate model is also updated with more DFT simulation results.

Kriging surrogate model is one of the interpolated surface to approximate the true surface. Thus, model-form uncertainties associate to the surrogate model is inevitable. Besides, the input data which are obtained from the DFT calculation for the kriging model also involves uncertainties. To capture this two types of errors, we introduce the kriging model of hidden Gaussian process which uses two Gaussian processes to represent the model-form uncertainties associated with kriging method and to represent the uncertainties associated with the input data from DFT calculation, separately. This idea is also implemented in the stochastic kriging method.

As we know, every update of searching steps in the existing saddle point search methods requires a function evaluation based on DFT calculation at the current position which is computational expensive. The efficient and robust algorithm starts searching local minima and saddle points by evaluating each function value using DFT calculation and records the information (i.e. position and the corresponding value of total energy and gradient) at each updated position. After a predefined number of iterations, the algorithm constructs a surrogate model which represents the true PES using stochastic kriging method based on the data points collected in the previous searching iterations. Then all the function evaluations during the searching process are conducted using the surrogate model which is much cheaper in terms of computation cost. Since the surrogate model involves uncertainties, the algorithm needs more data to update the surrogate model to better approximate the true PES. Thus, after a predefined number of searching iteration on the surrogate model, the function evaluation is conducted again using DFT calculation. This ‘real-surrogate-real’ model continues until the saddle point position converges.

1.3 Dissertation Contributions

In this research, we develop three new saddle point search methods to provide a global view of energy landscape efficiently and robust estimation of activation energy. In particular, we develop a concurrent search algorithm, a curve swarm search algorithm, and an efficient and robust saddle point search algorithm. The novel contributions of this research are summarized as

- A new method is developed to search multiple local minima and saddle point concurrently along a path with multiple transition stages. A new approach based on parametric Bézier curves to represent transition paths is proposed, where new concepts of degree elevation and reduction are used to maintain the even distribution of images along the path. In addition, a curve subdivision scheme is developed to break one transition path into two stages of transition paths, which allows the algorithm to dynamically adjust the number of total images along one transition path one the fly during the searching process.
- A new method that uses groups of curves to exhaustively explore a PES is developed. A new communication scheme among groups of curves is developed to maintain a balance between local and global exploration of a PES. Specifically, a scheme of determining the parameters of the potential model is developed to incorporate the information of the potential energy surface on the fly during the searching process. Also, a scheme is developed to determine how to apply the collective force to the curves to maintain cohesion and avoid collision.
- A new method is developed to improve the efficiency of the searching process and to incorporate the model-form error in DFT calculation on the fly during the searching process, which provides a robust estimation of activation energy. A surrogate model is introduced to incorporate the error from both the surrogate model itself and the input data from the DFT calculation. A new scheme is developed to determine the searching region for particular local minimum.

The rest of the dissertation is organized as follows. CHAPTER II provides some background knowledge related to the topics of transition pathway search methods, saddle point search methods, conjugate gradient methods and their convergence analysis, the Bézier curve degree reduction methods, PES fitting methods and associated uncertainties, and kriging methods. CHAPTER III, CHAPTER IV and CHAPTER V give a detailed description and demonstration of the concurrent search algorithm, curve swarm search algorithm, and efficient and robust saddle point search algorithm respectively. CHAPTER VI concludes with a brief summary of the accomplishment and discussions about the future extension.

CHAPTER II
BACKGROUND

This chapter provides the basic background information associated with activation energy search including saddle point search methods and uncertainties in PES model, conjugate gradient methods and associated convergence analysis, degree elevation and reduction methods of the Bézier curve, and kriging methods. A phase transition is a rare event characterized by occasional transitions from one atomic configuration referred to initial state to another referred to final state. To simulate these rare events, classical molecular dynamics method that is based on Newton's second law is not efficient because the transitions of interest are many orders of magnitude slower than the vibration of atoms. To solve this time-scale problem, other simulation approaches, for example kinetic Monte Carlo (kMC) method, are developed and widely used to simulate the rare events. One of the key inputs for such simulation approaches is the transition rates for all possible rare events. The transition rates for each rare events is determined by the activation energy associated with that particular event. Thus, the accuracy of the simulation results depends on the accuracy of the activation energy.

2.1 Existing Saddle Point Search Methods

The challenge for phase transition simulation is to search the transition rate which is determined by the activation energy between two states. An activation energy barrier always exits between two states. In 1931, Erying and Polanyi [32, 33] proposed the transition state theory (TST) as a means to calculate the transition rates using the activation energy to characterize reactions. Most of the simulation methods developed recently are based on the TST and harmonic transition state theory (hTST) [34]. Some variants of TST (Variational Transition State Theory [35] and Reaction Path Hamiltonian [36]) are also used. The general procedure to simulate a phase transition process is as follows. First a PES is generated. Then a MEP which is the most probable physical pathway of transition among all possible ones is located. Finally the activation energy is obtained by finding the maximum energy on the MEP and the transition rate is calculated using TST. Subsequently the phase transition simulation can be done using KMC, Accelerated Molecular Dynamics (AMD) [37] or other simulation methods. The accuracy of the simulation depends on the accuracy of the rate constants. In other words, it depends on the accuracy of the activation energy. The research on transition pathway search and saddle point search aims to find the accurate MEP and the saddle point.

To search saddle points on a PES, a number of algorithms [21-29] are developed in the past few decades. The algorithms can generally be categorized into two groups: single-ended methods and double-ended methods. Single-ended methods start with one configuration to search the saddle point on a PES without locating the corresponding MEP. Double-ended methods are to locate the saddle point and corresponding MEP between two starting states.

The major group of single-ended methods is eigenvector-following methods [38-47] that follow the eigenvector of Hessian matrix with local quadratic approximations of the PES. Newton-Raphson method [38] optimizes the energy iteratively along the eigenvector directions using Newton-Raphson optimizer. Surface walking algorithms [39-44] determine the walking steps by introducing lagrange multipliers [39], a trust radius [40, 42], and parameterized step vector [43, 44]. Partitioned rational function optimization method [41] makes a local rational function approximation to the PES with augmented Hessian matrix. Frontier mode-following method [45] identifies the uphill direction to the saddle point by following more than one eigenvectors. Hybrid eigenvector-following methods [46, 47] update the eigenvectors of the Hessian matrix using a stand shifting technique and minimize the energy in tangent space using conjugate gradient minimization.

Other single-ended methods are presented as follows. Distinguished coordinate method [48-50] defines one internal coordinate as the reaction coordinate and iteratively minimizes the energy level along all other internal coordinates within a gradually converging lower and upper bounds of the reaction coordinate. Gradient norm minimization method [51] identifies the transition states by minimizing the Euclidian norm of the gradient of the potential energy function using a generalized least-squares technique. Constrained uphill walk method [52, 53] employs simplex method to minimize the energy over a set of hypersphere surfaces, the center of which are several points on or near the reaction path. Image function methods [54-56] formulate an image function with the local minima located at position of the saddle points on the original PES. Gradient-only method [57] starts at the bottom of a valley and traces to a saddle point in the ascent direction that is defined based on gradient information. Activation-Relaxation technique method [58] can travel between many saddle points using a two-step process; an image first jumps from a local minimum to a saddle point, and then back down to another minimum. Reduced gradient following [59, 60] and reduced potential energy surface model [61] methods use intersections of zero-gradient curves and surfaces, with saddle point search occurring within the subspace of these curves or surfaces. Interval Newton's method [62] is capable of finding all stationary points by solving the equation of vanishing gradient.

The most popular double-ended methods are chain-of-states methods [63-85] including nudged elastic method (NEB) [73-76, 80, 83-85], string methods [77-79, 81, 82], and other methods [63-72]. Chain-of-states methods rely on a collection of images that represent intermediate states of the atomic structure as it transforms from initial to final configurations along the transition path. These discrete states are chained to each other after the search converges, and the transition path and saddle point are obtained. The most common one among these methods is the NEB [73], which relies on a series of images connected by springs. To increase the resolution at the region of interest (ROI) and the accuracy of saddle point energy estimates, the NEB method omits the perpendicular component of the spring force, as well as the parallel component of the true force due to the gradient of the potential energy. In some cases, this method produces paths with unwanted kinks, or may not have any images that are directly on the saddle point. The improved tangent NEB [74] and doubly nudged elastic band [80] methods reduce the appearance of kinks by generating a better estimate of the tangent direction of the path and re-introducing a perpendicular spring force component. The adaptive nudged elastic band [76] method increase the resolution around the saddle point adaptively to improve the efficiency and accuracy of NEB. The free-end NEB method [83] only requires knowledge of either the initial or final state, rather than both. The climbing image NEB [75] allows the image with the highest energy to climb in order to locate the saddle point. A generalized solid-state NEB (G-SSNEB) [85] was developed with a modified approach to estimate spring forces and perpendicular projection to search MEP for the process of solid-solid transformations which have both atomic and unit-cell degrees of freedom involved. Eigenvector following optimization can be applied to the result of NEB to locate actual saddle points, and the resolution of ROI can be increased by using adaptive spring constants [84].

String methods [77, 78] represent the transition path continuously as Splines that evolve and converge to the MEP. As opposed to NEB, the number of points used in the String method can be modified dynamically. The growing string [79] takes advantage of this by starting with points at the reactant and product, and then adding points which meet at the saddle point. The quadratic string method [81] is a variation that uses a multi-objective optimization approach. The string and NEB methods are comparable in terms of computational efficiency [86]. In addition to NEB and string methods, other chain-of-states methods have been developed to find transition paths. Gaussian chain method [64] minimizes the average value of the potential energy along the path by formulating an objective function. Self-penalty walk (SPW) method [66] reformulates the Gaussian chain objective function by adding a penalty function of distance. Locally updated planes method [67, 68] minimizes the energy of each image in all directions expect their corresponding tangent directions estimated at the position of each images along the path. Variational Verlet method [69] formulates an objective function similar to the Gaussian chain objective function to locate dynamical paths. Sevick-Bell-Theodorou method [70] optimizes the energy with a constraint of fixed distance between images. Path energy minimization method [71] modifies the SPW objective function by including a high value of exponential index that increase the weight of the image with highest energy along the path.

In addition to chain-of-states methods, other double-ended methods have been developed to search transition paths. Ridge [87] and dimer [88, 89] methods use a pair of images to search for the saddle point. Dewar-Healy-Stewart method [90] searches the saddle point by iteratively reducing the distance between reactant and product images.

Step and slide method [91] uses an image from the initial and final states. Energy levels of each are increased gradually, and the distance between them is minimized while remaining on the same isoenergy surface. Concerted variational strategy method [92] employs Maupertuis' and Hamilton's Principles to obtain a transition path that will be further refined using conjugate residual method [93] to locate the saddle point. Conjugate gradient methods [94-96] are based on conjugate gradient function minimization. Conjugate peak refinement method [97] finds saddle points and the MEP by searching the maximum of one direction and the minima of all other conjugate directions iteratively. Accelerated Langevin dynamics method [98] is a stochastic transition path sampling method that involves activation and deactivation paths described by Langevin equations. Hamilton-Jacobi method [99] relies on the solution of a Hamilton-Jacobi type equation to generate the MEP. Missing connection method [100] identifies multistep paths between two local minima by connecting one minimum to its adjacent minimum one by one. Each intervening transition states is located by using the doubly NEB methods. For each cycle, the next targeted minimum that will be connected is determined by the criteria of minimum distance in Euclideam distance space using Dijkstra algorithm. Synchronous transit method [21, 95, 101] estimates the transition states and refines the saddle point by combining the conjugate gradient method and the quasi-Newton minimization method. Intersection surface method [102] formulates an intersecting surface model based on the quadratic basin approximations around the two local minima on the original PES. The minima of the intersecting surface are good approximation to the positions of saddle points on the original PES. Contingency curve method [103] formulates a bi-Gaussian model for the PES. The transition path is represented by using the equipotential contour contingency curve that connects two local minima and the corresponding saddle points.

2.2 Conjugate Gradient Method

Conjugate gradient method is an effective iterative method for solving large, sparse systems of linear equations numerically, provided that the coefficient matrix of which is symmetric and positive definite. This method is originally proposed by Hestenes and Stiefel [104]. In solving the linear equations $A\mathbf{x}=\mathbf{b}$, the key idea of this method is to minimize the residual $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$ along conjugate directions. This method solves the equation in at most n steps provided that there is no round-off error. A detailed explanation and development of this method could be found in [105, 106]. Later on, this method is extended to solve nonlinear systems of equations and unconstrained optimization problems such as potential energy minimization. In [107], Fletcher and Reeves developed an algorithm namely Fletcher-Reeves method by using the conjugate gradient method to minimize a general function. For a quadratic function, the method could locate the minimum in at most n steps apart from rounding off errors. For non-quadratic function, it usually takes more than n iterations to locate the minimum. Later, Polak and Ribière [108] modified the Fletcher-Reeves method by changing the way how the conjugate directions are calculated. Hestenes and Stiefel [104] also have their own approach to calculate the conjugate directions.

Same as other iterative methods used in minimization problem, convergence is always the key issue. When the conjugate gradient method is applied to a general function, a local quadratic approximation is always involved. In order to determine the step size in each conjugate direction for a general function, a line minimization search process is adopted, which could affect the convergence properties of the method depending on which convergence criteria is used for the line searches. When the method is used to solve unconstrained optimization problems, extensive research has been conducted on the convergence properties of this method. Zoutendijk [109] proved that the Fletcher-Reeves method with exact line searches is globally convergent on general functions. Al-Baali [110] extended Zoutendijk's conclusion to inexact line searches, particular for a variation of the Wolfe line-search. Powell [111] demonstrated that the Polak- Ribière and Hestenes-Stiefel methods are not globally convergent even with exact line searches. Meanwhile, Hu and Storey [112], and Gilbert and Nocedal [113] studied the global convergence of the algorithm related to the Fletcher-Reeves method with the strong Wolfe line search which indicates that the analysis is under the sufficient decent condition. In [113], Gilbert and Nocedal also studied the global convergence for the algorithms related to the Polak-Ribière method by considering different choice of β_k. Liu, Han and Yin [114] demonstrated the global convergence of the Fletcher-Reeves method under some conditions that are weaker than those in [110]. Dai and Yuan proved that the Fletcher-Reeves method with the strong Wolfe line search is globally convergent provided that the search direction is downhill in each iteration. Dai, Han, Liu., et al [115] demonstrated the global convergence of the Fletcher-Reeves- and Polak-Ribière-type methods without assuming the sufficient descent condition. This paper showed that the sufficient descent condition is no longer a need in the analysis of the global convergence of the Fletcher-Reeves and Polak-Ribière method.

2.3 Existing Methods for Degree Reduction of Bézier Curve

In this dissertation, Bézier curve is used to represent the transition path. Bézier curve is a parametric curve which is defined by a set of control points. The number of control points determines the degree of the Bernstein polynomial basis functions that describe the shape of the curve. The curve interpolates its first and last control points and is tangent to the first and last sides of the open polygon defined by these control points [116]. We use degree elevation and reduction to maintain the space between control points during MEP search. Degree elevation of Bézier curve is exact. In contrast, degree reduction always has approximation involved. Degree reduction of Bézier curve is a process that uses a lower order curve to approximate a higher order curve. There are two major applications for the degree reduction of Bézier curve. One is to generate a piecewise linear approximation to a prescribed curve. The other is to transfer data from one geometric modeling system to another. During the process, usually a curve with higher degree must be approximated by several lower degree curves due to the limitation on the maximum polynomial degree that certain systems can store and work with.

Intensive research has been conducted on the degree reduction of Bézier curve in order to minimize the error between the original and reduced curve. The approximation to the original curve depends heavily on the chosen distance or error function to be minimized. There are typically two categories of approaches. One is to approximate the shape for the curve. The other is to approximate polynomial function that defines the curve. It treats the degree reduction process as the inverse process of degree elevation. Hence, the degree reduction problem is shifted to the problem of solving an over-determined linear system for the polynomial coefficients that defines the curve. Watkins and Worsey [117] developed a degree reduction method which is based on minimax approximation techniques. The method looks at the degree reduction from the perspective of the curve itself and as a result can achieve a better approximation to the original curve measured in the uniform norm. Eck [118] extended the degree reduction scheme originally proposed by Forrest [119] by introducing real weighting factors to blend the two set of coefficients together. It is a linear interpolation of the two set of coefficients. This shifts the problem of degree reduction to a problem of determining a set of real weighting factors. By minimizing the maximal Euclidean distance between the reduced curve and original curve based constrained Chebyshev polynomials, the weights could be determined. However, this algorithm requires intensive implementation effort because the constrained Chebyshev polynomials are known implicitly in general. In [120], Eck improved the algorithm by minimizing the least squares distance function using constrained Legendre polynomials.

Bogacki and Weinstein [121] developed two algorithms (one-degree reduction and multiple degree reduction) that compute a constrained approximation of an n^{th} degree Bézier curve by an $m^{th}(m<n)$ degree curve. The approximation is performed in the uniform norm applied component-wise with endpoint interpolation. Brunnett et al. [122] studied the optimal degree reduction (optimal approximation of an n^{th} degree Bézier curve by an $m^{th}(m<n)$ degree curve) with respect to different norms, particularly to L_p norms and the uniform norm ($p=\infty$). Kim and Moon [123] addressed the degree reduction problem in the L^1 norm with endpoint interpolation. The scheme gives the best one-degree reduction of Bézier curve of the degree less than six with endpoint interpolation by using splines. For higher order curves, they proposed a scheme which is based on an appropriate transform of the Chebyshev polynomials of second kind. Kim and Ahn [124] developed a C^1 constrained degree reduction method using the constrained Jacobi polynomials, the coefficients of which are represented explicitly, as the error function for good degree reduction of Bézier curve. In [125], Ahn extended the C^1 constrained degree reduction method to C^k $(k=2,3)$ constrained degree reduction using the constrained Jacobi polynomials.

In recent years, multi-degree reduction has been intensively studied. Instead of conducting the multi-degree reduction stepwise, methods are developed to perform multi-degree reduction at a time. Chen and Wang [126] developed a method named MDR by L_2 norm, which gives an explicit form of the least squares solution of multi-degree solution of Bézier curve with constraints of endpoints continuity. Sunwoo [127] generalized Chen and Wang's work [126] by finding an explicit form of the multi-degree reduction matrix for a Bézier curve with constraints of endpoints continuity. The control points of the degree reduced curve can be expressed as a product of degree reduction matrix and the vector of original control points. Lu and Wang [128, 129] developed a multi-degree reduction method with G^2 continuity under L_2 norm. Later on, they developed another multi-degree reduction with respect to the $\sqrt{t-t^2}$ -weighted square norm by using the transformation matrix between Bernstein and Chebyshev basis. Rababah and Lee [130] developed a simple matrix form for r times degree reduction with respect to the weighted L_2 -norm by using the matrices of transformations between Chebyshev and Bernstein basis. Woźny and Lewanowicz [131] proposed a multi-degree reduction method under L_2 norm by using dual Bernstein basis polynomials. A re-parameterization-based multi-degree method is developed by Chen and Ma [132] recently, which introduces a piecewise linear function to replace the general t in the least square distance.

2.4 PES Fitting Methods and Associated Uncertainties

The prediction of transition rates for a phase transition process requires the knowledge of the PES for the transition system. A PES is a mathematical function that provides the electronic ground state energy for a system of atoms defined by their relative positions in the space. The general process to construct a PES is as follows. First, ground state energy data points are collected by conducting *ab initio* calculations or experiments. Second, energy data points are fitted using analytical functions that are able to reproduce the energy at those energy data points. During the past three decades, a number of fitting methods [133-138] have been developed to construct the analytical forms of PESs. Those methods can be categorized as global and local methods.

Global methods fit the PES without weighting any *ab initio* data points. The adjustment made to any fitting data points will change the potential values at each point on the fitted PES. Global methods include spline methods [139-143], hybrid methods such as Morse-spline [144] and rotated Morse-spline [145-147] methods, least square methods [148] , reproducing kernel Hilbert space methods [149-151], and neural methods [137, 152-154]. The spline interpolation methods [139-143] describe the PES as an expansion of polynomial functions. The coefficients of the polynomial functions are determined by interpolating the *ab initio* points. Since the determination of the coefficients of the spline interpolation methods requires a large number of *ab initio* data points, the Morse-spline method [144] is developed. The method first fits one coordinate using Morse function with four fitting parameters that are functions of the rest coordinates. Then the explicit form of the four parameters is determined by spline interpolation on the data points of the rest coordinates. The rotated Morse-spline methods [145-147] are based on the rotated Morse function. The methods share the generalized ideals of the Morse-spline method.

The least square method [148] represents a PES as an expansion of basis functions with the geometry of the system as argument. The coefficients of basis functions are determined by minimizing the sum of weighted square errors between the calculated energies from the fitted functions and the one from *ab initio* calculations. The reproducing kernel Hilbert space methods [149-151] are a general way to do interpolation, in which the interpolation functions referred as kernels are determined by the order of smoothness and the boundary conditions associated with the chosen coordinates. The neural methods [137, 152-154] represent the fitting function as a neural network expression in which the parameters and weights are determined by learning processes.

In contrast to global methods, local methods add weights to each *ab initio* data points. As a result, each geometry of the fitted PES is determined by only the *ab initio* data points that are close to the geometry. The changes at one *ab initio* data point will affect only the fitted region that are close to the point. Local methods include modified Shepard interpolation methods [155-163], interpolating moving least squares (IMLS) methods, and neural networks methods. The modified Shepard methods [155-163] represent the PES as a weighted sum of Taylor expansions about each available *ab initio* data points, which requires the information of *ab initio* energies, gradients, and Hessians. The weight function is formulated as a function of the distance between the fitted point and the corresponding *ab initio* data point. For one particular *ab initio* point, the corresponding weight function is large when the interpolated position is close to the point and is small when is far from the point. Efforts have been made to optimize the weight function [162, 164] to improve the accuracy of the PES. The interpolating moving least square methods [136, 165-167] combine the moving least square method [168] and Shepard interpolation method without the requirement of *ab initio* gradient and hessian calculations.

The PES model that is constructed using one of the above fitting methods involves uncertainties that arise from four sources. First, *ab initio* data points are a source of error. Those data points are from *ab initio* calculations in which approximations are made to solve the Schrödinger equation. Second, the distribution of data points is another source of error. Lack of sufficient number of data points in the region of interested could lead to huge interpolation errors. For example, for the modified Shepard method, if there is no data point available in the region of interest, an error will be introduced since the accuracy of the local Taylor expansion depends on the expansion radius. Third, fitting methods are sources of error. Fitting and interpolation methods have coefficients or weights that are determined using optimization methods. In addition, methods have their own limitations. For example, the local Taylor expansion in the modified Shepard method introduces truncation errors. The weight functions that could be optimized by an optimal weighing scheme are sources of errors. Other examples include, but not limited to, the polynomial functions in the spline interpolation methods, and the basis and weight functions in least square methods.

2.5 Kriging Method

Kriging was originally developed in geostatistics by a South African mining engineer, Krige [169]. This method is further developed by Matheron [170, 171]. In 1989, Sacks et al. [172] first brought the application of Kriging method to the design and analysis of computer experiments. Currin et al. [173], and Welch et al. [174] further expand the application of Kriging in computer simulation. Morris et al. [175] extend the method to include the first derivative information in model construction. Since then, the Kriging method becomes popular in constructing surrogate models for deterministic but computationally expensive simulations and design optimization [176-184].

Kriging predicts functional values from a limited number of existing ones by modeling the unknown function as a Gaussian process. The response function $\mathrm{y}(\mathbf{x})$ with *n*-dimensional input **x** is composed by a polynomial model and a Gaussian process model with zero mean as [172]

$$\mathrm{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \varepsilon(\mathbf{x}) \tag{2.1}$$

in which $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_p(\mathbf{x})]^T$ is a vector of *p* basis polynomial functions, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^T$ are the corresponding unknown regression coefficients, and $\varepsilon(\mathbf{x}) \sim \mathrm{GP}(0, \sigma^2 \mathbf{R})$ represents the Gaussian process with zero mean and covariance

$$\mathrm{cov}(\varepsilon(\mathbf{x}_i), \varepsilon(\mathbf{x}_j)) = \sigma^2 \mathbf{R}(\theta_\varepsilon) \tag{2.2}$$

in which σ^2 is the process variance and $\mathbf{R}(\theta_\varepsilon)$ is the *m*×*m* correlation matrix for *m* data points with θ_ε as the process correlation parameter. For instance, for Gaussian correlation function, the element *ij* is defined as

$$R(\theta_\varepsilon, \mathbf{x}_i, \mathbf{x}_j) = \exp(-\textstyle\sum_{k=1}^{n} \theta_\varepsilon^{(k)} (x_i^{(k)} - x_j^{(k)})^2) \tag{2.3}$$

for *n* dimensional space with parameter $\theta_\varepsilon = (\theta_\varepsilon^{(1)}, \ldots, \theta_\varepsilon^{(n)})$. For an unknown position **x**, the vector of correlation is $\mathbf{r}(\mathbf{x}) = [R(\theta_\varepsilon, \mathbf{x}, \mathbf{x}_1), \ldots, R(\theta_\varepsilon, \mathbf{x}, \mathbf{x}_m)]^T$.

Given *m* data points with inputs $\mathbf{x}^* = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$, the corresponding output vector $\mathbf{Y} = [y_1, \ldots, y_m]^T$, and an *m*×*p* matrix of polynomial values $\mathbf{F} = [\mathbf{f}_1(\mathbf{x}^*), \ldots, \mathbf{f}_m(\mathbf{x}^*)]^T$, θ_ε is determined by the maximum likelihood estimation (MLE), which is to maximize the likelihood function L defined as

$$\ln L = -\frac{n \ln \sigma^2 + \ln |\mathbf{R}|}{2} \tag{2.4}$$

in which $\sigma^2 = (\mathbf{Y} - \mathbf{F}\boldsymbol{\beta})^T \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}\boldsymbol{\beta}) / m$ is the estimator of the process variance and $\boldsymbol{\beta} = (\mathbf{F}^T \mathbf{R}^{-1}\mathbf{F})^{-1}\mathbf{F}^T \mathbf{R}^{-1}\mathbf{Y}$ is the generalized least square estimator of coefficients. Parameter θ is determined by solving the MLE problem. After θ is determined, the regression coefficient matrix $\boldsymbol{\beta}$ can be calculated and then followed by the variance σ^2. The estimation of functional value at the unknown position **x** is

$$\hat{y}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x})^T \mathbf{R}^{-1}(\mathbf{Y} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{2.5}$$

The kriging methods can be categorized into two types: the ordinary kriging methods (OKG) and the universal Kriging methods (UKG) based on the order of polynomials used in the mean structure. The OKG assumes a zero or constant mean structure on the entire domain. The UKG constructs the mean structure using first or second-order polynomials. One extension for the UKG is the blind Kriging method [185] and dynamic Kriging method [186], both of which assume that the mean structure is unknown. In the blind Kriging method, the unknown mean model is identified using a Bayesian variable selection technique based on experimental data, while the dynamic Kriging method uses a genetic algorithm to select the optimal basis function locally.

Besides the application in deterministic simulation models, the kriging method also has been extended to approximate random or stochastic simulation models [187-189]. More detailed review of the Kriging methods and its application can be found in [190-195].

CHAPTER III
CONCURRENT SEARCH ALGORITH

The concurrent search algorithm is to locate multiple local minima and saddle points along one transition path without the prior knowledge of initial and final stable configurations on a PES simultaneously. Staring from one initial curve that represents one transition path, the algorithm locates multiple curves with their end points connected together located at multiple local minima. In addition, each curve has one control point located at the corresponding saddle point position. Different from the existing transition pathway and saddle point search methods, the algorithm is able to locate multiple local minima, saddle points, and corresponding transition paths between two stable configurations simultaneously. In addition, the algorithm represents the transition path using a parametric Bézier curve thus enables an efficient way that involves only linear interpolation of control points or images to maintain their even distribution along the path. This chapter gives a detailed description of the concurrent search algorithm. In addition, the algorithm is demonstrated by examples of LEPS potential function, LEPS harmonic oscillator potential function, Rastrigin function and Schwefel function. Also, the algorithm is used to study the diffusion process of the hydrogen in Iron Titanium (FeTi) system and body-centered iron (Fe) system.

3.1 Overview for the Concurrent Search Algorithm

The concurrent search algorithm has three stages: 1) a single transition pathway search, 2) multiple transition pathways search, and 3) climbing process to locate the saddle position. The algorithm represents the transition path using the control polygon of a Bézier curve. Each control point represents one intermediate state along the transition path. At the first stage, the algorithm locates two local minima by minimizing two end control points of the curve using the conjugate gradient method. In addition, the algorithm refines the shape of the curve through moving the intermediate control points along their corresponding conjugate directions with positive eigenvalues of the Hessian matrix for the PES. At the second stage, the algorithm locates all local minima between two stable configurations obtained from the first stage. To locate all local minima along the path, we developed a curve subdivision scheme to check whether there is more than one saddle point with an extra local minimum between two local minima (i.e. end points of the optimized curve produced from the first stage). If one curve is breakable, the algorithm breaks the curve into two curve sections representing two stages of transition. At the third stage, the algorithm locates all saddle points on multiple curves with their end points connected together. Multiple saddle points can be located by a proposed climbing up scheme.

The general process for the search algorithm is shown in Figure 3. An initial guess of the transition path is first provided. Then the path is optimized by minimizing the two end control points using the conjugate gradient method and meanwhile moving all the intermediate control points along the corresponding conjugate gradient directions. When two local minima are located by the end control points of the curve, the algorithm determines whether the curve crosses an extra local minimum or not. If not, the maximum energy point on the curve climbs up along the conjugate gradient directions to locate the true saddle point. If yes, the curve breaks into two new curves which represent initial guess for two new transition paths. Those two new curves are optimized and then checked following the same procedure as the initial curve. The check-and-break procedure continues until all the curves are unbreakable with their end control points locating at local minima and one intermediate control point locating at the saddle point.

3.2 A Single Transition Pathway Search

For the initial guess of a transition path which is represented by a single Bézier curve, the searching process for the stable configurations and the MEP is carried out in a sequential manner within a given iteration. A total of five control points are used for the initial curve. The more control points the curve has, the more accurate the search results will be, but with higher computational costs. The general process for a single transition pathway search is as follows. First, the two end control points of the curve are minimized by using the conjugate gradient method. Then, a set of conjugate directions for each intermediate control point is determined based on the new positions of the two end control points. Several minimization steps are applied to each intermediate control points along their associated conjugate directions. After several iterations, the two end control points of the curve will gradually converge to the minimum energy positions and the curve will approach to the MEP. Figure 4 illustrates the procedure of searching a single transition path. Table 1 lists the pseudo-code of the algorithm for a single transition pathway search. The details are described in the following subsections.

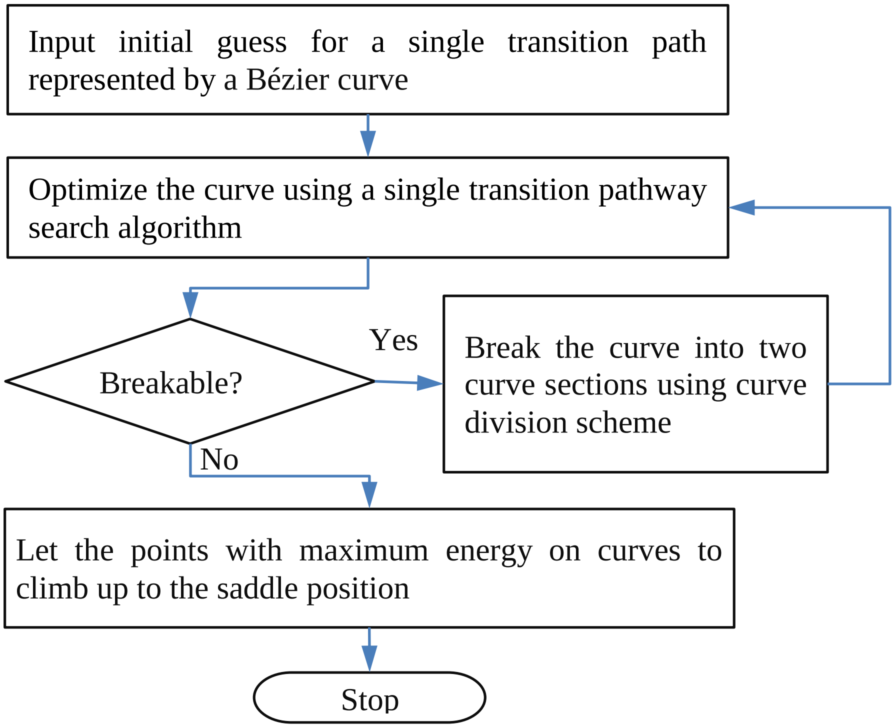

Input initial guess for a single transition path represented by a Bézier curve
Optimize the curve using a single transition pathway search algorithm
Breakable?
Yes
Break the curve into two curve sections using curve division scheme
No
Let the points with maximum energy on curves to climb up to the saddle position
Stop

Figure 3: Procedure for the concurrent search algorithm

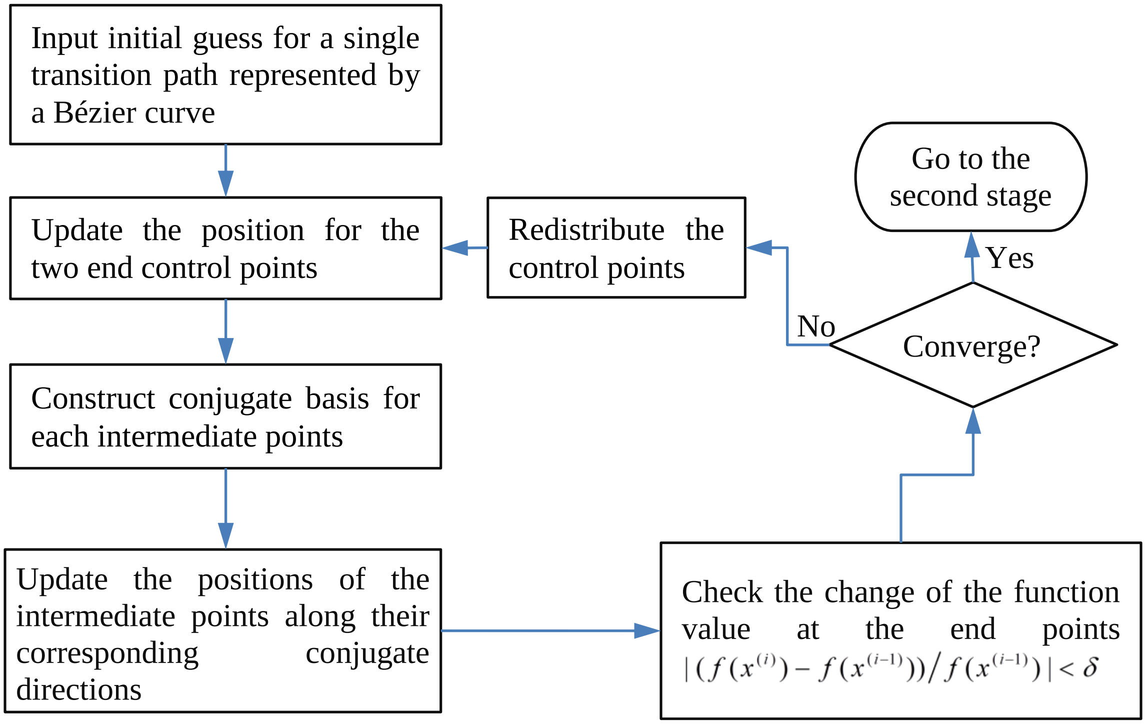

Input initial guess for a single transition path represented by a Bézier curve
Update the position for the two end control points
Redistribute the control points
Go to the second stage
Yes
No
Converge?
Construct conjugate basis for each intermediate points
Update the positions of the intermediate points along their corresponding conjugate directions
Check the change of the function value at the end points $|(f(x^{(i)})-f(x^{(i-1)}))/f(x^{(i-1)})|<\delta$

Figure 4: Procedure for the single transition path search

Table 1: Pseudo-code of the algorithm for a single transition pathway search

```
INPUT: Initial guess of a curve with control points of $\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4$.
OUTPUT: A curve with two end points located at two local minima and the curve itself approaches to the MEP.
TOL= threshold for the percentage of change in potential energy value;
$V(\mathbf{p}_*^{(\#)})$ =potential energy value at points $\mathbf{p}_*^{(\#)}$;
$\mathbf{D}_0^{(i)},\mathbf{D}_4^{(i)}$= search direction at $\mathbf{p}_0$ and $\mathbf{p}_4$ respectively;
$\alpha_0^{(i)},\alpha_4^{(i)}$=step size for minimizing $\mathbf{p}_0$ and $\mathbf{p}_4$ respectively;

WHILE True
    IF $\left\|(V(\mathbf{p}_0^{(i)})-V(\mathbf{p}_0^{(i-1)}))/V(\mathbf{p}_0^{(i-1)})\right\|>TOL$ and
$\left\|(V(\mathbf{p}_4^{(i)})-V(\mathbf{p}_4^{(i-1)}))/V(\mathbf{p}_4^{(i-1)})\right\|>TOL$
        $\mathbf{p}_0^{(i)}=\mathbf{p}_0^{(i-1)}+\alpha_0^{(i)}\mathbf{D}_0^{(i)}$; $\mathbf{p}_4^{(i)}=\mathbf{p}_4^{(i-1)}+\alpha_4^{(i)}\mathbf{D}_4^{(i)}$;
        Minimize $\mathbf{p}_1^{(i-1)},\mathbf{p}_2^{(i-1)},\mathbf{p}_3^{(i-1)}$ in their associated conjugate directions to get a new set of
        intermediate control points $\mathbf{p}_1^{(i)},\mathbf{p}_2^{(i)},\mathbf{p}_3^{(i)}$ (see section 3.2.2).
        IF There is zigzag along the curve
            Do degree elevation or reduction locally (see Section 3.2.3 and 3.2.4).
        END IF
    ELSEIF $\left\|(V(\mathbf{p}_0^{(i)})-V(\mathbf{p}_0^{(i-1)}))/V(\mathbf{p}_0^{(i-1)})\right\| < TOL$ and
$\left\|(V(\mathbf{p}_4^{(i)})-V(\mathbf{p}_4^{(i-1)}))/V(\mathbf{p}_4^{(i-1)})\right\| > TOL$
            $\mathbf{p}_4^{(i)} = \mathbf{p}_4^{(i-1)} + \alpha_4^{(i)}\mathbf{D}_4^{(i)}$;
            Minimize $\mathbf{p}_1^{(i-1)}, \mathbf{p}_2^{(i-1)}, \mathbf{p}_3^{(i-1)}$ in their corresponding conjugate directions to get a new
            set of intermediate control points $\mathbf{p}_1^{(i)}, \mathbf{p}_2^{(i)}, \mathbf{p}_3^{(i)}$.
            IF There is zigzag along the curve
                Do degree elevation or reduction locally (see Section 3.2.3 and 3.2.4).
            END IF
    ELSEIF $\left\|(V(\mathbf{p}_0^{(i)})-V(\mathbf{p}_0^{(i-1)}))/V(\mathbf{p}_0^{(i-1)})\right\| < TOL$ and
$\left\|(V(\mathbf{p}_4^{(i)})-V(\mathbf{p}_4^{(i-1)}))/V(\mathbf{p}_4^{(i-1)})\right\| > TOL$
            $\mathbf{p}_0^{(i)} = \mathbf{p}_0^{(i-1)} + \alpha_0^{(i)}\mathbf{D}_0^{(i)}$;
            Minimize $\mathbf{p}_1^{(i-1)}, \mathbf{p}_2^{(i-1)}, \mathbf{p}_3^{(i-1)}$ in their corresponding conjugate directions to get a new
            set of intermediate control points $\mathbf{p}_1^{(i)}, \mathbf{p}_2^{(i)}, \mathbf{p}_3^{(i)}$.
            IF There is zigzag along the curve
                Do degree elevation or reduction locally (see Section 3.2.3 and 3.2.4).
            END IF
    ELSE
        Stop.
    END IF
END WHILE
IF Two end points converge to the same local minimum
    Re-input the initial guess of the control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4$.
END
```

3.2.1 Searching the Stable Configuration

As shown in Table 1, the local minima are located by minimizing the two end control points of the curve iteratively. By definition the minimum energy location $\mathbf{x}^*$ on the PES satisfies

$$\nabla V\left(\mathbf{x}^*\right) = 0 \tag{3.1}$$

where $V(\mathbf{x}^*)$ is the potential energy function with respect to the position vector $\mathbf{x}^*$ in an *n*-dimensional configuration space; $\nabla V\left(\mathbf{x}^*\right)$ is the gradient of the potential on the PES at the location $\mathbf{x}^*$. The iterative location update during the minimization is given by

$$\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha^{(i)}\mathbf{d}^{(i)} \tag{3.2}$$

where $\alpha^{(i)}$ is the step length, and $\mathbf{d}^{(i)}$ is the search direction. The minimization process for the end points is carried out using the conjugate gradient method [107]. Detailed description about this method can be found in section 2.2. In our algorithm, the Fletcher-Reeves method is employed. The conjugate searching direction $\mathbf{d}^{(i)}$ in the i^{th} iteration are defined as a linear combination of $-\mathbf{g}^{(i)}$ and $\mathbf{d}^{(i-1)}$ [107],

$$\mathbf{d}^{(i)} = \begin{cases} -\mathbf{g}^{(i)} & \text{for } i = 1 \\ -\mathbf{g}^{(i)} + \dfrac{\left\|\mathbf{g}^{(i)}\right\|^2}{\left\|\mathbf{g}^{(i-1)}\right\|^2}\mathbf{d}^{(i-1)} & \text{for } i \geq 2 \end{cases} \tag{3.3}$$

The step size $\alpha^{(i)}$ is determined by using inexact line search which is presented in Section 3.2.5 along the corresponding conjugate directions. Namely, along each conjugate direction, several mini-steps are applied to the end points in order to locate the minima along that direction. The minimum position in one conjugate gradient direction is the starting point for the corresponding followed conjugate searching direction. In Table 1, in an n-dimensional search space, the search direction $\mathbf{D}^{(i)}$ in i^{th} iteration can be represented as

$$\alpha^{(i)}\mathbf{D}^{(i)} = \alpha_1^{(i)}\mathbf{d}_1^{(i)} + \cdots + \alpha_n^{(i)}\mathbf{d}_n^{(i)} \tag{3.4}$$

For a quadratic potential function with n-dimensional inputs, the local minimum can be determined in at most n steps. For a non-quadratic function, local quadratic approximation is involved during the minimization process. For a non-quadratic function with n-dimensional inputs, it requires more than n steps to locate a minimum. For those functions, the conjugate searching directions which are built based on the Eq. (3.3) will gradually lose conjugacy when searching process continues, which could lead to divergence. In our algorithm, we recalculate the conjugate directions from one iteration to another, namely after n steps of conjugate search in order to avoid the divergence.

3.2.2 Searching the MEP

Mathematically, for the n-dimension PES, the Hessian matrix $\mathbf{H}$(the matrix of the second derivative of the energy) at the first-order saddle points has one negative eigenvalue and $n-1$ positive ones. The eigenvectors $\mathbf{s}_i$ form a conjugate basis (i.e.

$\mathbf{s}_i{}^T\mathbf{H}\mathbf{s}_j = 0, \forall i \neq j$) with respect to the Hessian matrix. For a set of conjugate direction $\mathbf{s}_i$'s, in the vicinity of a first-order saddle point, there is one direction $\mathbf{s}_0$ along which the potential energy has a local maximum. For each of the other $n-1$ directions, the potential energy has a local minimum. The method presented here constructs a set of conjugate directions by making use of the Eq. (3.5) develop by Beale [196] which starts with a given arbitrary direction $\mathbf{s}_0$. The rest conjugate directions are defined as

$$\begin{aligned} \mathbf{s}_1 &= -\mathbf{g}_1 + \frac{\mathbf{g}_1{}^T(\mathbf{g}_1 - \mathbf{g}_0)}{\mathbf{s}_0{}^T(\mathbf{g}_1 - \mathbf{g}_0)}\mathbf{s}_0 \\ \mathbf{s}_{i+1} &= -\mathbf{g}_{i+1} + \frac{\mathbf{g}_{i+1}{}^T(\mathbf{g}_1 - \mathbf{g}_0)}{\mathbf{s}_0{}^T(\mathbf{g}_1 - \mathbf{g}_0)}\mathbf{s}_0 + \frac{\mathbf{g}_{i+1}{}^T\mathbf{g}_{i+1}}{\mathbf{g}_i{}^T\mathbf{g}_i}\mathbf{s}_i, \; i \geq 1 \end{aligned} \tag{3.5}$$

In this algorithm, for each intermediate control points, a set of corresponding conjugate directions are constructed by setting the $\mathbf{s}_0$ as the tangent direction approximated by the backward finite difference for the first half of the intermediate points and by the forward finite difference for the second half respectively. For example, for the k^{th} control point $\mathbf{p}_k$,

$$\mathbf{s}_0 = \begin{cases} \mathbf{p}_k - \mathbf{p}_{k-1} & \text{if } k \leq \left\lceil \dfrac{N}{2} \right\rceil \\ \mathbf{p}_k - \mathbf{p}_{k+1} & \text{if } k > \left\lceil \dfrac{N}{2} \right\rceil \end{cases} \tag{3.6}$$

where N is the total number of control points and $\lceil \quad \rceil$ rounds up to an integer. In order to calculate $\mathbf{s}_1$ in Eq.(3.5), we first need to determine $\mathbf{g}_0$ and $\mathbf{g}_1$. Here for the k^{th} control point $\mathbf{p}_k$, $\mathbf{g}_0$ is defined as the gradient at the middle point of the line segment connecting $\mathbf{p}_k$ and its neighbor, namely,

$$\mathbf{g}_0 = \begin{cases} \nabla V(\dfrac{\mathbf{p}_{k-1} + \mathbf{p}_k}{2}) & \text{if } k < \left\lceil \dfrac{N}{2} \right\rceil \\ \nabla V(\dfrac{\mathbf{p}_k + \mathbf{p}_{k+1}}{2}) & \text{if } k > \left\lceil \dfrac{N}{2} \right\rceil \end{cases} \tag{3.7}$$

$\mathbf{g}_1$ is defined as the gradient at the position with maximum energy $\mathbf{p}_{\max}$ along the direction $\mathbf{s}_0$. Several steps of line maximization are applied to the point $\mathbf{p}_k$ along $\mathbf{s}_0$ in order to locate $\mathbf{p}_{\max}$. Then several steps of line minimization along the conjugate direction $\mathbf{s}_1$ are applied to $\mathbf{p}_k$. The rest of the conjugate basis set are then built recursively using Eq.(3.5). Simultaneously, each time when a new conjugate direction is determined, several steps of line minimization along this direction are applied to the associated new positions of $\mathbf{p}_{\max}$.

3.2.3 Constrained Degree Elevation and Degree Reduction

After the evolution of the intermediate control points along the conjugate directions, those control points may become too close to each other. As a result, the control points only capture part of the information along the transition path. The resolution around the saddle region may be too low. This could lead to an underestimation of the energy barrier. Similar to the re-parameterization process in the string method [197], a redistribution process of the control points after each evolution step is introduced in order to ensure that these intermediate control points are relatively well distributed. The degree elevation and reduction scheme for the Bézier curve are employed to redistribute the intermediate control points in our algorithm.

Degree elevation increases the flexibility of a curve by introducing more degrees of freedom for control. By adding an extra control point to the definition of a Bézier curve, its degree is raised by one. The advantage of using the degree elevation technique is that we can increase the degree of a Bézier curve without changing its shape. The degree elevation of an n^{th} order Bézier curve by one produces an $(n+1)^{th}$ order Bézier curve with a new set of vertices $\mathbf{q}_k$ defined by [198]

$$\begin{cases} \mathbf{q}_k = \mathbf{p}_k, & k = 0 \\ \mathbf{q}_k = \dfrac{k}{n+1}\mathbf{p}_{k-1} + \left(1 - \dfrac{k}{n+1}\right)\mathbf{p}_k, & k = 1,\ldots,n \\ \mathbf{q}_k = \mathbf{p}_{k-1}, & k = n+1 \end{cases} \tag{3.8}$$

where $\mathbf{p}_k$'s are the original vertices of the n^{th} order Bézier curve. Eq. (3.8) can be written in a matrix form as

$$\mathbf{Q} = T_{\mathrm{n}}\mathbf{P} \tag{3.9}$$

where $\mathbf{Q} = (\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{n+1})^{\mathrm{T}}$, $\mathbf{P} = (\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_n)^{\mathrm{T}}$, and T_{n} is a $(n+2)$ by $(n+1)$ matrix defined as

$$T_{\mathrm{n}} = \frac{1}{n+1}\begin{bmatrix} n+1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & n & 0 & \cdots & 0 & 0 & 0 \\ 0 & 2 & n-1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & n-1 & 2 & 0 \\ 0 & 0 & 0 & \cdots & 0 & n & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & n+1 \end{bmatrix}$$

The Bézier curve can be elevated more than one degree by applying Eq.(3.4) multiple times. In our algorithm, the curve is elevated only once within each iteration in order to make control points well distributed.

The purpose of degree elevation in our algorithm is to redistribute the intermediate control points. In other words, we are concerned more about how well the procedure makes the control points distributed than about how small the error between the elevated curve and the original curve may have, as long as the introduced error is within a tolerance range. Based on those two considerations, a constraint is added to the original degree elevation scheme in order to better serve our purpose. When two control points become too close to each other after the degree elevation by Eq.(3.8), we manually set the new control point to be the arithmetic average of the two adjacent control points in the original curve. In other words, for each newly created control points of the elevated curve, we calculate the Euclidean distance between this point and the middle point of its corresponding adjacent points of the original curve. For the k^{th} control points $\mathbf{q}_k$ of the elevated curve, if it satisfies the condition

$$\left\| \mathbf{q}_k - \frac{\mathbf{p}_{k-1} + \mathbf{p}_k}{2} \right\| > c \left\| \mathbf{p}_k - \mathbf{p}_{k-1} \right\| \tag{3.10}$$

where $c\,(0 < c < 1)$ is a predefined constant, then $\mathbf{q}_k$ is set as the middle point of the straight line $\overrightarrow{p_{k-1}p_k}$. Since it is too computationally expensive to keep elevating the curve recursively, degree reduction is introduced to keep a balance with degree elevation and maintain a reasonable computational cost.

Degree reduction approximates an n^{th} order Bézier curve with an m^{th} $(m < n)$ order curve. Different from degree elevation, no exact degree reduction is possible in practice. Thus, approximation is inevitable. Similar to some of the existing methods reviewed in section 2.3, we treat the degree reduction as an inverse process of the degree elevation. Equation (3.8) shows that the control points of an elevated Bézier curve can be exactly determined by the control points of the original Bézier curve through linear interpolation of the two adjacent points. For the degree reduction, we need to solve the over-determined system in Eq.(3.8) for the unknowns $\{\mathbf{p}_k\}_{k=0}^{n}$ as a linear combination of $\{\mathbf{q}_k\}_{k=0}^{n+1}$.

Many methods are available to solve Eq. (3.8) approximately. Here, we developed a reduction scheme similar to Eck's method [120] which solves the equations by three steps. In Eck's method, the Eq. (3.8) is solved first in the forward direction. That is, for $k = 1, \dots, n-1$, we receive

$$\mathbf{p}_k^I = \frac{1}{n+1-k}\left(\left(n+1\right)\mathbf{q}_k - k\mathbf{p}^I{}_{k-1}\right) \tag{3.11}$$

where the superscript I indicates that the control points are obtained from the forward procedure. Then the control points $\left\{\mathbf{p}^I{}_k\right\}_{k=1}^{n-1}$ can be obtained recursively by setting $\mathbf{p}_{k-1}^I = \mathbf{q}_{k-1}$ when $k = 1$, which indicates that the two end control points are fixed for the degree reduction. Then Eq. (3.8) is solved in the backward direction. That is, for $k = n, \dots, 2$, we receive

$$\mathbf{p}_{k-1}^{II} = \frac{1}{k}\left(\left(n+1\right)\mathbf{q}_k - \left(n+1-k\right)\mathbf{p}^{II}{}_k\right) \tag{3.12}$$

where the superscript II indicates the control points is obtained from the backward procedure. And the control points $\left\{\mathbf{p}^{II}{}_k\right\}_{k=1}^{n-1}$ can be obtained recursively by setting $\mathbf{p}_k^{II} = \mathbf{q}_{k+1}$ when $k = n$. Thirdly, the unknown control points $\{\mathbf{p}_k\}_{k=0}^{n}$ of the reduced Bézier curve are calculated as a linear combination of the control points $\left\{\mathbf{p}^I{}_k\right\}_{k=1}^{n-1}$ and $\left\{\mathbf{p}^{II}{}_k\right\}_{k=1}^{n-1}$ as

$$\begin{cases} \mathbf{p}_k = \mathbf{q}_k & k=0 \\ \mathbf{p}_k = (1-\lambda_k)\mathbf{p}_k^I + \lambda_k \mathbf{p}_k^{II} & k=1,\ldots,n-1 \\ \mathbf{p}_k = \mathbf{q}_k & k=n \end{cases} \tag{3.13}$$

where λ_k is the weights for $\mathbf{p}_k^{II}$.

The degree reduction problem is then converted to the one of determining the weights of the corresponding control points. In Eck's method, λ_k's are determined by minimizing the least square distance between the original curve and the reduced curve, which is too costly for our purpose. Since the degree reduction in our algorithm is to redistribute the control points instead of transforming geometric information of curves which requires the error between the reduced curve and the original curve should be as small as possible. In order to reduce the computational cost, here the weights λ_k are defined as

$$\lambda_k = \frac{k}{n} \quad (k=1,\cdots,n-1) \tag{3.14}$$

The implementation test shows that this simplified degree reduction scheme makes the distribution of the points worse for some cases. Sometimes it introduces loops, which is undesirable for our algorithm. Thus we developed a reduction scheme similar to Eck's [120] but with a modified forward and backward procedure. In order to determine the new control points for the reduced curve, we make use of the information of three adjacent points instead of one as in Eck's scheme from the original curve. The three-step procedure is described as follows. In the forward step, three sets of points are calculated by using

$$\mathbf{p}_{k,1}^I = \frac{1}{n+1-k}\left((n+1)\mathbf{q}_k - k\mathbf{p}^I{}_{k-1}\right) \tag{3.15}$$

$$\mathbf{p}_{k,2}^I = \frac{1}{n+1-k}\left((n+1)\mathbf{q}_{k+1} - k\mathbf{q}_k\right) \tag{3.16}$$

and

$$\mathbf{p}_{k,3}^I = \frac{1}{n+1-k}\left((n+1)\mathbf{q}_{k+2} - k\mathbf{q}_{k+1}\right) \tag{3.17}$$

where $k=1,\ldots,n-1$. Then an average of them

$$\mathbf{p}_k{}^{I} = \frac{\mathbf{p}_{k,1}{}^{I} + \mathbf{p}_{k,2}{}^{I} + \mathbf{p}_{k,3}{}^{I}}{3} \tag{3.18}$$

forms a new set of points $\left\{\mathbf{p}^{I}{}_{k}\right\}_{k=1}^{n-1}$. Similarly, in the backward step, a new set of control points $\left\{\mathbf{p}^{II}{}_{k}\right\}_{k=1}^{n-1}$can be obtained by using Eqs.(3.19), (3.20), (3.21), and (3.22).

$$\mathbf{p}^{II}_{k-1,1} = \frac{1}{k}\left(\left(n+1\right)\mathbf{q}_k - \left(n+1-k\right)\mathbf{p}^{II}{}_{k}\right) \tag{3.19}$$

$$\mathbf{p}^{II}_{k-1,2} = \frac{1}{k}\left(\left(n+1\right)\mathbf{q}_{k-1} - \left(n+1-k\right)\mathbf{q}_{k}\right) \tag{3.20}$$

$$\mathbf{p}^{II}_{k-1,3} = \frac{1}{k}\left(\left(n+1\right)\mathbf{q}_{k-2} - \left(n+1-k\right)\mathbf{q}_{k-1}\right) \tag{3.21}$$

$$\mathbf{p}_{k-1}{}^{II} = \frac{\mathbf{p}_{k-1,1}{}^{II} + \mathbf{p}_{k-1,2}{}^{II} + \mathbf{p}_{k-1,3}{}^{II}}{3} \tag{3.22}$$

Finally, the new control points $\left\{\mathbf{p}_k\right\}_{k=0}^{n}$can be obtained by using Eq.(3.13) and (3.14).

3.2.4 Local Degree Elevation and Degree Reduction

The degree elevation and reduction of a Bézier curve changes the shape of the curve globally, which will gradually smooth out the curve. Consequently, this prevents the curve from converging to a curved MEP. The remedy for this issue is to introduce a local degree elevation and reduction scheme. Within each iteration, we first check whether there is zigzag along the curve or not. If there is no zigzag along the curve, we do not do degree elevation and reduction to the curve. Otherwise, we do degree elevation and reduction locally based on the distribution of the zigzag. For example, for the k^{th} control point $\mathbf{p}_k$ $(i = 1,\ldots,n-1)$, if it satisfies the condition

$$\arccos\left(\frac{\overrightarrow{p_{k-1}p_k} \cdot \overrightarrow{p_k p_{k+1}}}{\left\|\overrightarrow{p_{k-1}p_k}\right\|\left\|\overrightarrow{p_k p_{k+1}}\right\|}\right) > \beta \tag{3.23}$$

where $\beta (0 < \beta < \pi)$is a predefined constant, then it indicates that there is zigzag at the control point $\mathbf{p}_k$. We check each of the intermediate control points within each iteration. If there is no zigzag along the curve, degree elevation or reduction is not needed; otherwise, degree elevation or reduction is done locally. If the zigzag only exists within the first half of control points, degree elevation or reduction is only performed to the first half of control points. Similarly, it is performed only to the second half of control points if the zigzag only exists within the second half. If the zigzag exists in both, we do degree elevation or reduction globally.

3.2.5 Inexact line search

In multivariable optimization, the line search method is used to determine the searching direction and a suitable step length along that direction. Usually, the searching direction can be computed by various methods, such as gradient descent and conjugate gradient. In the concurrent search algorithm, the searching directions for the end control points are determined by conjugate gradient method, specifically by Fletcher-Reeves method. The searching directions for the intermediate control points are determined by the scheme developed by Beale [196].

After defined the searching direction, the key problem is to determine a step length in the searching direction such that the objective function value have certain amount decrease for a minimization problem and certain amount increase for a maximization problem. For example in Eqn. (3.2), let define

$$\phi(\alpha^{(i)}) = V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) \tag{3.24}$$

To find the step length along $d^{(i)}$ is to find $\alpha^{(i)}$ such that

$$\phi(\alpha^{(i)}) < \phi(0)$$

If the $\alpha^{(i)}$ we found makes the objective function a minimum along the direction $d^{(i)}$, i.e.,

$$V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) = \min_{\alpha>0} V(\mathbf{x}^{(i\text{-}1)} + \alpha\mathbf{d}^{(i)}) \tag{3.25}$$

such a line search is referred as exact line search and the step length $\alpha^{(i)}$ is referred as optimal step length. Usually, it is very computational expensive to find the optimal step length. In addition, for most of the cases, it is not necessary to find the optimal step length since the determined searching direction is an approximation to the ideal one which has the fast convergence rate when the step size along that direction is an optimal one. Hence, it is more practical to identify a step length such that the objective function value has acceptable amount decrease, i.e.,

$$V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) - V(\mathbf{x}^{(i\text{-}1)}) < -\zeta$$

in which ζ is a positive constant that represents the acceptable changes in the objective function value, such a line search is referred as inexact line search. The line search method for the algorithm developed in this dissertation is inexact line search.

To conduct the inexact line search, the key is to determine the termination conditions such that the identified step length $\alpha^{(i)}$ satisfies specified requirements. There are three popular termination conditions which are the Wolfe conditions, the strong Wolfe conditions and the Goldstein conditions. The Wolfe conditions stipulates that the step length $\alpha^{(i)}$ should give sufficient decrease in the objective function value. The condition is usually presented by inequality

$$V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) \le V(\mathbf{x}^{(i\text{-}1)}) + c_1\alpha^{(i)}\nabla V(\mathbf{x}^{(i\text{-}1)})^T\mathbf{d}^{(i)} \qquad (3.26)$$

in which c_1 is some constant between 0 and 1, i.e., $0 < c_1 < 1$. This condition means that the reduction in objective function V should be proportional to the step length $\alpha^{(i)}$ and the directional derivative $\nabla V(\mathbf{x}^{(i\text{-}1)})^T\mathbf{d}^{(i)}$. However, this Wolfe condition is not enough to ensure that the algorithm makes reasonable progress since the condition in (3.26) can be satisfied by any sufficient small $\alpha^{(i)}$. To eliminate such small step length, the curvature condition is added which requires $\alpha^{(i)}$ to satisfy the inequality

$$\nabla V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)})^T\mathbf{d}^{(i)} \ge c_2\nabla V(\mathbf{x}^{(i\text{-}1)})^T\mathbf{d}^{(i)} \qquad (3.27)$$

in which c_2 is some constant between c_1 and 1, i.e., $c_1 < c_2 < 1$. The strong Wolfe conditions requires the step length $\alpha^{(i)}$ to satisfy both the Wolfe condition and the curvature condition.

The Goldstein conditions ensures that the objective function has sufficient decrease with the identified step length $\alpha^{(i)}$ while preventing the step length $\alpha^{(i)}$ from being too small. The condition is usually presented in two inequalities

$$\begin{cases} V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) \le c\alpha^{(i)}\nabla V(\mathbf{x}^{(i\text{-}1)})^T\mathbf{d}^{(i)} \\ V(\mathbf{x}^{(i\text{-}1)} + \alpha^{(i)}\mathbf{d}^{(i)}) \ge (1-c)\alpha^{(i)}\nabla V(\mathbf{x}^{(i\text{-}1)})^T\mathbf{d}^{(i)} \end{cases} \qquad (3.28)$$

in which c is some constant between 0 and 1, i.e., $0 < c < \frac{1}{2}$. The first inequality is the Wolfe condition which ensures that the objective function has sufficient decrease. The second inequality ensures that the step length is not too small, which means the algorithm will make some progress after the advancement with step length $\alpha^{(i)}$. However, the second condition may excludes all the step length that are close to the minimum at that direction. In this dissertation, we implemented the inexact line search with strong Wolf conditions.

3.3 Multiple Transition Pathway Search

Here, we present how to search multiple transition paths on the PES. Our algorithm starts with the initial guess of a single transition path. Once the local minima are found as described in Section 3.2.1, this single path will be divided into two curves if an extra basin is located along the path. Both subdivided curves will then be treated individually and the algorithm will be applied to them. This subdivision process continues recursively until there is only one possible saddle point between any pair of local minima. As a result, multiple local minima and transition paths can be found within a target search area. Therefore, the initial guess of this single path should be set up such that the search area of interest can be covered.

During the multiple transition path search stage, a curve with two end control points located at the two local minima obtained from the single transition path search will be examined by using the curve subdivision scheme. It determines whether the curve crosses an extra basin with another local minimum. If yes, the curve is divided into two new curves at the intermediate control point that is located in the extra basin. Since the number of control points for those two newly created curves may be less than five, the degree elevation is applied to the two curves recursively until the number of control points for each curve reaches five. Those two elevated curves now represent the initial guesses for the two new transition paths. The elevated curves are optimized using the procedure listed in Table 1. After their respective local minima are identified, the curve subdivision scheme is applied to them again. The check-and-break procedure continues until all of the curves are unbreakable with their end control points located at local minima. By now, those curves are still the approximations of the individual MEPs. In order to find the actual energy barrier for each curve, the algorithm selects the control point with the maximum energy and makes it climb up to locate the saddle point. During the climbing process, a set of conjugate directions corresponding to the identified control point with the maximum energy are constructed. Different from the procedure in the single transition path search, the point with the maximum energy will be first maximized along $\mathbf{s}_0$ direction, and then minimized along other directions $\mathbf{s}_i$'s ($i \geq 1$). The same procedure in the single transition path search, i.e. minimization along directions with positive eigenvalues, is applied to the rest of intermediate control points during the climbing process. This further makes the curve converge to the MEP.

Table 2 lists the pseudo-code of the algorithm for multiple transition path search. A curve with two end control points locating at two local minima is obtained from the single transition pathway search. The curve subdivision scheme is used to determine whether the curve crosses an extra local minimum or not. If not, energy point with maximum energy value on the curve climbs up along the conjugate gradient directions to locate the saddle point. If yes, the curve breaks into two new curves each of which has a total number of control points less than five. The degree elevation is applied to the two curves recursively until the number of control points for each curve reaches five. Those two elevated curves represent initial guess for two new transition paths. The elevated curves are optimized and then checked following the same procedure as the initial curve. The check-and-break procedure continues until all the curves are unbreakable with their end control points locating at local minima and one intermediate control point locating at the saddle point.

The major step during the multiple transition path search is to determine some criteria of whether a curve is breakable and which intermediate control point we should select to break the curve. Here the subdivision scheme for the fourth-order (with five control points) and fifth-order (with six control points) curves are used to demonstrate. If we use a curve with a degree lower than four, the limited number of control points may miss the detailed curvature information of the actual path on the PES. As a result, some of the local minima will be missed. The subdivision scheme can be similarly extended to higher-order curves.

Table 2: Pseudo-code of the algorithm for multiple transition pathway search

```
INPUT: A curve $\varphi(\mathbf{x})$ with two end control points located at two local minima.
OUTPUT: Multiple curves with their end points connected together locating at multiple local minima. Besides, each curve has one point locating at the saddle point position.

$N_i$=number of newly produced curve for $i^{th}$ iteration ($N_0$ is set to be 1).
$i=0$
WHILE There exists newly produced curves in $i^{th}$ iteration
    $i=i+1$;
    $N_i$=0;
    FOR $j=1,2...,N_{i-1}$
        IF $\varphi_j(\mathbf{x})$ is breakable (using the scheme listed in Table 5)
            Break the curve $\varphi_j(\mathbf{x})$ into two curves $\varphi_j^1(\mathbf{x})$ and $\varphi_j^2(\mathbf{x})$.
            $N_i = N_i + 2$;
        END IF
        IF the number of control points for $\varphi_j^1(\mathbf{x})$ or $\varphi_j^2(\mathbf{x})$ is less than five
            Do degree elevation to the curve $\varphi_j^1(\mathbf{x})$ or $\varphi_j^2(\mathbf{x})$.
        END IF
        Optimize $\varphi_j^1(\mathbf{x})$ and $\varphi_j^2(\mathbf{x})$ after degree elevation to get two optimized curve
        $\varphi_k(\mathbf{x})$ and $\varphi_{k+1}(\mathbf{x})$ ($k = N_i - 1$).
    END FOR
END WHILE
FOR $j=1,2...,$ (total number of non-breakable curves produced during the WHILE loop )
    Select the maxi-energy control point of the $\varphi_j(\mathbf{x})$ to climb up in order to locate the saddle point.
END FOR
```

3.3.1 Scheme for Selecting Breakpoint

In this section, we present a curve subdivision scheme to determine whether a curve can be divided into two curves and which control point to be selected as the breakpoint for this breakable curve. This curve subdivision scheme is based on an assumption that the control points of a Bézier curve are relatively evenly distributed in a sequential manner. In other words, the curve itself has no loop or big curvature. We make use of the information of the gradient and potential energy value at each of the intermediate control points as well as their relative positions. Figure 5 shows a Bézier polygon on the PES with two end control points located at the minima of two separate basins of local minima. $\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3$ and $\mathbf{p}_4$ are control points. $-\nabla V(\mathbf{p}_1)$, $-\nabla V(\mathbf{p}_2)$,and $-\nabla V(\mathbf{p}_3)$ illustrate the negative gradient directions at the position $\mathbf{p}_1,\mathbf{p}_2,$ and $\mathbf{p}_3$ respectively. $\theta_1,\theta_2,$ and θ_3 are the angles between the negative gradient and the control polygon. By examining the three angles as well as the potential energy values at those intermediate control points, it is able to determine whether the curve crosses a third basin of local minima. There are a total of eight combinations with the angle distributions. The process of this scheme includes three steps. The first step is to check the combination of θ_1 and θ_3. If no conclusion can be reached, a second step is to check θ_2. If we still cannot decide by the second step, the energy values at the intermediate points will be considered as the third step.

Table 3 summarizes the curve subdivision scheme for a fourth order curve. The details about the scheme to determine a breakpoint for a fourth-order curve is described in the remainder of this section.

The first step of the process is to check the angles θ_1 and θ_3. If both θ_1 and θ_3 are larger than $\pi/2$ (i.e. Case 2 and Case 3 in Table 3) not only $\mathbf{p}_0$ and $\mathbf{p}_1$ are in different basins of local minima but also $\mathbf{p}_3$ and $\mathbf{p}_4$, which indicates that the curve crosses at least a third basin of local minimum. Any of the three intermediate control points could be a breakpoint. In our algorithm, we choose $\mathbf{p}_2$ as the breakpoint. If either θ_1 or θ_3 is less than $\pi/2$ (i.e. Case 1, 4, 5, 6, 7, and 8 in Table 3), it is not guaranteed that the curve would go through a third basin of local minimum by checking θ_1 and θ_3 only. For example, when θ_1 is larger than $\pi/2$ and θ_3 is less than $\pi/2$, there are two sets of possible positions for the control points, i.e. Case 1 and 4. Since θ_3 is less than $\pi/2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ could be located in the same basin. If $\mathbf{p}_2$ is located in a different basin from $\mathbf{p}_3$, the curve crosses the third basin. Otherwise, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$ could be in the same basin and the curve crosses only two adjacent basins. Therefore, we are unable to decide whether the curve is breakable or not with the only information that θ_1 is larger than $\pi/2$ and θ_3 is less than $\pi/2$. More information is required.

As a second step, we take $\mathbf{p}_2$ into consideration by checking θ_2. Here we use Cases 1 and 4 to illustrate. When θ_2 is less than $\pi/2$ (Case 1 in Table 3), it indicates that $\mathbf{p}_2$ cannot be located in the same basin as $\mathbf{p}_3$ and $\mathbf{p}_4$. Also as discussed in the first step, $\mathbf{p}_0$ and $\mathbf{p}_1$ are located in two different basins as in Case 1. Thus the curve should cross at least a third basin. Either $\mathbf{p}_1$ or $\mathbf{p}_2$ could be a breakpoint. Here, we select $\mathbf{p}_2$ to break. When θ_2 is larger than $\pi/2$ (Case 4 in Table 3), the negative gradients at the position $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are in the similar directions. $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$ could be in the same basin which means that the curve crosses only two adjacent basins of local minima. Thus we need further information to determine if the curve is breakable.

In the third step, the potential energy values at $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are considered. If the potential energy values at positions $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ have the monotonic relationship $V(\mathbf{p}_1) > V(\mathbf{p}_2) > V(\mathbf{p}_3)$, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are considered as in the same basin, although there is still a slight chance that they are not. The curve is defined as unbreakable under this condition; otherwise, we break up the curve at the point $\mathbf{p}_1$.

The above three-step procedure for Cases 1 and 4 can be extended to Cases 5, 6, 7, and 8. For Cases 5 and 8, θ_1 is less than $\pi/2$ and θ_3 is larger than $\pi/2$. When θ_2 is larger than $\pi/2$ (Case 8 in Table 3), the curve is breakable at the points $\mathbf{p}_2$ and $\mathbf{p}_3$. Here we select $\mathbf{p}_2$ as the break point. When θ_2 is less than $\pi/2$ (Case 8 in Table 3), and $V(\mathbf{p}_3) > V(\mathbf{p}_2) > V(\mathbf{p}_1)$, the curve is unbreakable; otherwise, we break it at $\mathbf{p}_3$. When both θ_1 and θ_3 are less than $\pi/2$ (Cases 6 and 7 in Table 3), the additional information of θ_2 does not help to determine. Hence we use the potential energy value directly. When a curve crosses two adjacent basins and the control points are relatively evenly distributed, the energy level at the middle point should be the largest. Based on this fact, when

$V(\mathbf{p}_2) > V(\mathbf{p}_1)$ and $V(\mathbf{p}_2) > V(\mathbf{p}_3)$, the curve is defined as unbreakable in the algorithm; otherwise, at the break point is chosen as $\mathbf{p}_2$.

The above procedure for breaking a fourth order curve can be extended to higher order curves. As an example, Table 4 summarizes the curve subdivision scheme for a fifth order curve. The discussion of the algorithm in this thesis is based on a fourth order curve. As an example, Table 5 lists the pseudo-code for breaking a curve with five and six control points.

Table 3: Curve subdivision scheme (five control points)

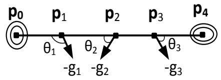
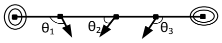
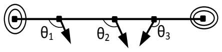
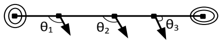
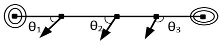
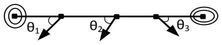
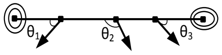
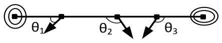
Eight Cases		Greater (>) or smaller (<) than $\pi/2$			Breakable ?
		θ_1	θ_2	θ_3	
1	 p₀ p₁ p₂ p₃ p₄ θ₁ θ₂ θ₃ -g₁ -g₂ -g₃	>	<	<	Break at $\mathbf{p}_2$
2		>	<	>	Break at $\mathbf{p}_2$
3		>	>	>	Break at $\mathbf{p}_2$
4		>	>	<	If $V(\mathbf{p}_1) > V(\mathbf{p}_2) > V(\mathbf{p}_3)$, the curve is defined as unbreakable; Otherwise, break at $\mathbf{p}_1$.
5		<	<	>	If $V(\mathbf{p}_3) > V(\mathbf{p}_2) > V(\mathbf{p}_1)$, the curve is defined as unbreakable; Otherwise, break at $\mathbf{p}_3$
6		<	<	<	If $V(\mathbf{p}_1) < V(\mathbf{p}_2)$ and $V(\mathbf{p}_3) < V(\mathbf{p}_2)$, the curve is defined as unbreakable; Otherwise, break at $\mathbf{p}_2$
7		<	>	<	Same as case 6
8		<	>	>	Break at $\mathbf{p}_2$

Table 4: Curve subdivision scheme (six control points)

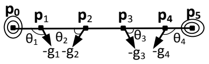
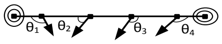
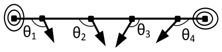
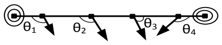
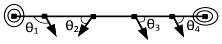
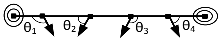
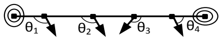
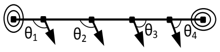
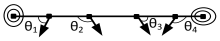
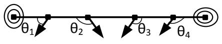
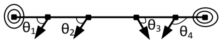
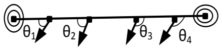
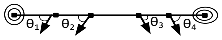
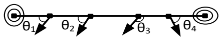
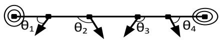
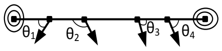
Sixteen Cases		Greater (>) or smaller (<) than $\pi/2$				Breakable?
		θ_1	θ_2	θ_3	θ_4	
1		>	<	<	>	Break at $\mathbf{p}_2$
2		>	<	>	>	Break at $\mathbf{p}_2$
3		>	>	>	>	Break at $\mathbf{p}_2$
4		>	>	<	>	Break at $\mathbf{p}_2$
5		>	<	<	<	Break at $\mathbf{p}_2$
6		>	<	>	<	Break at $\mathbf{p}_3$
7		>	>	>	<	Break at $\mathbf{p}_3$
8		>	>	<	<	If $V(\mathbf{p}_1)>V(\mathbf{p}_2)>V(\mathbf{p}_3)>V(\mathbf{p}_4)$, the curve is defined as unbreakable; otherwise, break at $\mathbf{p}_2$
9		<	>	<	>	Break at $\mathbf{p}_3$
10		<	>	>	>	Break at $\mathbf{p}_3$
11		<	<	<	>	Break at $\mathbf{p}_3$
12		<	<	>	>	If $V(\mathbf{p}_4)>V(\mathbf{p}_3)>V(\mathbf{p}_2)>V(\mathbf{p}_1)$, the curve is defined as unbreakable; otherwise, break at $\mathbf{p}_4$
13		<	<	<	<	If $V(\mathbf{p}_3)>V(\mathbf{p}_4)$ and $V(\mathbf{p}_2)>V(\mathbf{p}_1)$, the curve is defined as unbreakable; otherwise, if $V(\mathbf{p}_2)<V(\mathbf{p}_1)$, break at $\mathbf{p}_2$, else break at $\mathbf{p}_3$
14		<	<	>	<	If $V(\mathbf{p}_3)>V(\mathbf{p}_2)>V(\mathbf{p}_1)$, the curve is defined as unbreakable; otherwise, break at $\mathbf{p}_3$
15		<	>	>	<	Break at $\mathbf{p}_2$
16		<	>	<	<	If $V(\mathbf{p}_2)>V(\mathbf{p}_3)>V(\mathbf{p}_4)$, the curve is defined as unbreakable; otherwise, break at $\mathbf{p}_2$

Table 5: Pseudo-code of the curve division scheme (five and six control points)

```
INPUT: An optimized curve $\varphi(\mathbf{x})$ with two end control points located at two local minima.
OUTPUT: Two curve sections of $\varphi(\mathbf{x})$
IF $\varphi(\mathbf{x})$ has five control points $\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4$ (refer to Figure 5)
   IF $\theta_1 > \pi/2$ and $\theta_3 > \pi/2$
      $\mathbf{p}_2$ is selected as breakpoint
   ELSEIF $\theta_1 > \pi/2$ and $\theta_3 < \pi/2$
         IF $\theta_2 < \pi/2$
            $\mathbf{p}_2$ is selected as breakpoint
         ELSE
            IF $V(\mathbf{p}_1) > V(\mathbf{p}_2) > V(\mathbf{p}_3)$
               $\varphi(\mathbf{x})$ is non-breakable
            ELSE
               $\mathbf{p}_1$ is selected as breakpoint
            END IF
         END IF
   ELSEIF $\theta_1 < \pi/2$ && $\theta_3 > \pi/2$
         IF $\theta_2 > \pi/2$
            $\mathbf{p}_2$ is selected as breakpoint
         ELSE
            IF $V(\mathbf{p}_3) > V(\mathbf{p}_2) > V(\mathbf{p}_1)$
               $\varphi(\mathbf{x})$ is non-breakable
            ELSE
               $\mathbf{p}_1$ is selected as breakpoint
            END IF
         END IF
   ELSEIF $\theta_1 < \pi/2$ and $\theta_3 < \pi/2$
         IF $V(\mathbf{p}_2) > V(\mathbf{p}_1)$ and $V(\mathbf{p}_2) > V(\mathbf{p}_3)$
            $\varphi(\mathbf{x})$ is non-breakable
         ELSE
            $\mathbf{p}_2$ is selected as breakpoint
         END IF
   END IF
ELSEIF $\varphi(\mathbf{x})$ has six control points $\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4,\mathbf{p}_5$ (refer to Figure 6)
   IF $\theta_1 > \pi/2$ and $\theta_4 > \pi/2$
      $\mathbf{p}_2$ is selected as breakpoint
```

```
  ELSEIF $\theta_1 > \pi/2$ or $\theta_4 < \pi/2$
      IF Either $\theta_2 < \pi/2$ or $\theta_3 < \pi/2$
         $\mathbf{p}_2$ is selected as breakpoint
      ELSE
         IF $V(\mathbf{p}_1) > V(\mathbf{p}_2) > V(\mathbf{p}_3) > V(\mathbf{p}_4)$
             $\varphi(\mathbf{x})$ is non-breakable
          ELSE
               $\mathbf{p}_1$ is selected as breakpoint
          END IF
       END IF
  ELSEIF $\theta_1 < \pi/2$ and $\theta_4 > \pi/2$
      IF Either $\theta_2 > \pi/2$ or $\theta_3 > \pi/2$
         $\mathbf{p}_3$ is selected as breakpoint
      ELSE
         IF $V(\mathbf{p}_4) > V(\mathbf{p}_3) > V(\mathbf{p}_2) > V(\mathbf{p}_1)$
             $\varphi(\mathbf{x})$ is non-breakable
          ELSE
               $\mathbf{p}_4$ is selected as breakpoint
          END IF
       END IF
ELSEIF $\theta_1 < \pi/2$ and $\theta_4 < \pi/2$
     IF $\theta_2 < \pi/2$ and $\theta_3 > \pi/2$
           IF $V(\mathbf{p}_3) > V(\mathbf{p}_2) > V(\mathbf{p}_1)$
               $\varphi(\mathbf{x})$ is non-breakable
           ELSE
               IF $V(\mathbf{p}_1) > V(\mathbf{p}_2)$
                   $\mathbf{p}_2$ is selected as breakpoint
                ELSEIF $V(\mathbf{p}_2) > V(\mathbf{p}_3)$
                   $\mathbf{p}_3$ is selected as breakpoint
                END IF
           END IF
       ELSEIF $\theta_2 < \pi/2$ and $\theta_3 < \pi/2$
              IF $V(\mathbf{p}_2) > V(\mathbf{p}_1)$ and $V(\mathbf{p}_3) > V(\mathbf{p}_4)$
               $\varphi(\mathbf{x})$ is non-breakable
              ELSE
                  $\mathbf{p}_3$ is selected as breakpoint
               END IF
       ELSEIF $\theta_2 > \pi/2$ and $\theta_3 < \pi/2$
```

```
                    IF $V(\mathbf{p}_2) > V(\mathbf{p}_3) > V(\mathbf{p}_4)$
                        $\varphi(\mathbf{x})$ is non-breakable
                    ELSE
                        $\mathbf{p}_2$ is selected as breakpoint
                    END IF
            ELSEIF $\theta_2 > \pi/2$ and $\theta_3 > \pi/2$
                        $\mathbf{p}_2$ is selected as breakpoint
            END IF
    END IF
END IF
```

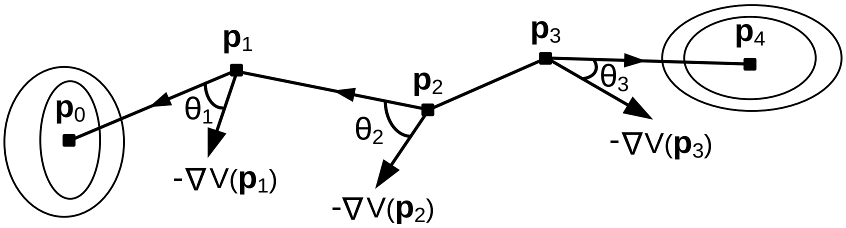


Figure 5: Illustration for multiple pathway search (five control points)

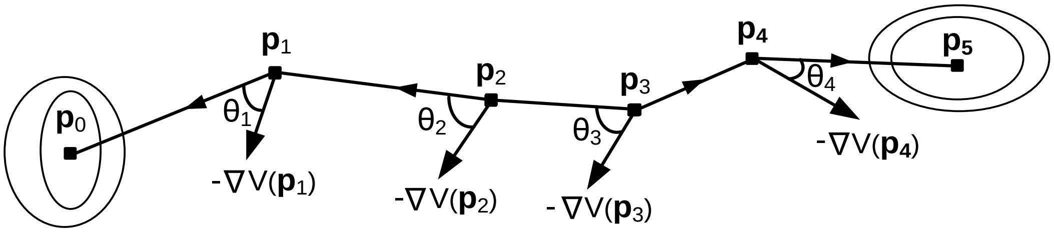


Figure 6: Illustration for multiple pathway search (six control points)

3.3.2 Discussion for the Subdivision Scheme

The proposed curve subdivision scheme for selecting the breakpoint is not perfect. It could treat some breakable curves as unbreakable ones. For example, for a curve with five control points, when $\theta_1 > \pi/2$, $\theta_3 < \pi/2$ and $V(\mathbf{p}_1) > V(\mathbf{p}_2) > V(\mathbf{p}_3)$, we define the curve as unbreakable. It is true if the curve only passes through two adjacent basins of local minima. But if the curve covers a long range with several extra local minima, there is still a small chance that the control points are positioned in the manner which satisfies the unbreakable conditions. The scheme will treat both of the two curves as unbreakable. A remedy for missing breakable curves is adding an extra step to double check each unbreakable curve for one more time. If a curve is identified as unbreakable curve for the first time, the control points of the curve will be redistributed by using degree elevation or degree reduction. Then this elevated or reduced curve will be checked again to see whether it is breakable. This extra step will increase the accuracy of subdivision but also with extra computational cost.

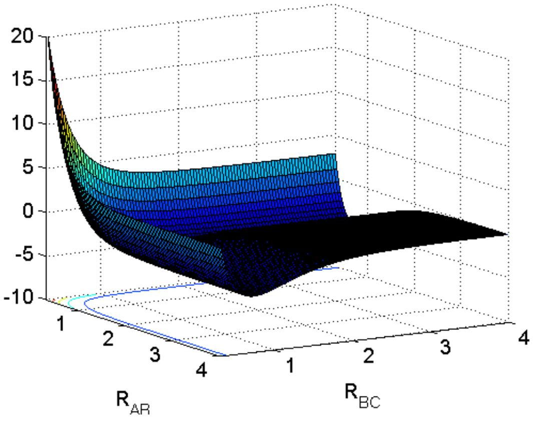
3.4 Implementation and Demonstration

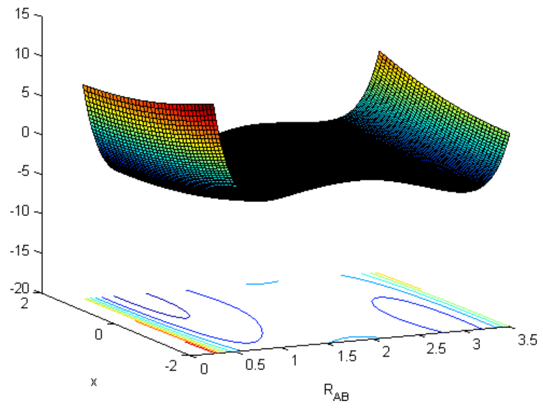
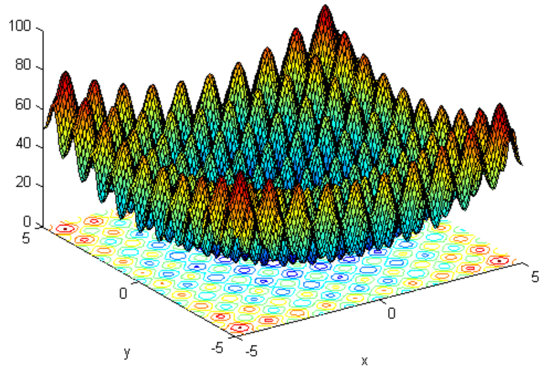
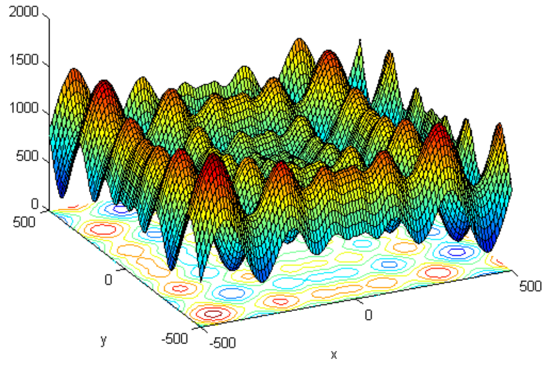
This chapter demonstrates the proposed concurrent search algorithm for multiple phase transition pathways. First, we test the algorithm of a single transition pathway on LEPS potential and LEPS plus harmonic oscillator potential [73, 199]. These two potential functions are two-dimensional benchmark problems which are frequently used to test transition pathway search methods. These two potential model mimics a reaction involving three atoms. Details will present in Section 3.4.1 and 3.4.2. Then we demonstrate the proposed multiple transition pathway search algorithm by applying it to search the saddle points and local minima on two different two-dimensional PESs defined by the Rastrigin function and Schwefel function respectively. The implementation was done using MATLAB. Source codes are included in the Appendix. Table 6 lists the definition and graphic in two-dimension for the four test functions. For LEPS potential, the Q functions illustrate Coulomb interactions between the electron clouds and the nuclei. The J functions illustrate the quantum mechanical exchange interactions. The parameters for the LEPS potential are defined as $a = 0.05,$ and $\alpha = 1.942$. $b = 0.30,$ $c = 0.05,$ $d_{AB} = 4.746,$ $d_{BC} = 4.746,$ $d_{AC} = 3.445,$ $r_0 = 0.742,$ For LEPS plus harmonic oscillator potential, the parameters are defined as $r_{AC} = 3.742,$ $k_c = 0.2025,$ and $c = 1.154$. All the other parameters are the same as the ones defined for LEPS potential except $b = 0.80$ in LEPS plus harmonic oscillator potential.

In addition, the algorithm is demonstrated on high-dimensional PESs by example of the diffusion process of hydrogen in iron-titanium (FeTi) and body centered iron (Fe).

FeTi is being extensively studied as a candidate material for hydrogen storage applications. However, the major issues for the FeTi as a hydrogen storage materials is that it requires a complicated activation process i.e. high temperature and hydrogen pressure. Extensively research has been done to improve the activation process by using elemental substitution [200-202]. It is meaningful to understand the diffusion process of the hydrogen in the FeTi system to design the elemental substitution. Steel (Fe) is one of the widely used engineering structure materials. Hydrogen can greatly change the mechanical properties of structural metals and alloys. Hydrogen embrittlement is believed to be one of the main reasons for cracking of steel structures under stress. To investigate the solution to control and prevent the hydrogen embrittlement, it is meaningful to understand the diffusion process of the hydrogen in the iron system at atomic level. The detailed implementation of the FeTi-H system and Fe8-H is presented in Section 3.4.5 and 3.4.6.

Table 6: Test functions

Function	Definition	Graphic in two-dimension
LEPS potential	$V^{LEPS}(r_{AB},r_{BC})$ $= \frac{Q_{AB}}{1+a} + \frac{Q_{BC}}{1+b} + \frac{Q_{AC}}{1+c} - [\frac{J_{AB}^2}{(1+a)^2} + \frac{J_{AB}^2}{(1+a)^2} + \frac{J_{AB}^2}{(1+a)^2}$ $- \frac{J_{AB}J_{BC}}{(1+a)(1+b)} - \frac{J_{BC}J_{AC}}{(1+b)(1+c)} - \frac{J_{AB}J_{AC}}{(1+a)(1+c)}]^{1/2}$ where $Q(r) = \frac{d}{2}\left(\frac{3}{2}e^{-2\alpha(r-r_0)} - e^{-\alpha(r-r_0)}\right)$ $J(r) = \frac{d}{4}\left(e^{-2\alpha(r-r_0)} - 6e^{-\alpha(r-r_0)}\right)$	

LEPS + Harmonic oscillator potential	$V(r_{AB},x)$ $=VLEPS(r_{AB},r_{AC}-r_{AB})+2k_c(r_{AB}-(r_{AC}/2-x/c))^2$	
Rastrigin	$f(\mathbf{x})=10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)]$	
Schwefel	$f(\mathbf{x})=418.9829n-\sum_{i=1}^{n}x_i\sin\sqrt{\lvert x_i\rvert}$	

3.4.1 Test Result for LEPS Potential

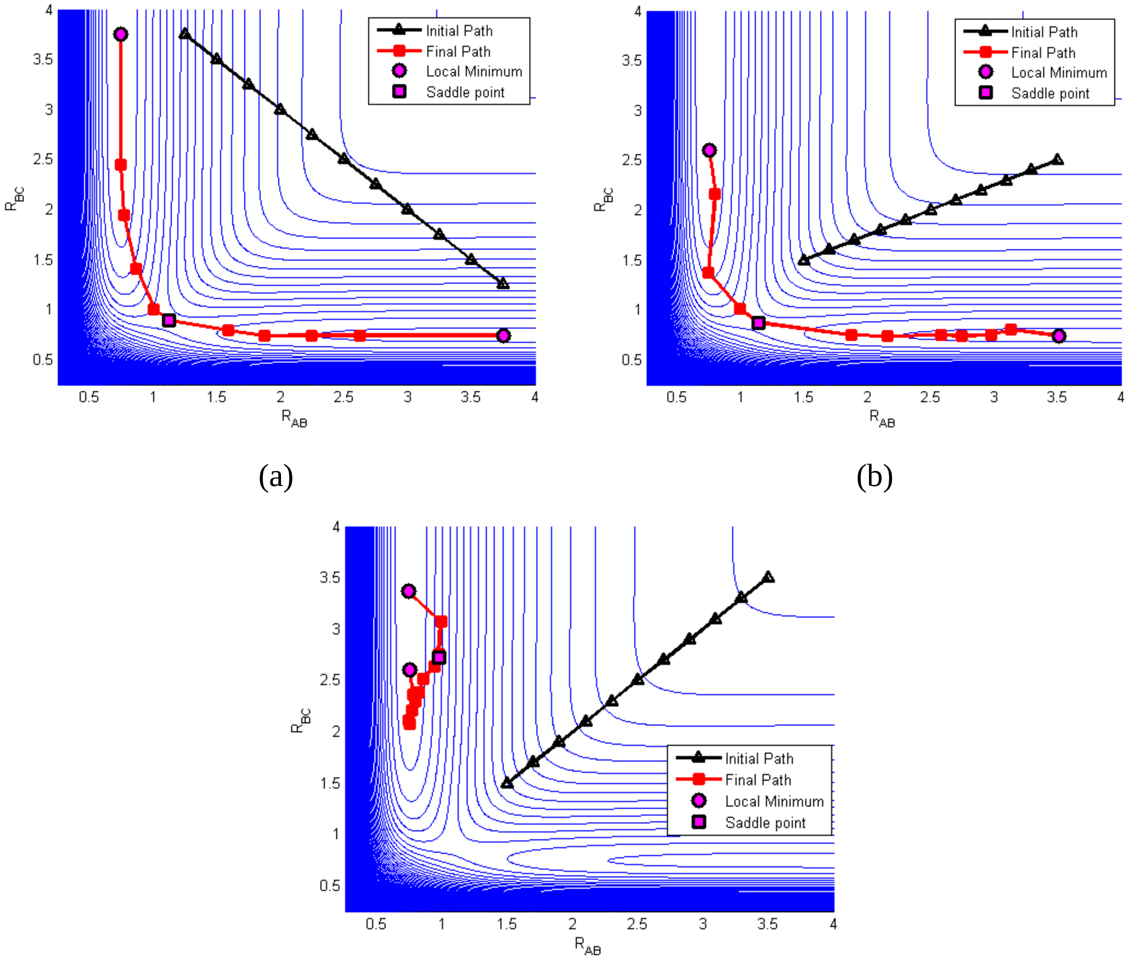
The LEPS potential model mimics a reaction involving three atoms A, B, and C constricted to motion along a straight line. There is only one bond formed, either between atoms A and B or between B and C [73, 199]. The detailed description and a 3-D graphic of the potential function can be found in Table 6. We test the single transition pathway search algorithm on this function by using two different initial positions of the transition path. The constant coefficient of the step size for minimizing the end control points and moving the intermediate control points is set to be 0.01 and 0.025 respectively. The results are illustrated in Figure 7 using contour plot. The black line represents the initial path. The red line represents the final path identified by using the algorithm listed in Table 1. The purple circle markers indicate the position of local minima, while the purple square marker represents the position of saddle point.

For different initial positions (refer to (a) and (b) in Figure 7), the algorithm locates the same saddle point while it locates different local minima. The result for locating different local minima is sensitive to the initial positions. This is due to the characteristic of the LEPS potential function. The 3-D graphic for LEPS potential function in Table 6 shows that there is a long flat valley around the local minimum region. Each point along the valley could be a potential local minimum. The algorithm will stop searching local minima as long as it locates one of the potential local minima. Starting from different initial positions, the algorithm will follow different searching path. As a result, it will locate different potential local minima. That explains why the located local minima are different for different initial positions.

For the initial position in Figure 7 (c), the algorithm failed to locate the saddle point. The two end control points tend to converge to the same local minimum. This is because that the conjugate gradient method is a local search method. The control point will converge to the nearest local optimal point. Since in the algorithm developed in this thesis, the two end control points are optimized independently based on conjugate gradient method, the two end control points will converge to the same local minima when the initial positions of the two points are close to the same local minimum.

3.4.2 Test Result for LEPS Plus Harmonic Oscillator Potential

Different with the LEPS potential model, the location of the two end atoms A and C in this model is fixed. Only atom B is allowed to move. In addition, this model introduces an additional degree of freedom which can be interpreted as a fourth atom that is coupled in a harmonic way with the atom B [73, 199]. The detailed description and a 3-D graphic of the potential function can be found in Table 6. We test the single transition pathway search algorithm on this function by using three different initial positions of the transition path. The constant coefficient of the step size for minimizing the end control points and moving the intermediate control points is set to be 1/35 and 1/45 respectively. The results are illustrated in Figure 8 using contour plot. The black line with triangle markers represents the initial path. The red line with square markers represents the final path identified by using the algorithm listed in Table 1.The purple circle markers indicate the position of local minima, while the purple square marker represents the position of saddle point. The results show that the algorithm is able to locate the local minima and saddle points for different initial positions.

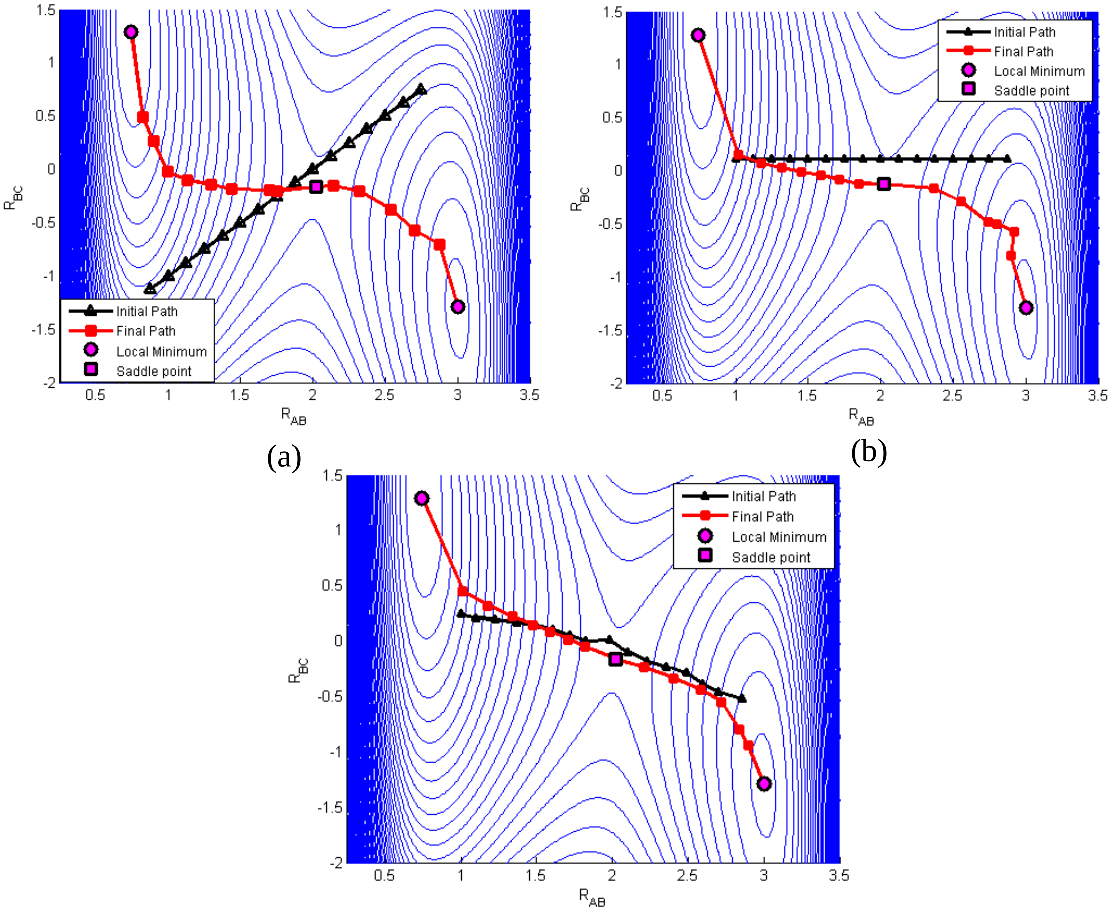


(a)

(b)

(c)

Figure 7: Test results for LEPS potential function



(a)

(b)

(c)

Figure 8: Test results for LEPS plus harmonic oscillator potential function

3.4.3 Test Result for Rastrigin Function

The Rasrigin function is a non-convex function frequently used to test the optimization algorithm. The function has a global minimum at $\mathbf{x} = (0, \cdots, 0)$ as well as several local minima. As discussed in Section 3.3, the initial guess for the transition path should be a curve with five control points which are relatively evenly distributed. Here, we choose a curve $\varphi(\mathbf{x})$ with five control points located at (−2.81, 0.50), (−1.43, 2.90), (0.23, −2.47), (1.57, 2.67), and (2.91, −0.11), which are visualized in Figure 9 as 'initial path'. First, the optimization procedure listed in Table 1 is applied to $\varphi(\mathbf{x})$, which produces a curve $\varphi'(\mathbf{x})$ with two end control points located at the two local minima. Then the multi-transition pathway search algorithm listed in Table 2 is applied to $\varphi'(\mathbf{x})$. A total of seven local minima and six corresponding saddle points are located by this algorithm. The positions of those local minima and the corresponding saddle points are listed in Table 7. Figure 9 shows the result using contour graphs. In order to test the robust of the algorithm, we test the algorithm by using a set of different initial positions located at (−2.81, −1.50), (−1.43, −1.50), (0.23, −1.50), (1.57, −1.50), and (2.91, −1.50), which are shown in Figure 10 as black dots. The result shows that the algorithm performs well for different initial positions. For the second set of initial positions, a total of seven local minima and six corresponding saddle points are located. The details about the local minima and corresponding saddle points are listed in Table 8. The result is also illustrated in Figure 10.

Table 7: Test results on Rastrigin function (contour plot refer Figure 9)

Path No	Local minima		Saddle
1	(−2.9849, 0)	(−1.9899, 0)	(−2.5516, 0.0201)
2	(−1.9899, 0)	(−0.9950, 0)	(−1.5484, 0.0210)
3	(−0.9950, 0)	(0, 0)	(−0.5345, −0.0133)
4	(0, 0)	(0.9950, 0)	(0.4656, −0.0116)
5	(0.9950, 0)	(1.9899, 0)	(1.4688, −0.0132)
6	(1.9899, 0)	(2.9849, 0)	(2.4742, 0.0472)

Table 8: Test results on Rastrigin function (contour plot refer Figure 10)

Path No	Local minima		Saddle
1	(−2.9849, −0.9950)	(−1.9899, −0.9950)	(−2.5497, −0.9832)
2	(0, −0.9950)	(0.9950, −0.9950)	(0.4591, −0.9764)
3	(−1.9899, −0.9950)	(−0.9950,−0.9950)	(−1.5496, −0.9776)
4	(−0.9950, −0.9950)	(0, −0.9950)	(−0.5441, −0.9738)
5	(0.9950, −0.9950)	(1.9899, −0.9950)	(1.5399, −0.9666)
6	(1.9899, −0.9950)	(2.9849, −0.9950)	(2.5491, −0.9816)

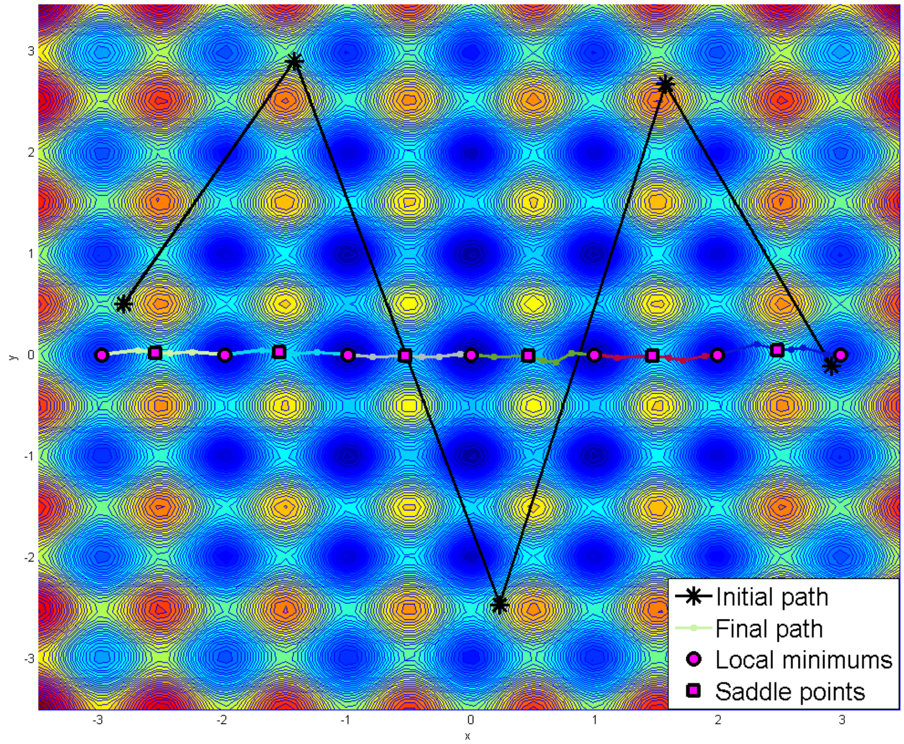


Figure 9: Test result for Rastrigin function with the initial position at (−2.81, 0.50), (−1.43, 2.90), (0.23, −2.47), (1.57, 2.67), and (2.91, −0.11).

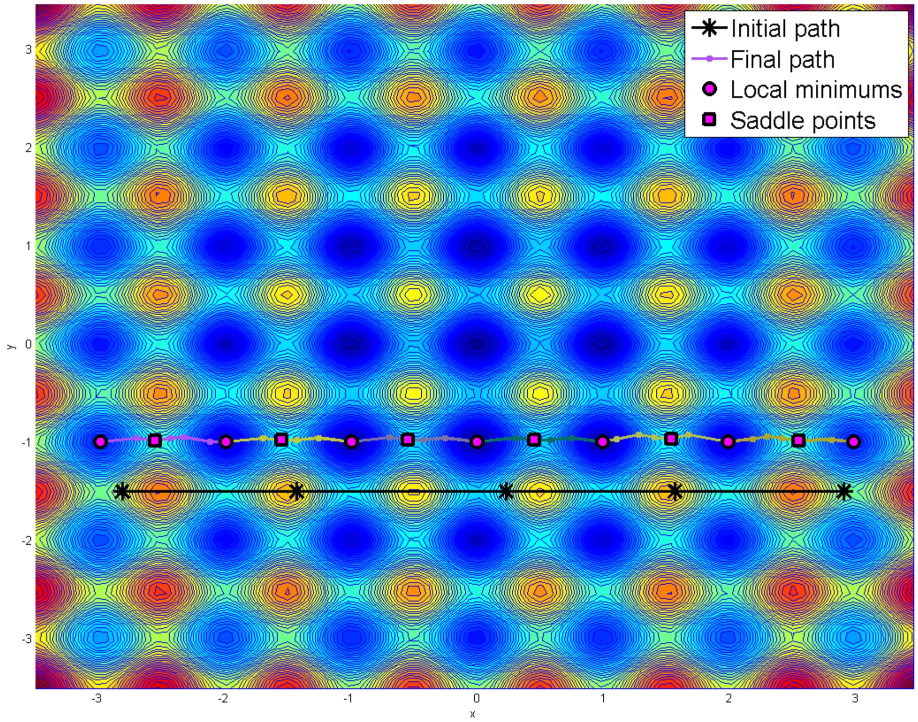


Figure 10: Test result for Rastrigin function with the initial position at (−2.81, −1.50), (−1.43, −1.50), (0.23, −1.50), (1.57, −1.50), and (2.91, −1.50).

3.4.4 Test Result for Schwefel Function

Section 3.4.3 shows that the algorithm works very well on the PES defined by Rastrigin function. But the contour of the Rastrigin function is uniformly distributed as we can see from the contour plot in Figure 9. In real application, most of the potential energy surfaces are non-uniform. Thus the Schwefel function which has a relatively non-uniform potential energy surface is selected to test our algorithm. We test our algorithm on the Schwefel surface following the same procedure as we did on Rastrigin surface. We also test the algorithm with two set of initial positions. The first initial positions are located at (−100.3, 25), (−40.5, 40), (17.8, −10), (69.8, 70.6), and (130.2, 98.7), which are illustrated in Figure 11 with black dots. A total of six local minima and five corresponding saddle points are located. Details are listed in Table 9. Also, the results are visualized in Figure 11. The second initial positions are located at (−100.3, −70), (−40.5, −70), (17.8, −70), (69.8, −70), and (130.2, −70), as shown in Figure 12 with black dots. A total of five local minima and four corresponding saddle points are located. Details are listed in Table 10. The results are visualized in Figure 12.

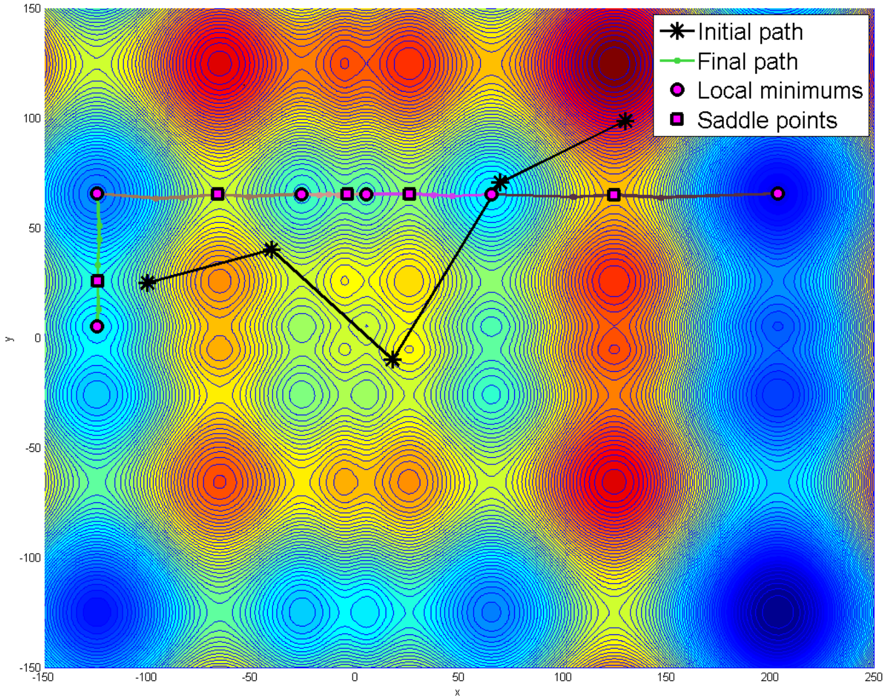


Figure 11: Test result for Schwefel function with the initial position at (−100.3, 25), (−40.5, -45), (17.8, 50.3), (69.8, 70.6), and (130.2, 98.7).

Table 9: Test results on Schwefel function (contour plot refer Figure 11)

Path No	Local minima		Saddle
1	(−124.8170, 5.2615)	(−124.4369, 65.4794)	(−124.2794, 25.7779)
2	(5.2807, 65.4046)	(65.5185, 65.2612)	(26.0773, 65.3830)
3	(65.5185, 65.2612)	(203.7441, 65.5489)	(124.8765, 65.0411)
4	(−124.6262, 65.5132)	(−26.2200, 65.4095)	(−66.1930, 65.1174)
5	(−26.2200, 65.4095)	(5.2516, 65.4740)	(−3.9133, 65.3002)

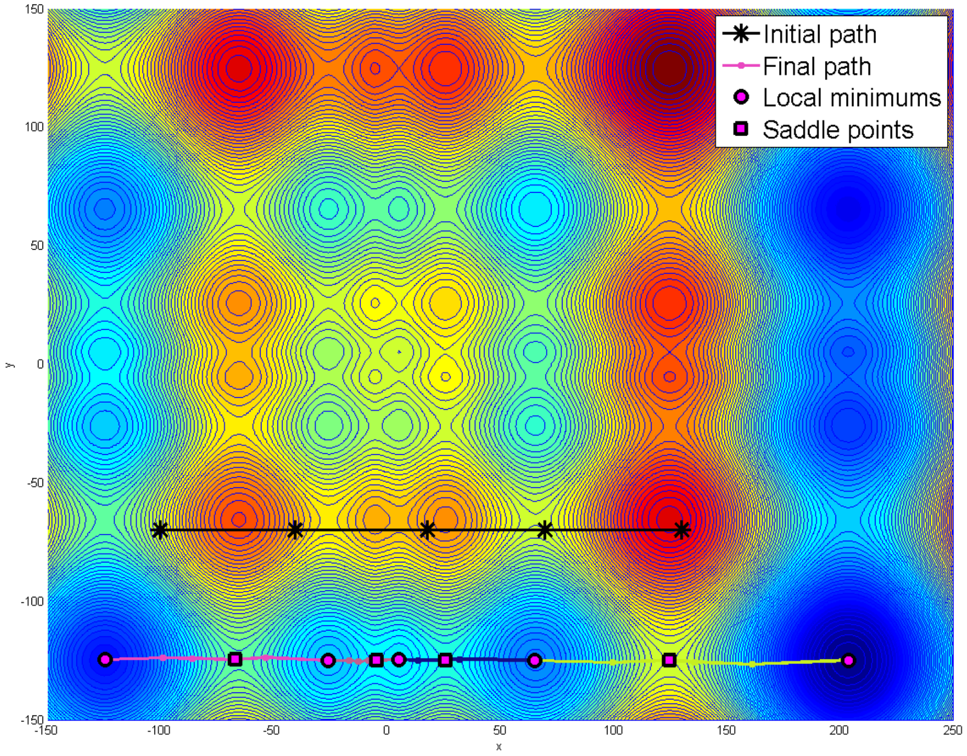


Figure 12: Test result for Schwefel function with the initial position at (−100.3, 25), (−40.5, 25), (17.8, 25), (69.8, 25), and (130.2, 25).

Table 10: Test results on Schwefel function (contour plot refer Figure 12)

Path No	Local minima		Saddle
1	(−124.8274, −124.7472)	(−26.1493, −125.1159)	(−67.1757, −124.7392)
2	(−26.1493, −125.1159)	(5.2426, −124.6550)	(−4.6699, −124.8731)
3	(5.2402, −124.7391)	(65.3225, −124.8080)	(25.8758, −124.8290)
4	(65.4076, −125.0250)	(203.7606, −124.8008)	(124.7793, −124.8429)

3.4.5 Test Results for FeTi-H Transition

FeTi experiences transition from a body-centric structure to an orthorhombic state where it can hold two hydrogen (H) atoms. FeTi has CsC1-type structure which corresponds to $Pm\bar{3}m$ space group with a lattice parameter of 2.9789 Å. Figure 14 (a) shows four unit cells of the FeTi structure at its initial state. The unit cell of FeTi is a body-centered where the Ti atoms are at the center and Fe atoms at the corners. Figure 14 (b) shows one of the possible final states when two H atoms are absorbed in each unit cell forming the structure of FeTiH. The unit cell of FeTiH is an orthorhombic structure of dimension $a = 2.956$ Å, $b = 4.543$ Å, and $c = 4.388$ Å. The Fe atoms occupy the corner, and the center position of the front and back face. The Ti atoms sits at the center of the rest of faces and the H atoms are located on two side faces. Notice that Figure 14(b) shows two unit cells of FeTiH, which correspond to four unit cells of FeTi.

3.4.5.1 Computational Details

Since searching the saddle point of the transition process where H atoms are absorbed, requires us to have the same number of atoms in a unit cell, H atoms are introduced into the body centered cubic FeTi structure to match the final FeTiH structure. As shown in Figure 14(b), there are two basis atoms for each type of Fe, Ti, and H in one unit cell of FeTiH as the final structures. Correspondingly, for two unit cell of the body-centered FeTi, there are two Fe atoms and two Ti atoms as the basis of the initial structures, in addition to the two H atoms. There are multiple adsorption sites of H atoms onto the free surface of Iron-Titanium alloy FeTi. Here, we study four potential adsorption sites of H atoms which are a) on the Fe-Fe bridge along the first primitive lattice vector, b) on the Fe-Fe bridge along the third primitive lattice vector, c) at the center of the (001) surface, and (d) at the center of the (010) surface. The corresponding initial structures are shown in Figure 15. It important to mention that the size of the unit cell for the initial structures is also set to be the size of the final structure, where a meta-stable structure is likely to form. For the purpose of convenient reference to the four transition paths in the following paragraphs, we refer the transition processes from the initial structures in Figure 15(a), (b), (c), and (d) to the final structure (i.e. β phase FeTiH) as FeTiH_tran_a, FeTiH_tran_b, FeTiH_tran_c, and FeTiH_tran_d, respectively.

In this study, the total energy of the system and the forces on each atom are calculated based on density functional theory (DFT) calculation using Vienna Ab-initio Simulation Package (VASP) [203, 204]. The projector augmented wave potentials [205, 206], specifically LDA potentials [207], are used here. The convergence test for the k-point sampling with respect to the total energy shows that 26×13×13 gamma centered grid of k-point sampling is adequate for study of FeTiH structure. Here, to reduce computational time, we use a k-point sampling of 4×2×2 for all the four scenarios. Since the algorithm is implemented in Matlab, an interface between the Matlab and VASP is developed as shown in Figure 13.

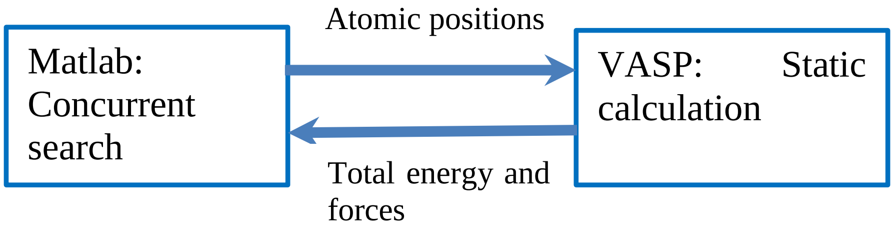
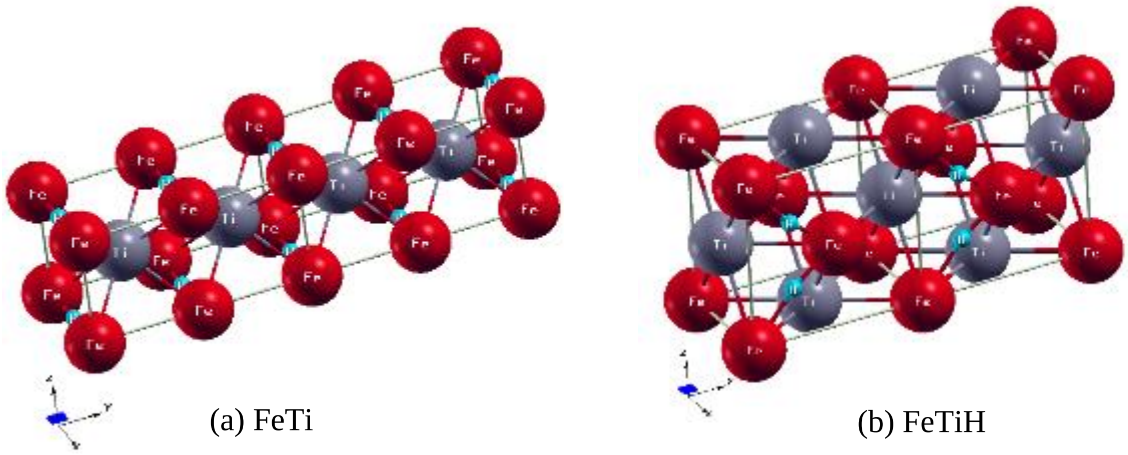


Figure 13: Illustration of the interface between Matlab and VASP

The fractional coordinates for the four initial structures and the final structure are listed in Table 11 and Table 12 respectively. The unit cell for both initial and final structures are defined as $a = 2.956$ Å, $b = 4.543$ Å, and $c = 4.388$ Å. The coordinate of the intermediate images are obtained by linear interpolation of the coordinate of the initial and final images. To demonstrate the algorithm’s capability to locate stable states, we slightly shift the hydrogen atoms from the equilibrium positions which are shown in Figure 15.



(a) FeTi

(b) FeTiH

Figure 14: Comparison between FeTi and FeTiH

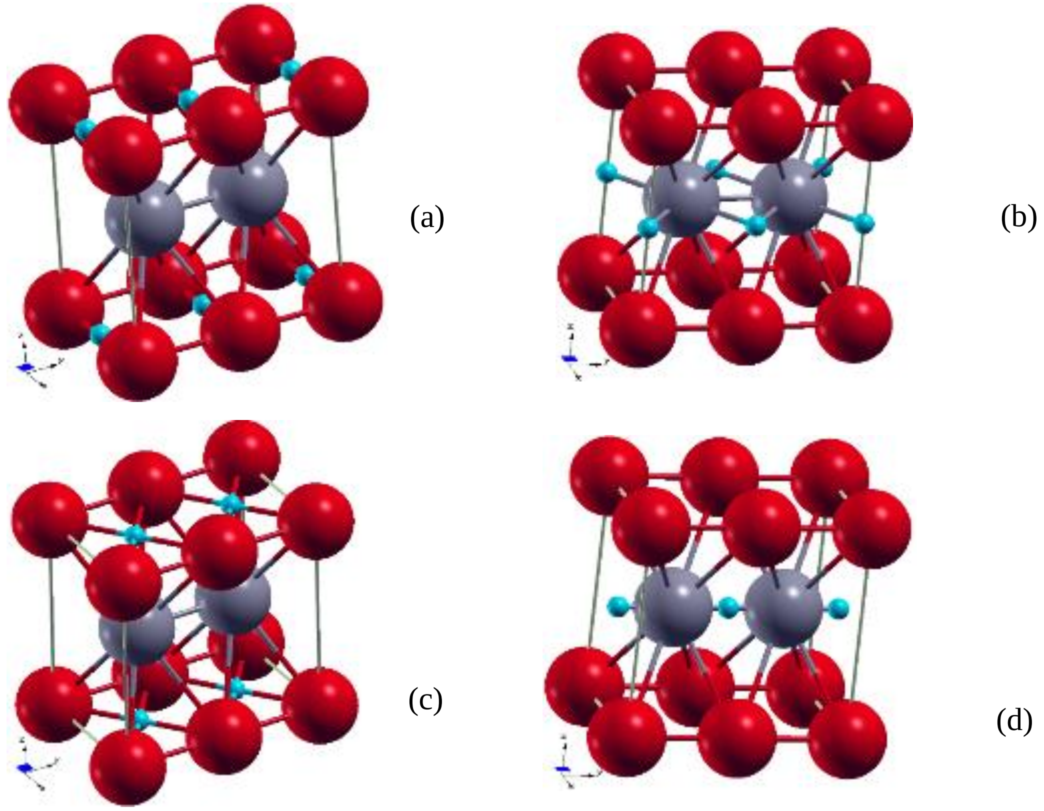


Figure 15: Initial structures of FeTiH with H atoms locating at the (a) Fe-Fe bridge along the first primitive lattice vector (b) Fe-Fe bridge along the third primitive lattice vector (c) center of the (001) surface (d) center of the (010) surface

There are total four initial curves which requires to run the concurrent search algorithm four times separately. The following procedure is applied to each of the initial curves. Firstly, the optimization procedure listed in Table 1 is applied to the initial curve which locates two stable configurations. If there is at least one extra stable configuration between these two stable configurations, the curve is broken into two curve sections. Secondly, the optimization procedure listed in Table 2 is applied to the two curve sections to locate the extra stable configuration. The process continues until all the curves are unbreakable. Thirdly, for each unbreakable curve, the image with highest energy is selected to climb to the saddle position.

Table 11: Fractional coordinates for the four initial structures of FeTiH

		$\hat{a}$	$\hat{b}$	$\hat{c}$			$\hat{a}$	$\hat{b}$	$\hat{c}$
(a)	Fe	0	0	0	(b)	Fe	0	0	0
	Fe	0	0.5	0		Fe	0	0.5	0
	Ti	0.5	0.25	0.5		Ti	0.5	0.25	0.5
	Ti	0.5	0.75	0.5		Ti	0.5	0.75	0.5
	H	0.6	0	0		H	0	0	0.6
	H	0.4	0.5	0		H	0	0.5	0.4
(c)	Fe	0	0	0	(d)	Fe	0	0	0
	Fe	0	0.5	0		Fe	0	0.5	0
	Ti	0.5	0.25	0.5		Ti	0.5	0.25	0.5
	Ti	0.5	0.75	0.5		Ti	0.5	0.75	0.5
	H	0	0	0.6		H	0.45	0	0.55
	H	0	0.5	0.4		H	0.5	0.5	0.55

Table 12: Fractional coordinates for the final structure of FeTiH

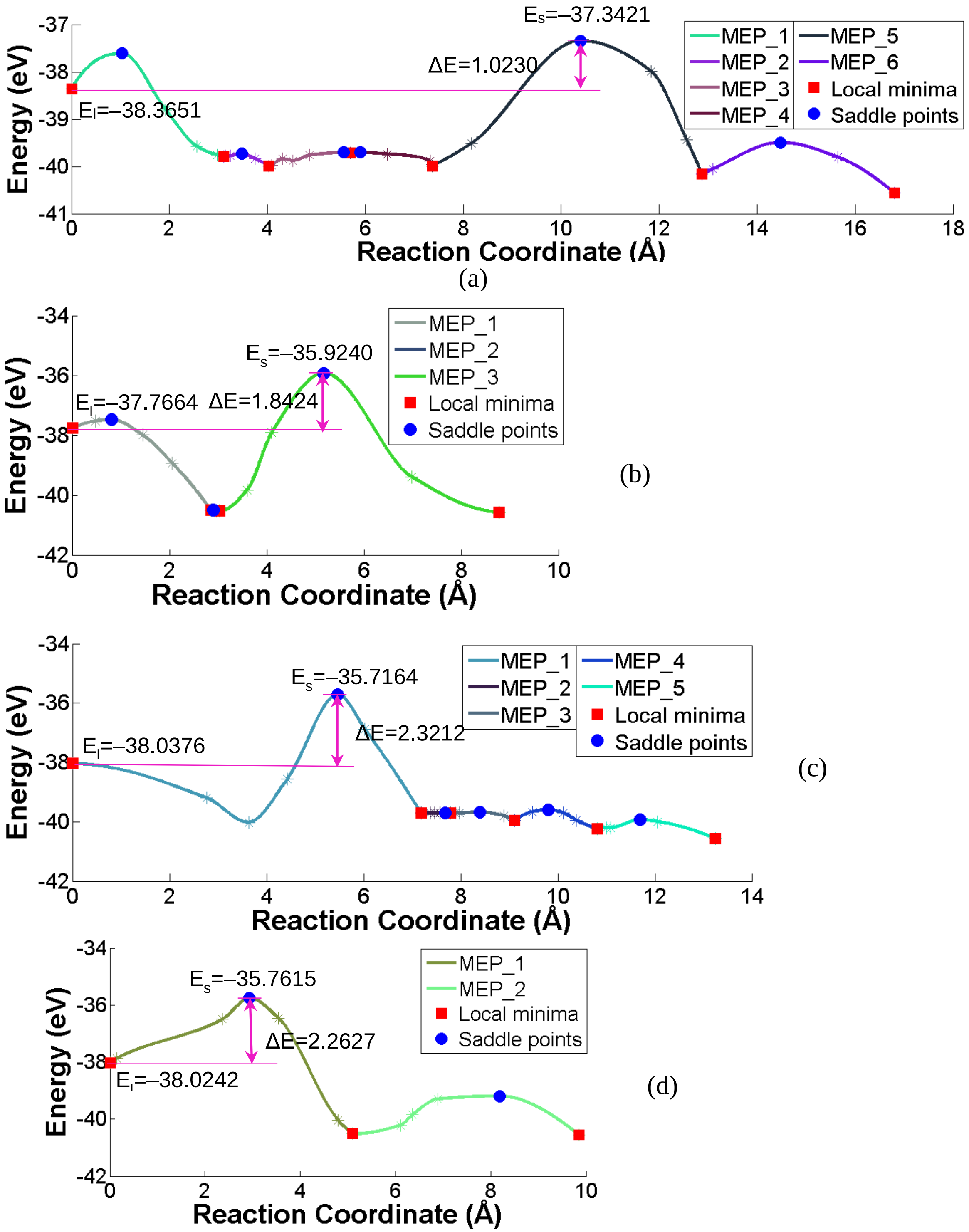
	$\hat{a}$	$\hat{b}$	$\hat{c}$
Fe	0	0	0
Fe	0	0.5	0.5
Ti	0.5	0	0.5
Ti	0.5	0.5	0
H	0	0.25	0.3
H	0	0.25	0.7

3.4.5.2 Computational Results and Disscussion

The located minimum energy paths (MEPs) for the four initial curves are plotted in terms of total energy (eV) with respect to the reaction coordinates which are shown in Figure 16. Here, reaction coordinate which is an abstract one-dimensional coordinate to represent the progress of the atomic configuration along the transition path is used to shown the relative distribution of the images along the path. The E_I and E_s are the total energy at the initial states and transition states (i.e. saddle points with highest energy or global saddle points along the transition path). The ΔE represents the activation energy. The square markers in red are the local minima and the round markers in blue are the saddle points. The asterisk markers are intermediate images along the MEPs. Each curve section in different color represents a sub-stage transition path. For example, in Figure 16(b), there are three sub-stage curves which are shown in blue, dark green and purple. For the first initial curve, the algorithm locates six MEPs with seven local minima and six saddle points which are shown in Figure 16 (a). The activation energy ΔE is the energy difference between the saddle point with highest energy (i.e. global saddle point along the transition path) and the initial state. The global saddle point is the one on the fifth MEP with the total energy of –37.3421 eV. The total energy of the initial structure is –38.3651 eV. Thus the activation energy for the transition from the initial structure with hydrogen atoms locating at the Fe-Fe bridge along the first primitive lattice vector to the final orthorhombic structure (i.e. FeTiH_tran_a) is 1.0230 eV.

Similarly, for the second initial curve, the algorithm locates three MEPs with four local minima and three saddle point which are shown in Figure 16(b). The total energy for the initial structure is –37.7664 eV and for the global saddle point is –35.9240 eV. The activation energy for the transition from the initial structure with hydrogen locating at the Fe-Fe bridge along the third primitive lattice vector to the final orthorhombic structure (i.e. FeTiH_tran_b) is 1.8424 eV. For the third initial curve, the algorithm locates five MEPs with six local minima and five saddle points which are shown in Figure 16(c). The total energy for the initial structure and saddle point is –38.0376 eV and –35.7164 eV. The activation energy for the transition from the initial structure with hydrogen locating at the center of the (001) surface to the final orthorhombic structure (i.e. FeTiH_tran_c) is 2.3212 eV. For the fourth initial curve, the algorithm locates two MEPs with three local minima and two saddle points which are shown in Figure 16(d). The total energy for the initial structure is –38.0242 eV and for the saddle point is –35.7615 eV. The activation energy for the transition from the initial structure with hydrogen locating at the center of the (010) surface to the final orthorhombic structure (i.e. FeTiH_tran_d) is 2.2627 eV. The results are summarized in Table 13. The last column listed the experimental result of the activation energy for the diffusion of hydrogen in the β phase FeTiH which is 1±0.05 eV per H_2 [208].

The transition process from the initial structures (a) with hydrogen locating at the Fe-Fe bridge along the first primitive lattice vector to the final structure has the lowest energy barrier with the activation energy of 1.023 eV. The difference between the calculated activation energy and the experimental one is only 2.3% which is quite acceptable in the saddle point search with first principle DFT calculation. The activation energy for all the other three transitions starting from the three initial states in Figure 15(b-d) are all higher than the experimental one. This is reasonable which can be explained in the following aspects. First, the formation of the β phase FeTiH includes two steps which are 1) adsorption of the free hydrogen onto the free surface of Iron-Titanium alloy FeTi and 2) the hydrogen diffusion in the FeTiH system. The transition processes studied in the dissertation are potential diffusion paths which could happen in step 2 depending on external conditions. The initial structures of these transition processes are the final products of the adsorption process (i.e. step 1). The theoretical study using ab initio calculation shows that there are many possible hydrogen adsorption sites on the free surface of Iron-Titanium alloy and the hydrogen position in the initial structure in Figure 15(a) is one of the favorite adsorption sites [209]. This explains that the transition FeTiH_tran_a has a higher probability of occurring in physical world than the rest three transitions. In other words, the process captured by the experiment will most likely be the transition FeTiH_tran_a. Second, according to the harmonic transition state theory, the lower the activation energy for one transition is, the higher the probability that the transition will occur in physical world. This also explains that the transition in the experiment will be the FeTiH_tran_a. In addition, this indicates that the activation energy of the rest three transitions should be higher than the transition FeTiH_tran_a; otherwise, one of the transition with lowest activation energy should occur during the experiment. In conclusion, the computational results for the hydrogen diffusion process in the dissertation match well with the experimental results as well as theoretical explanations.



(a)

(b)

(c)

(d)

Figure 16: Minimum energy paths obtained by concurrent search algorithm for hydrogen diffusion in the FeTiH structure starting from different initial structures with hydrogen locating at (a) the Fe-Fe bridge along the first primitive lattice vector (b) Fe-Fe bridge along the third primitive lattice vector (c) center of the (001) surface (d) center of the (100) surface, to the same final orthorhombic structure

Figure 17-Figure 19 show the detailed transition paths for each atom in one unit cell along the MEPs in the transition FeTiH_tran_a, FeTiH_tran_b, FeTiH_tran_d, and FeTiH_tran_c, respectively. Since the detailed distribution of the intermediate images can be found in Figure 16, here only the energies are shown in the boxes above the corresponding atomic configurations. The ones with the energy value in bold text are the local minima and the energy value in bold italic text are the saddle points.

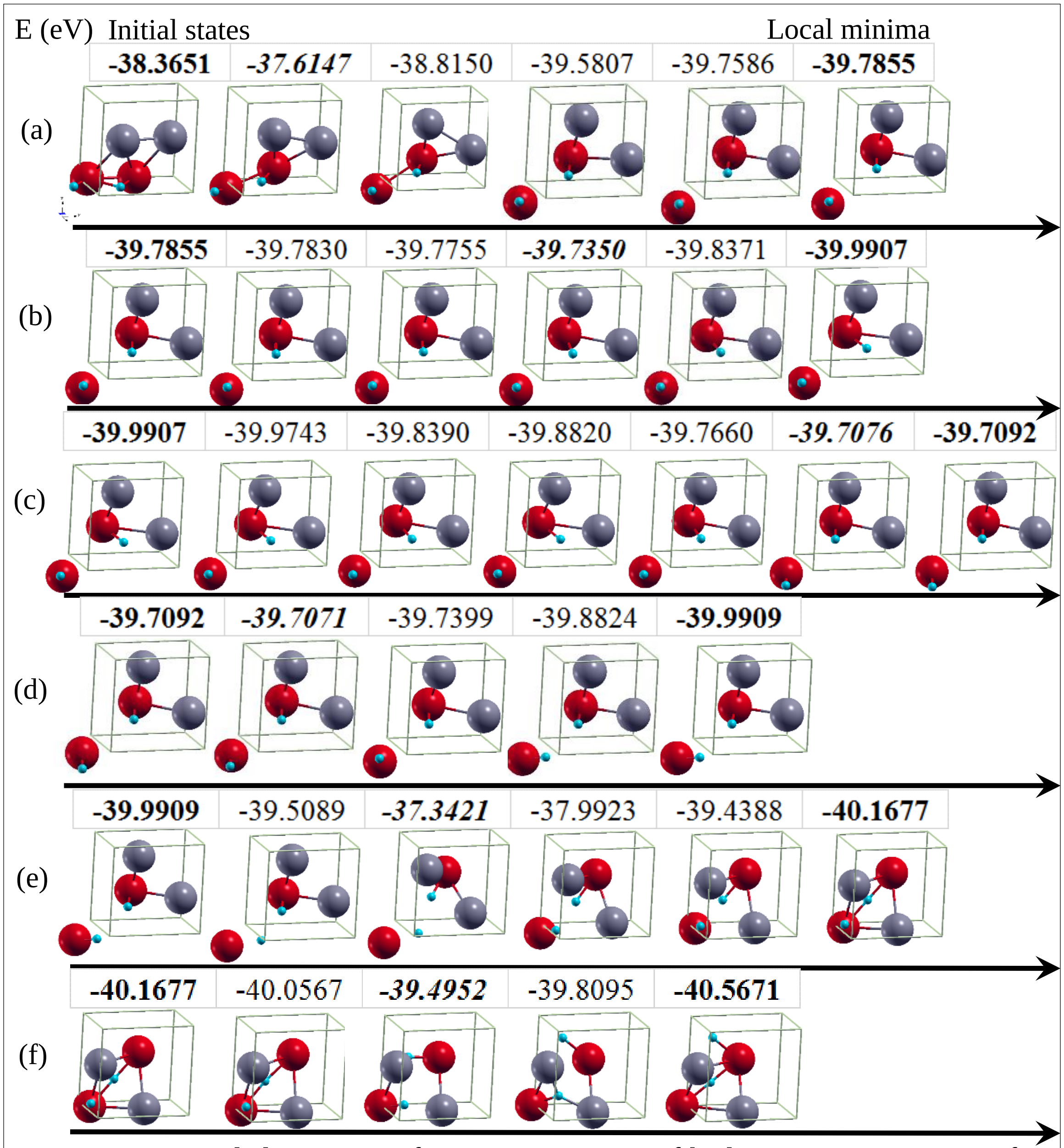


Figure 17: Detailed atomic configuration transition of hydrogen atoms in FeTiH for FeTiH_tran_a (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 (e) MEP_5 (f) MEP_6 obtained by concurrent search algorithm

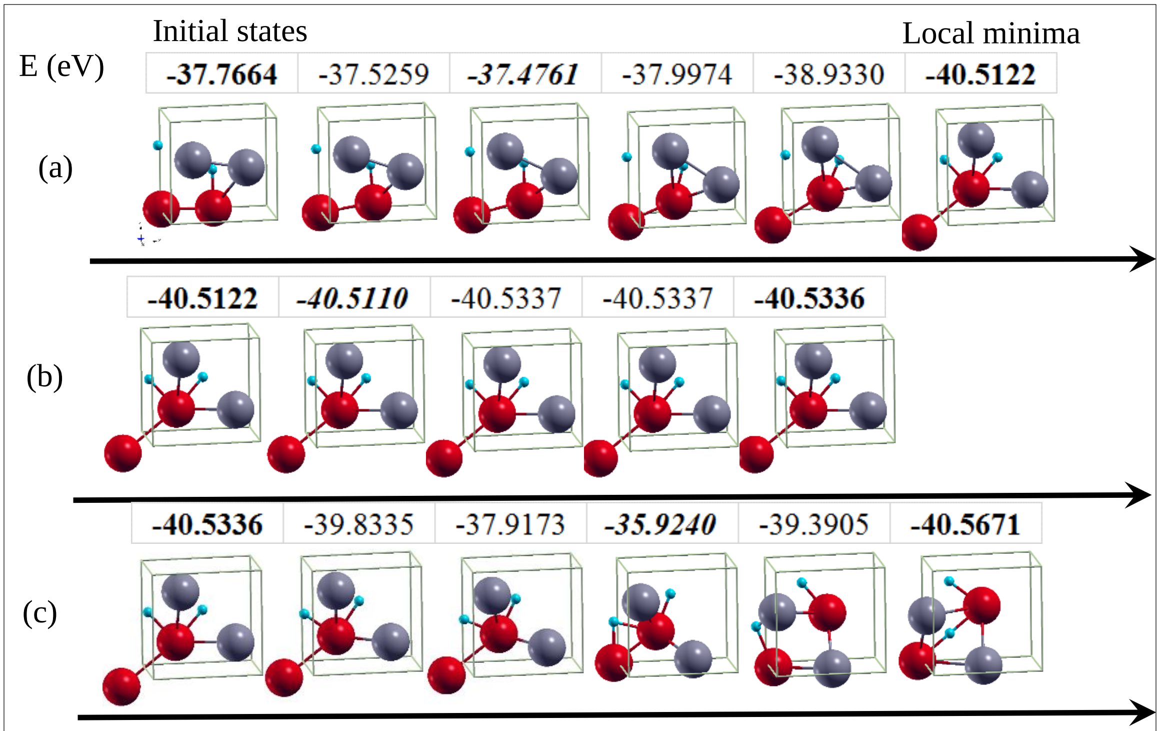


Figure 18: Detailed atomic configuration transition of hydrogen atoms in FeTiH for FeTiH_tran_b (a) MEP_1 (b) MEP_2 (c) MEP_3 obtained by concurrent search algorithm

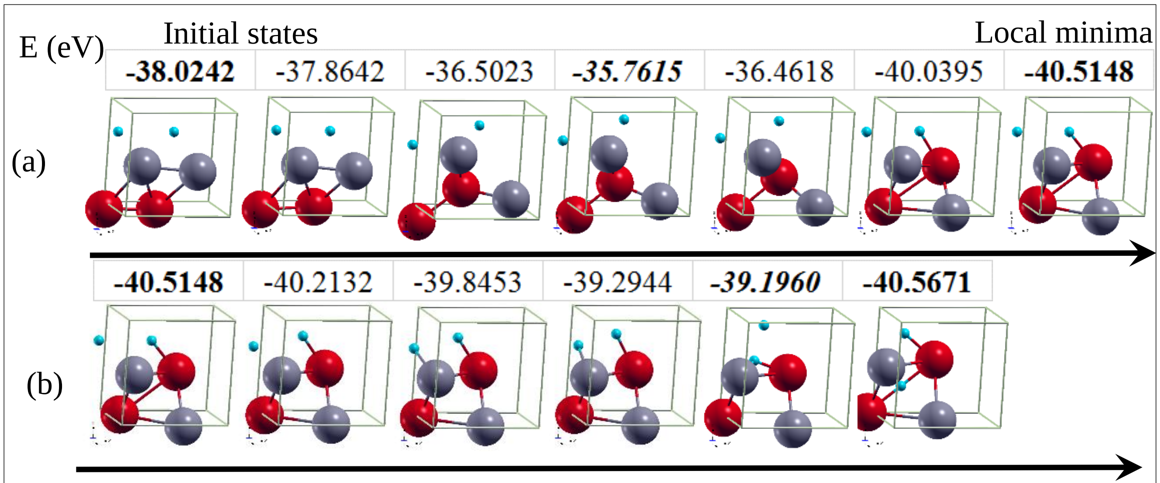


Figure 19: Detailed atomic configuration transition of hydrogen atoms in FeTiH for FeTiH_tran_d (a) MEP_1 (b) MEP_2 obtained by concurrent search algorithm

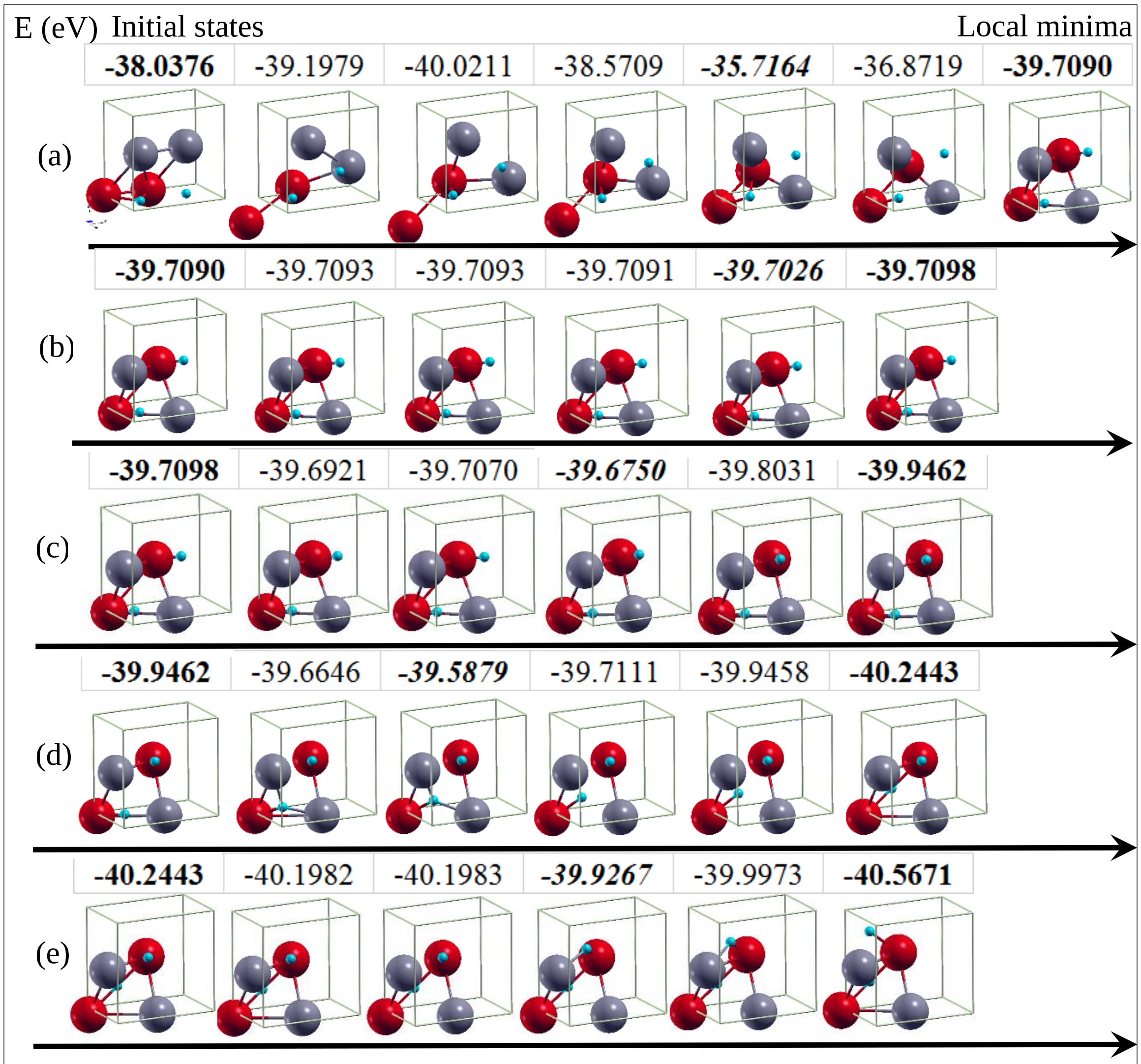


Figure 20: Detailed atomic configuration transition of hydrogen atoms in FeTiH for FeTiH_tran_c (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 (e) MEP_5 obtained by concurrent search algorithm

Table 13: Summary of the results for the concurrent search algorithm on FeTiH system

	# of MEPs	# of local minima	# of saddle points	Energy for initial state E_I (eV)	Energy for saddle point E_s (eV)	Activation energy ΔE (eV)	Experiment ΔE (eV)
FeTiH_tran_a	6	7	6	–38.3651	–37.3421	1.0230	1±0.05
FeTiH_tran_b	3	4	3	–37.7664	–35.9240	1.8424	
FeTiH_tran_c	5	6	5	–38.0376	–35.7164	2.3212	
FeTiH_tran_d	2	3	2	–38.0242	–35.7615	2.2627	

3.4.6 Test Results for Fe_8-H Transition

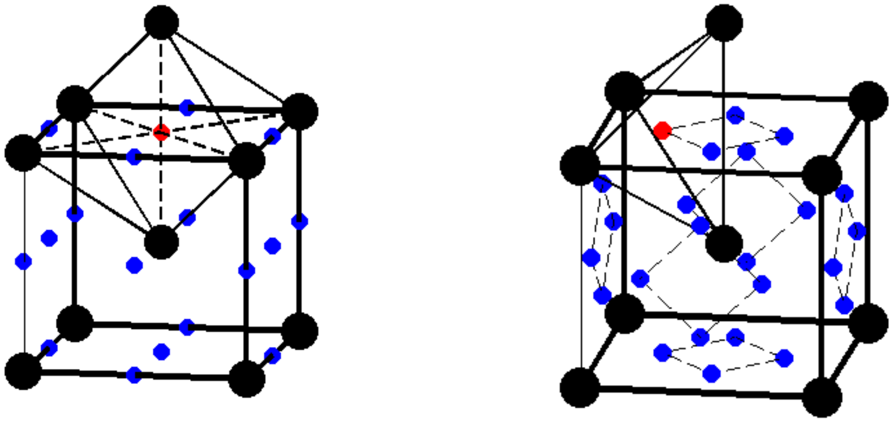
Here we study the diffusion process of hydrogen atoms in pure body-centered iron with a lattice parameter of 2.86 Å. There are two possible sites that H atoms reside in the body-centered iron. One is octahedral site, and the other is tetrahedral site which are shown in Figure 21(a) and (b). The big black dots represent the metal atoms in a body-centered unit cell. The small blue dots in Figure 21(a) are the octahedral site and those in Figure 21(b) are tetrahedral site. Both experimental and theoretical studies are conducted to uncover the favorite site for hydrogen. [210-212] found that hydrogen atoms prefer to reside at tetrahedral site whereas [213] shows that the preferable site is the octahedral site. [214] discovered that there is no preference between these two sites. Here, we use a supercell with four unit cells which include a total of eight Fe atoms. Since it is generally believed that hydrogen has low solubility in body-centered iron, we assume there is only one hydrogen atom in the supercell (Fe_8H). We study the diffusion process of the hydrogen from one octahedral site to the tetrahedral site and two other octahedral sites within the supercell.

3.4.6.1 Experiment Details

The lattice parameter for the unit cell (Fe_8H) is set to be $a = 5.72$ Å, $b = 2.86$ Å, and $c = 5.72$ Å for both initial and final structures. Figure 22 shows the initial structure with hydrogen residing on one of the octahedral site. Figure 23 shows three possible final structures with hydrogen residing on the a) tetrahedral site on the (100) surface (b) octahedral site on the (001) surface (c) octahedral site on the (100) surface. The fractional coordinates for the initial structure and the three final structures are listed in Table 14 and Table 15 respectively. The coordinates of the intermediate images are obtained by linear interpolation of the coordinate of the initial and final states. To demonstrate the algorithm's capability to locate stable states, we slightly shift the hydrogen atoms from the equilibrium positions which are shown in Figure 22 and Figure 23. For the purpose of convenient reference to the four transition paths in the following paragraphs, we refer the transition from the initial structures in Figure 22 to the final structures in Figure 23(a), (b), and (c) as Fe_8H_tran_a, Fe_8H_tran_b, and Fe_8H_tran_c, respectively.

Similar to the study of FeTiH system, the total energy of the system and the forces on each atom are calculated based on density functional theory (DFT) calculation using Vienna Ab-initio Simulation Package (VASP). The projector augmented wave potentials, specifically LDA potentials, are used here. The convergence test for the k-point sampling with respect to the total energy shows that $13\times26\times13$ gamma centered grid of k-point sampling is adequate for study of Fe_8H structure. Here, to reduce the computational time, we use a k-point sampling of $2\times4\times2$ for all the three scenarios.

There are total three initial curves which requires to run the concurrent search algorithm three times separately. Similar to the example of FeTiH, firstly, the optimization procedure listed in Table 1 is applied to the initial curve which locates two stable configurations. If there is at least one extra stable configuration between these two stable configurations, the curve is broken into two curve sections. Secondly, the optimization procedure listed in Table 2 is applied to the two curve sections to locate the extra stable configuration. The process continues until all the curves are unbreakable. Thirdly, for each unbreakable curve, the image with highest energy is selected to climb to the saddle position.



(a) (b)

Figure 21: The interstitial sites in a body-centered lattice with metal atoms (big black dot) (a) Octahedral site (small blue dot) (b) Tetrahedral site (small blue dot)

Table 14: Fractional coordinates for the initial structure of Fe_8H

	$\hat{a}$	$\hat{b}$	$\hat{c}$
Fe	0	0	0
Fe	0.25	0.5	0.25
Fe	0.5	0	0
Fe	0.75	0.5	0.25
Fe	0	0	0.5
Fe	0.25	0.5	0.75
Fe	0.5	0	0.5
Fe	0.75	0.5	0.75
H	0.23	1	0.25

Table 15: Fractional coordinates for the three final structures of Fe_8H

		$\hat{a}$	$\hat{b}$	$\hat{c}$			$\hat{a}$	$\hat{b}$	$\hat{c}$
(a)	Fe	0	0	0	(b)	Fe	0	0	0
	Fe	0.25	0.5	0.25		Fe	0.25	0.5	0.25
	Fe	0.5	0	0		Fe	0.5	0	0
	Fe	0.75	0.5	0.25		Fe	0.75	0.5	0.25
	Fe	0	0	0.5		Fe	0	0	0.5
	Fe	0.25	0.5	0.75		Fe	0.25	0.5	0.75
	Fe	0.5	0	0.5		Fe	0.5	0	0.5
	Fe	0.75	0.5	0.75		Fe	0.75	0.5	0.75
	H	0.63	1	0.125		H	0.25	0.48	1
(c)	Fe	0	0	0					
	Fe	0.25	0.5	0.25					
	Fe	0.5	0	0					
	Fe	0.75	0.5	0.25					
	Fe	0	0	0.5					
	Fe	0.25	0.5	0.75					
	Fe	0.5	0	0.5					
	Fe	0.75	0.5	0.75					
	H	0.78	0	0.75					

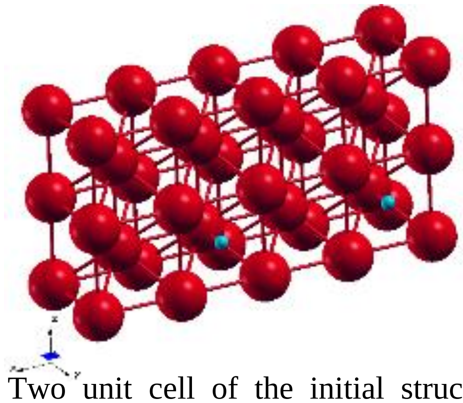


Figure 22: Two unit cell of the initial structure with hydrogen residing on the octahedral site

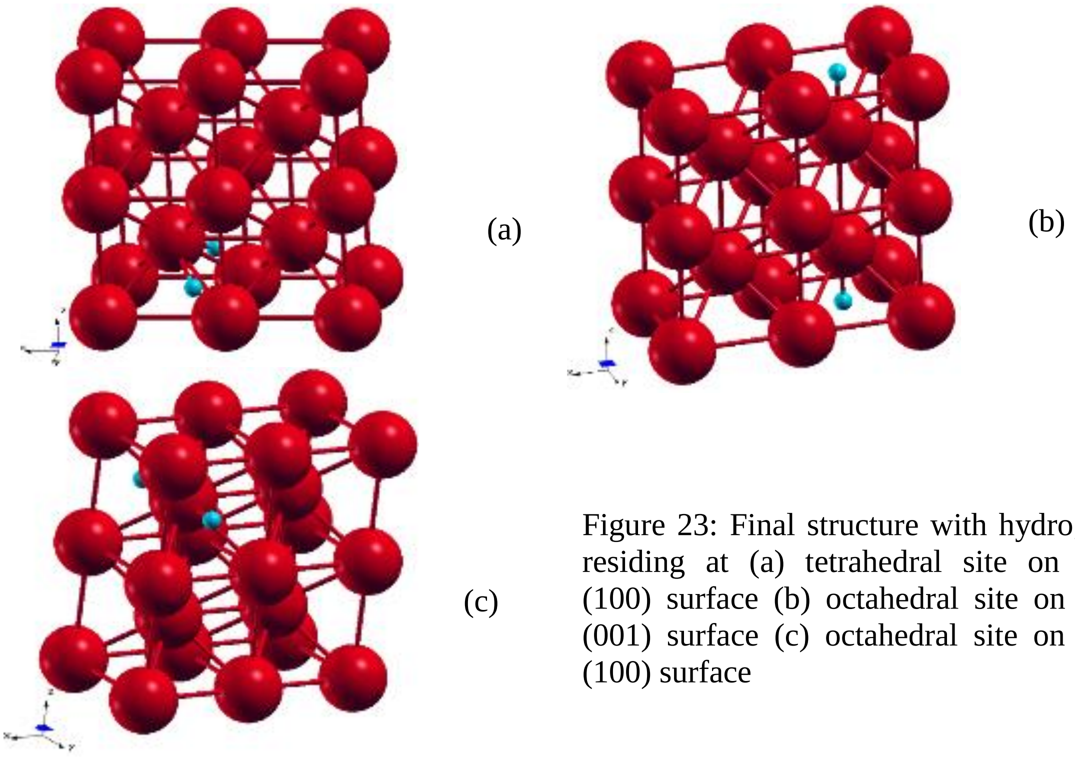


Figure 23: Final structure with hydrogen residing at (a) tetrahedral site on the (100) surface (b) octahedral site on the (001) surface (c) octahedral site on the (100) surface

3.4.6.2 Experiment Results and Discussion

The located minimum energy paths (MEPs) for the three initial curves are plotted in terms of total energy (eV) with respect to the reaction coordinates which are shown in Figure 24. Similarly, the E_I and E_s are the total energy at the initial state and transition states (i.e. saddle points with highest energy or global saddle points along the transition path). The ΔE represents the activation energy. The square marker in red are the local minima and the round marker in blue are the saddle points. The asterisk markers are intermediate images along the MEPs. Each curve section in different color represents a sub-stage transition path. For the first initial curve, the algorithm locates three MEPs with four local minima and three saddle points which are shown in Figure 24(a). The global saddle point is the one on the first MEP with the total energy of –74.7253 eV. The total energy of the initial structure is –75.0992 eV. Thus the activation energy for the transition from the initial structure to the final structure with hydrogen residing at tetrahedral site on the (100) surface (i.e. Fe_8H_tran_a) is 0.3739 eV.

For the second initial curve, the algorithm locates five MEPs with six local minima and five saddle points which are shown in Figure 24(b). The total energy for the initial structure is –75.0992 eV. The global saddle point is the one on the first MEP with the total energy of –74.8023 eV. The activation energy for the transition from the initial structure to the final structure with hydrogen residing on the octahedral site on the (001) surface (i.e. Fe_8H_tran_b) is 0.2969 eV. For the third initial curve, the algorithm locates five MEPs with six local minima and five saddle points which are shown in Figure 24(c). The total energy for the initial structure is –75.0992 eV. The global 'saddle point' is the one one the third MEPs with the total energy of –74.7950 eV. The activation energy for the transition from the initial structure to the final structure with hydrogen residing at the octahedral site on the (100) surface (i.e. Fe_8H_tran_c) is 0.3042. The results are summarized in Table 13.

The experimental results for the activation energy of the hydrogen diffusion in the iron is significantly affected by the impurities of the iron used for the study. Thus, the activation energy from the experiments conducted by ten research groups around the world are scattered from 0.035 eV to 0.142 eV, which are included in [215] by Hayashi and Shu. The results for all three transition are out of the range with a higher activation energy which does make sense since the algorithm might locates the transition paths with higher activation energy. The detailed explaination can be found in Section 3.4.5.2.

Figure 25-Figure 27 give the detailed transition paths for each atom in one unit cell along the MEPs in the transition Fe_8H_tran_a, Fe_8H_tran_b, Fe_8H_tran_c respectively.

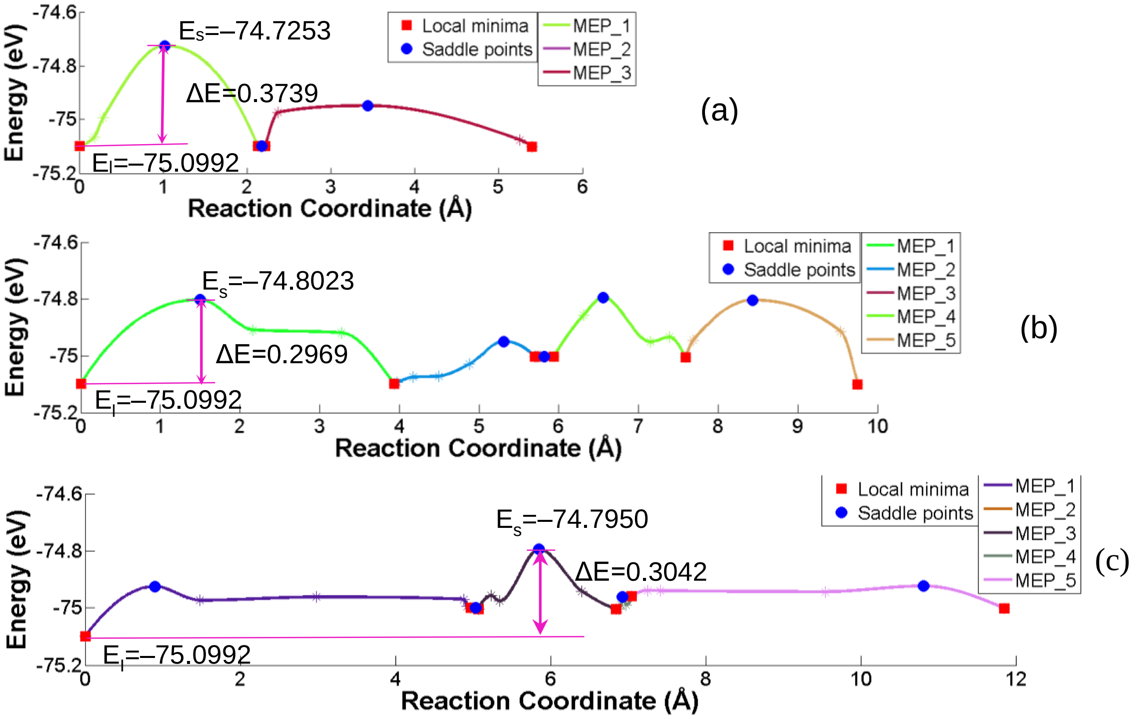


Figure 24: Minimum energy paths obtained by concurrent search algorithm for hydrogen diffusion in Fe_8H structure starting from the initial structure to final structures with hydrogen residing at (a) tetrahedral site on the (100) surface (b) octahedral site on the (001) surface (c) octahedral site on the (100) surface

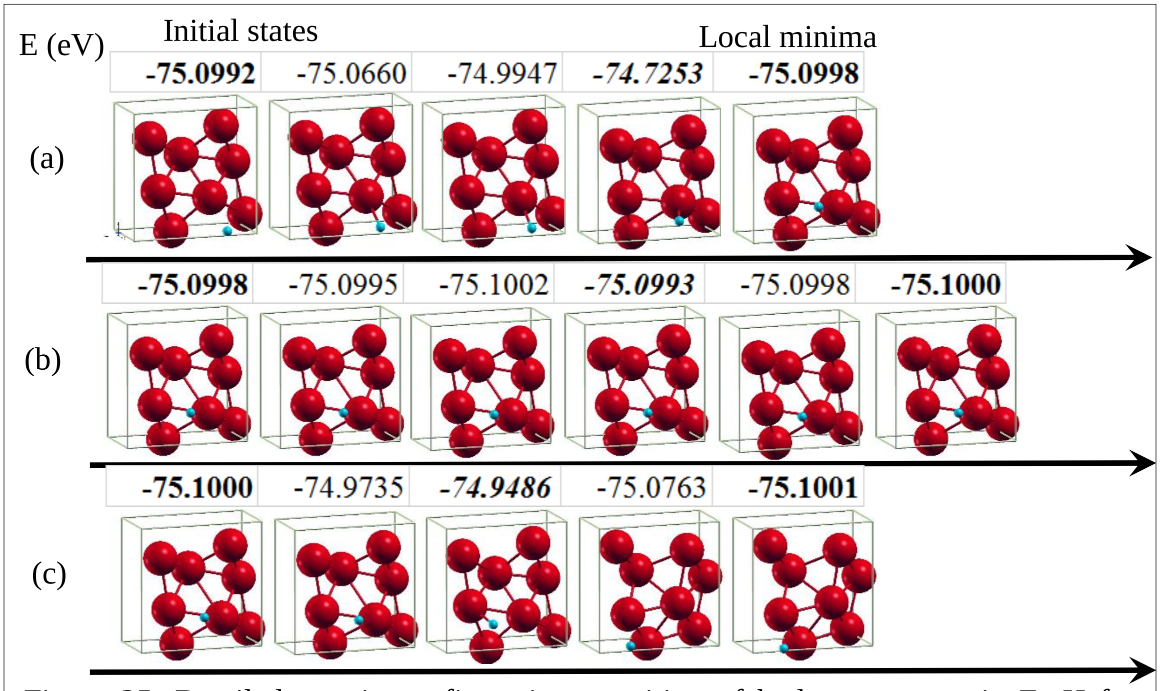


Figure 25: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_a (a) MEP_1 (b) MEP_2 (c) MEP_3 obtained by concurrent search algorithm

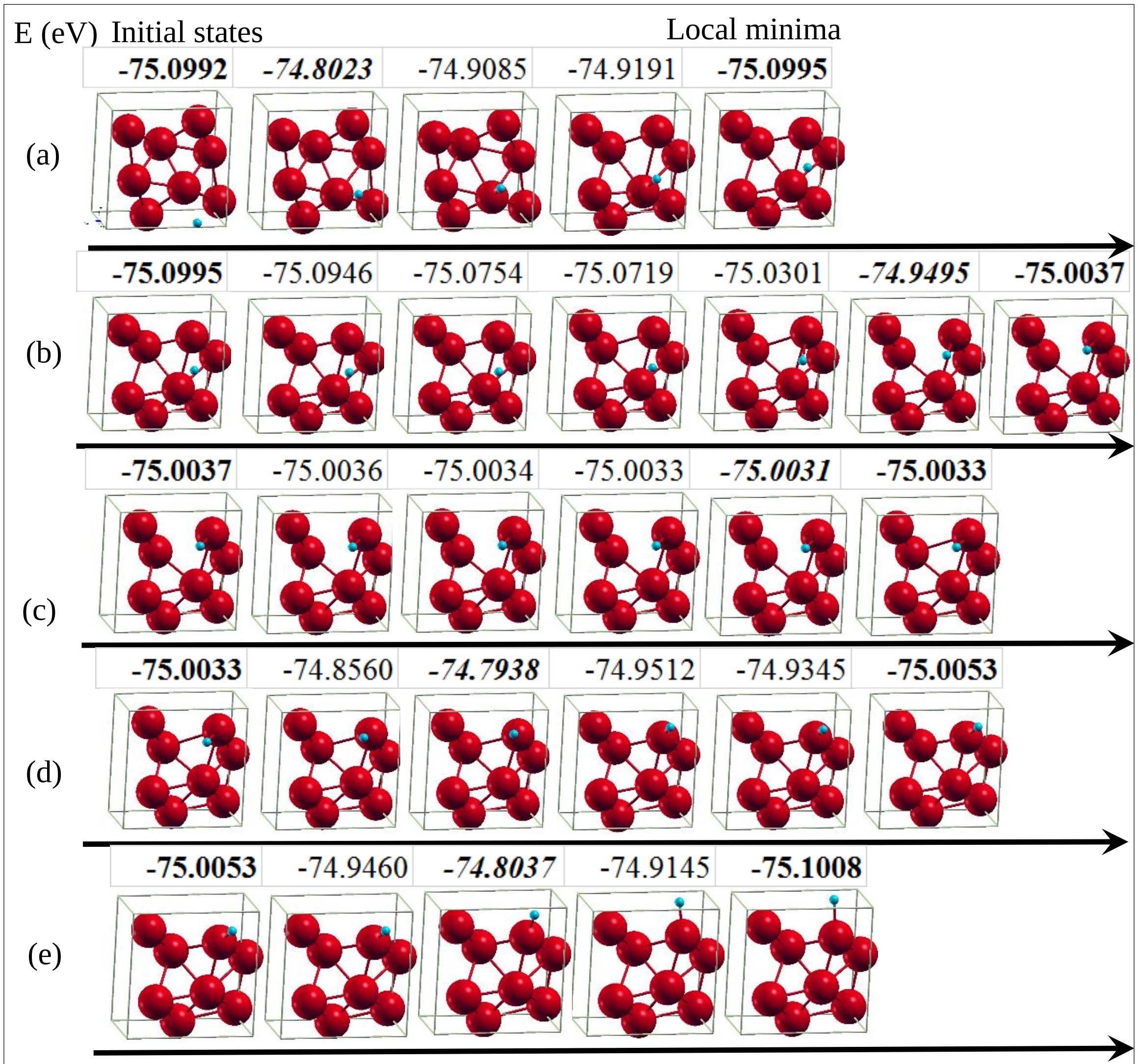


Figure 26: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_b (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 (e) MEP_5 obtained by concurrent search algorithm

Table 16: Summary of the results on Fe_8H system using concurrent search algoirthm

	# of MEPs	# of local minima	# of saddle points	Energy for initial state E_I (eV)	Energy for saddle point E_s (eV)	Activation energy ΔE (eV)	Experiment ΔE (eV)
Fe_8H_tran_a	3	4	3	–75.0992	–74.7253	0.3739	0.035~0.142
Fe_8H_tran_b	5	6	5	–75.0992	–74.8023	0.2969	
Fe_8H_tran_c	5	6	5	–75.0992	–74.7950	0.3042	

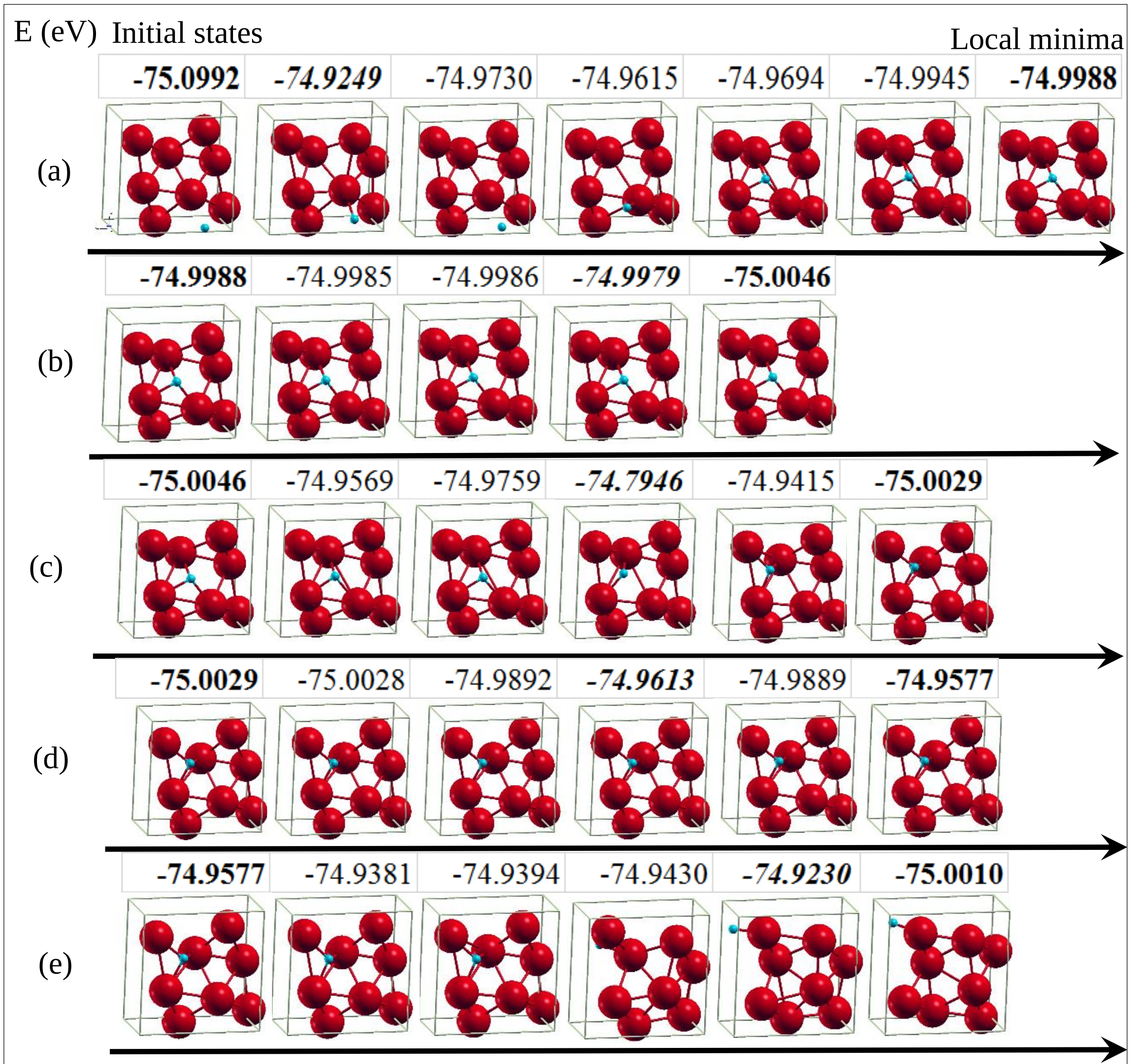


Figure 27: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_c (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 (e) MEP_5 obtained by concurrent search algorithm

3.5 Discussion

Section 3.4 demonstrates that the method is able to locate multiple local minima and saddle points along one transition path. Here, we give a brief discussion on the rate of convergence and limitations of the algorithm.

3.5.1 Convergence Analysis

The convergence of the algorithm to the local minima and the MEPs are discussed in this section. In this algorithm, the conjugate gradient method is adopted in the searching the local minimum. For a quadratic function with n variables, the method can guarantee the local minimum will be located in at most n iterations apart from round-off errors. The searching points converge to the local minimum quadratically. For a non-quadratic function with n variables, the searching process is usually iterative rather than n steps. The approximated conjugate directions generated using Eq. (3.3) are the directions corresponding to the local quadratic approximation to the non-quadratic function. The rate of convergence for the non-quadratic function depends on the response to changes in the local quadratic approximation from one iteration to another. When the searching point approaches the local minimum, it converges to the minimum quadratically. Hence, the choice of the initial position of the end points of the Bézier curve is very important. Those initial positions that require the least number of steps to converge to the bottom of the valley, where the local quadratic approximation is accurate, are the best choices for the initial position of the end control points. In addition, the line search method is employed to determine the step size in each conjugate direction. In each conjugate direction, only several limited mini-steps are applied to locate the minimum along that direction in order to reduce the computational cost. In other words, the point we locate by the mini-steps search may not be the minimum point along that direction. This could expand the approximation error in terms of conjugacy. Ultimately it will lead to more steps for the conjugate gradient method to converge. There is a trade-off between the number of functional evaluations during the line search process in each conjugate directions and the number of iterations needed for the conjugate gradient method to converge to the minimum.

3.5.2 Limitations for the Algorithm

As mentioned in section 3.3.2, the curve subdivision scheme will miss some breakable curves in some rare cases. The corresponding remedy is also introduced in the same section. However, this remedy is still not good enough, and it increases the computational cost.

In addition, this algorithm can locate several transition paths along one transition path. There is still possibility that it misses some paths which are important. For example, for two particular stable states, the current algorithm can only locate one transition path which consists of multiple curves with their end points connected together locating at multiple local minima. There could be other possible paths between those two states. Figure 28 gives an example of such cases, where A, B, C, D, and E are stable states. From the initial state A to final state B, there are two possible transition paths in this case. One is A-D-E-B, and the other is A-C-B. Our algorithm can only locate either the transition path A-D-E-B or A-C-B, not both. The ideal case is to locate both the transition paths. As we know, in most of cases we are interested in identifying the minimum energy barrier between two states. If the algorithm can only locate one transition path, this could lead to an overestimation of the minimum energy barrier between the states A and B because the identified path could be the one with the energy barrier higher than the minimum energy barrier. For example, if the path A-C-B is the one with minimum energy barrier between states A and B. The algorithm may locate the path A-D-E-B instead of A-C-B. As a result, the minimum energy barrier between states A and B will be overestimated.

The ideal case is to locate all the paths on the PES which gives us a better overview of the landscape of the PES. Once all the local minima as well as corresponding saddle points are identified, it is easy to locate the transition path that requires the least energy to transit from reactant to product. As a result, we can estimate the minimum energy barrier between two states accurately. Several trials with different initial guesses for the transition path are needed in order to locate all the transition paths on the PES. This trial procedure is computationally expensive. One trial with different initial guesses from previous trials may locate some transition paths which are already identified by the previous trials, since for each trial they can converge to several transition paths. In addition, in real applications, we actually do not know the PES in advance. As a result, it is difficult to determine whether all the transition paths are located or not. To solve this problem, we develop the curve swarm algorithm to exhaustively search the local minima and saddle points within a searching area simultaneously. The algorithm is presented in the next chapter.

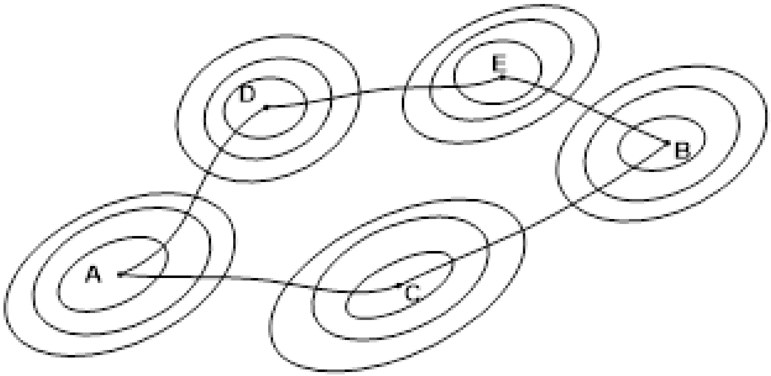


Figure 28: Illustration for two possible transition paths between two states

CHAPTER IV

A CURVE SWARM SEARCH ALGORITH FOR GLOBAL SEARCH OF PHASE TRANSTION PATHS

Previously, we developed a concurrent search algorithm to search multiple local minima and saddle points along one transition path. This improves the accuracy of activation energy estimation along one transition path. However, for complicated potential energy surfaces (PES's), the transition from one stable configuration to another can be done through more than one path. The lowest energy path (i.e. MEP) between two states should be located. Since the existing saddle point search methods are local methods, the result sensitively depends on the initial guess of the transition path. The path identified by those methods may not be the MEP. Consequently, this will lead to an overestimation of the energy barrier between two states. To solve this problem and provide a global view of the landscape of the PES, we developed the curve swarm search algorithm to exhaustively locate the local minima and saddle points on a PES within a searching area. In this chapter, we present the curve swarm search algorithm and demonstrate it by examples of Rastrigin function and Schwefel function. In addition, we applied the algorithm to study the diffusion process of the hydrogen in Iron Titantium alloy (FeTi) and pure Iron (Fe). The algorithm represents a transition path by a parametric Bézier curve with control points. It uses multiple groups of such curves, each of which represents a multi-stage transition path. During the searching process, each group of curves communicates with others to maintain cohesion and avoid collision based on a collective potential model. For each curve in one group, its two end control points are minimized to locate two local minima, while intermediate control points move along their corresponding conjugate directions to refine the shape of the curve. If extra minima are found along the curve, it is broken into multiple curve segments to form a multi-stage transition path.

4.1 Overview for the Curve Swarm Search Algorithm

The curve swarm algorithm uses several groups of curves to try to exhaustively search local minima and saddle points within one area on a PES. The algorithm represents the transition path using a parametric Bézier curve with control points. It uses multiple groups of such curves, each of which represents a multi-stage transition path. At the initial stage, each group includes only one curve representing one initial transition path. The curve in one group will gradually be divided into multiple curves to locate multiple local minima and saddle points along the transition path. At the final stage, each group includes multiple curves with their end points connected together and located at multiple local minima. In addition, each curve has one intermediate point located at the saddle point position. The central issue in the curve swarm algorithm is to maintain cohesion and to avoid collision between groups during the searching process. Cohesion means that groups should stay relatively close to each other to explore a PES thoroughly, thus locating all the local minima and saddle points. If two adjacent groups stay too far away from each other, some intermediate space between them may not be explored. Collision means that more than one group search the same area on a PES. The algorithm should avoid collision to prevent repetitive efforts, thus maintaining a global view and reducing computational cost.

To maintain cohesion and avoid collision between groups, in this dissertation, we introduce a collective potential model to describe the collective behavior among groups. In other words, we introduce collective forces that could be either attractive or repulsive between groups. The forces are applied to control points in each group. If two curves are too close to each other, the repulsive force is applied to the control points of the curves; otherwise, attractive force is applied.

The procedure to search local minima and saddle points within each group includes three stages that are 1) a single transition pathway search, 2) multiple transition pathway search, and 3) climbing process to locate the saddle position. In the first stage, the algorithm minimizes the two end control points to locate two local minima using conjugate gradient method, while the shape of the curve is refined by moving intermediate control points along the direction determined by both collective force and the parallel component of the potential force in the corresponding conjugate directions of the control point. As long as the curve identifies two local minima, the algorithm moves into the second stage in which the algorithm examines whether the curve crosses an extra energy basin based on curve subdivision scheme. If the curve crosses an extra energy basin, the algorithm breaks the curve into two curve sections at one break point. This point is then minimized using the conjugate gradient method to locate the minimum for the extra energy basin. At the same time, the shapes of the two curves are refined in the same way as in the first stage. The process continues till each curve crosses only two adjacent energy basins. Then in the third stage, the algorithm selects the point with maximum energy on each curve to climb up to locate the saddle point position. The pseudo code for the algorithm is included in Table 17.

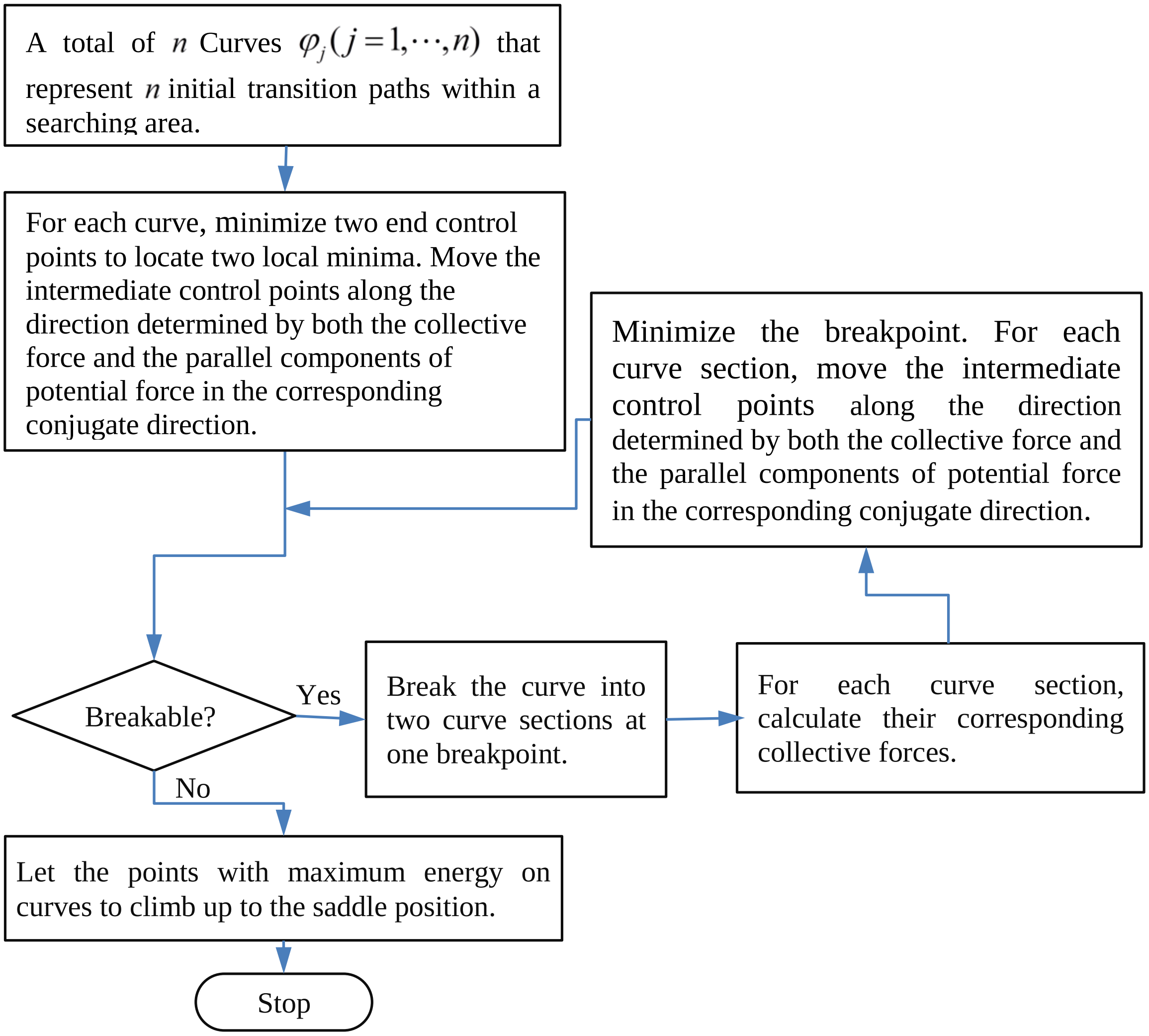

A total of n Curves $\varphi_j(j=1,\cdots,n)$ that represent n initial transition paths within a searching area.
For each curve, minimize two end control points to locate two local minima. Move the intermediate control points along the direction determined by both the collective force and the parallel components of potential force in the corresponding conjugate direction.
Minimize the breakpoint. For each curve section, move the intermediate control points along the direction determined by both the collective force and the parallel components of potential force in the corresponding conjugate direction.
Breakable?
Yes
Break the curve into two curve sections at one breakpoint.
For each curve section, calculate their corresponding collective forces.
No
Let the points with maximum energy on curves to climb up to the saddle position.
Stop

Figure 29: Flow chart of the curve swarm search algorithm

Table 17: Pseudo-code of the curve swarm search algorithm

```
INPUT: A number of initial curves $\varphi_j (j = 1, \cdots, n)$
OUTPUT: The same number of groups of curves. Each group includes multiple curves with their end control points connected together located at multiple local minima. In addition, each curve in one group has one control point located at the corresponding saddle point position.

$M_j^m$ =number of newly produced curve in the $m^{th}$ iteration for the $j^{th}$ group
 WHEN $m = 1$(the first iteration)
DO
Minimize the two end control points for each initial curve $\varphi_j (j = 1, \cdots, n)$ using conjugate gradient method. Move all the intermediate control points in their corresponding conjugate directions with positive eigenvalues.
END
$m = 1$;
WHILE There exists newly produced curves in the $m^{th}$ iteration
$m = m + 1$;
$M_j^m = 0$ ;
FOR $j = 1, \cdots, n$ (the number of total groups)
    FOR $\# = 1, \cdots, M_j^{m-1}$
        IF $\varphi_j^{\#}(\mathbf{x})$ is breakable
            Break the curve $\varphi_j^{\#}(\mathbf{x})$ into two curves $\varphi_j^{\#,1}(\mathbf{x})$ and $\varphi_j^{\#,2}(\mathbf{x})$ at one break point.
            $M_j^m = M_j^m + 2$;
        END IF
        IF The number of control points for $\varphi_j^{\#,1}(\mathbf{x})$ or $\varphi_j^{\#,2}(\mathbf{x})$ is less than five
            Do degree elevation to the curve $\varphi_j^{\#,1}(\mathbf{x})$ or $\varphi_j^{\#,2}(\mathbf{x})$.
        END IF
        Calculate the collective forces for the newly created curves.
        Minimize the break point using conjugate gradient method, while the intermediate control points for curve $\varphi_j^{\#,1}(\mathbf{x})$ and $\varphi_j^{\#,2}(\mathbf{x})$ moves along the directions defined by their corresponding conjugate directions and the collective forces from other curves
    END FOR
END FOR
END WHILE
FOR $u = 1, 2, \cdots,$ (total number of non-breakable curves produced during the WHILE loop
Do
Select the control point with maximum energy along each non-breakable curve to climb up to locate the corresponding saddle point.
END FOR
```

4.2 A Single Transition Pathway Search in Curve Swarm Algorithm

For the procedure at the first stage (i.e. a single transition pathway search), the difference between the concurrent search algorithm and curve swarm algorithm is the way to refine the shape of the curve. In the concurrent search algorithm, the shape of the curve is refined by moving the intermediate control points along the conjugate directions subject to only potential forces (i.e. the gradient of the PES at the intermediate control points), whereas in the curve swarm algorithm, the intermediate control points are subject to both the collective force from their neighbors and the potential force. In each iteration, a set of conjugate directions are constructed for each intermediate control points using Eqn.(3.5). Then the intermediate control points move along the directions determined by both the collective force and the parallel components of potential force in the corresponding conjugate direction. For example, for the k^{th} intermediate control point on a curve φ that needs to be optimized, the search direction associated to its i^{th} 'conjugate direction' is defined as

$$\mathbf{D}_{k,\varphi}^{(i)} = w_g(-\nabla V(\mathbf{x}_{k,\varphi}^{(i-1)}) \cdot \mathbf{s}_{k,\varphi}^{(i)} \mathbf{s}_{k,\varphi}^{(i)}) + w_c \nabla V_{k,\varphi}^{C} \tag{4.1}$$

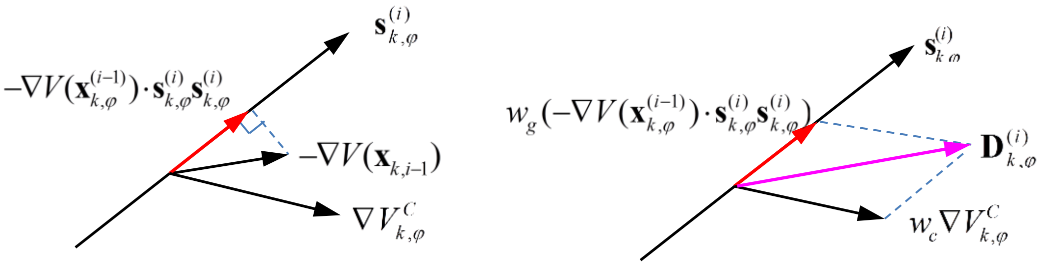
in which $\mathbf{x}_{k,\varphi}^{(i-1)}$ is the position for the k^{th} intermediate control point along i^{th} search direction $\mathbf{D}_{k,\varphi}^{(i)}$. $V_{k,\varphi}^{C}$ is the collective potential for the k^{th} point on the curve φ; w_g and w_c are the weights for the parallel component of the true potential force and collective force $\nabla V_{k,\varphi}^{C}$ respectively. The definition of the search direction $\mathbf{D}_{k,\varphi}^{(i)}$ is illustrated in Figure 30. In step (a), the true potential force is projected to the conjugate direction $\mathbf{s}_{k,\varphi}^{(i)}$ to obtain its parallel component. Then in step (b), the search direction is obtained by adding the weighted parallel component of the true potential force and the weighted collective force of the curve.

The collective potential for the k^{th} intermediate control point on a curve φ is defined as a weighted sum of the pair-wise potential between the control point and the curves in its neighbor groups. That is

$$V_{k,\varphi}^{C} = \sum_{l=1}^{L} w_l \psi(r_{kl}) \tag{4.2}$$

in which w_l is the weight for the l^{th} ($l=1\cdots L$) curve from its neighbor groups; L is the total number of curves from its neighbor groups; ψ is the pairwise potential between two points; r_{kl} is the approximate distance from the k^{th} point to the l^{th} curve in its neighbor groups. The detailed definition for r_{kl} is given in Section 4.3.2.

The key issue here is to model the pairwise potential that is able to maintain cohesion and avoid collision between two curves. The following section gives a detailed description on the pairwise potential model.


$\mathbf{s}^{(i)}_{k,\varphi}$
$-\nabla V(\mathbf{x}^{(i-1)}_{k,\varphi})\cdot\mathbf{s}^{(i)}_{k,\varphi}\mathbf{s}^{(i)}_{k,\varphi}$
$-\nabla V(\mathbf{x}_{k,i-1})$
$\nabla V^{C}_{k,\varphi}$
$\mathbf{s}^{(i)}_{k,\varphi}$
$w_g(-\nabla V(\mathbf{x}^{(i-1)}_{k,\varphi})\cdot\mathbf{s}^{(i)}_{k,\varphi}\mathbf{s}^{(i)}_{k,\varphi})$
$\mathbf{D}^{(i)}_{k,\varphi}$
$w_c\nabla V^{C}_{k,\varphi}$

Step (a)

Step (b)

Figure 30: An illustration of the definition of the searching direction for the intermediate control points in curve swarm search algorithm

4.3 Euclidean Distance Based Collective Potential Model

As stated in Section 4.1, we introduce a collective potential model to solve the central issue in the curve swarm algorithm that is to maintain cohesion and to avoid collision between groups during the searching process. Since the collective force between two curves could be either attractive or repulsive depending on the distance between two curves, the collective potential model we choose in this dissertation is Lennard-Jones potential that is widely used to describe the pairwise potential between two atoms or molecules. The potential is defined as

$$\psi = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{4.3}$$

in which ε is the depth of the potential well, σ is the characteristic distance at which the pair potential is zero, r is the distance between two atoms or molecules. The central issue here is to define the three parameters to make the model well serve our purpose that is to maintain cohesion and avoid collision among curves.

4.3.1 Parameter Determination

4.3.1.1 The depth of the potential well ε

Since the curves in this algorithm evolve under the forces from both true potential (i.e. energy potential) and imaginary potential (i.e. collective potential among curves), the criteria of determining ε is that the value of imaginary collective potentials are at the same magnitude with the true potential of the system. Also, to capture more information of the potential energy surface, the depth of the potential well is adaptive. Based on this criterion, we define ε as the average value of the potential energy at the two local minima that are identified by the end control points of the breakable curve, as

$$\varepsilon = \frac{E_{endpoint\_1} + E_{endpoint\_2}}{2} \tag{4.4}$$

in which $E_{endpoint\_1}$ and $E_{endpoint\_2}$ are the true potential energy at two local minima identified by the two end control points of the breakable curve which shows in Figure 32. In the algorithm, the depth of the potential well ε is adaptive to precisely capture the information of energy landscape on the fly of the searching process. In other words, for every two newly created curves, to calculate their collective forces with the curves from other groups, the depth of the well is redefined based on the energy level of the two local minima identified by the two end control points for the breakable curve.

4.3.1.2 The Characteristic distance σ

Beside of ε, another important parameter needed to be determined in Lennard-Jones potential is the characteristic distance at which the pair-wise potential is zero. Since movement of the curves are directly controlled by the collective forces, it is difficult to directly define the distance at which the potential between two curves is zero. Here, the characteristic distance is derived by defining a distance r_0 at which the collective force is zero (i.e. zero-force distance). Since the algorithm break one curve at only one breakpoint, here, we define the zero-force distance associated with these two curve sections φ_1 and

φ_2 as the minimum distance between the breakpoint and the corresponding two end control points, as

$$r_0 = \min(\| \mathbf{x}_{breakpoint} - \mathbf{x}_{endpoint\_1} \|, \| \mathbf{x}_{breakpoint} - \mathbf{x}_{endpoint\_2} \|) \tag{4.5}$$

in which $\mathbf{x}_{breakpoint}$, $\mathbf{x}_{endpoint\_1}$, and $\mathbf{x}_{endpoint\_2}$ are the positions for the breakpoint, the first endpoint, and the second endpoint of the breakable curve. The definition of the distance at which the collective force between two curves is zero is illustrated in Figure 32. The reason that we define the zero-force distance as the minimum distance instead of maximum distance between the breakpoint and the corresponding two end control points is given in the following separate paragraph. The characteristic distance is determined by solving

$$\frac{\partial \psi}{\partial r} = 0 \tag{4.6}$$

The solution to the Eqn. (4.6) is $r = 2^{1/6}\sigma$. By replacing r with r_0 in the equation $r = 2^{1/6}\sigma$, the characteristic distance is calculated as

$$\sigma = 0.8909 r_0 \tag{4.7}$$

In the algorithm, the characteristic distance is adaptive to precisely capture the information of energy landscape on the fly of the searching process. In other words, for every two newly created curves, to calculate their collective forces with the curves from other groups, the characteristic distance is updated by using the positions of their breakpoint and the end control points.

Here, we analyze and demonstrate that it is better to define the zero-force distance as the minimum distance between the breakpoint and the corresponding two end control points. The curve subdivision scheme in Section 3.3.1 defines the breakpoint in the way which guarantees that the breakpoint lies within the third energy basin which is different from the two whose local minima are located by the two end control points. In other words, there is no need to apply either attractive force or repulsive force between endpoint and breakpoint. Thus, the distance between breakpoint from both endpoints can be defined as zero-force distance. As shown in Figure 32(a), the zero-force distance r_0 is the critical distance from repulsive force to attractive force. Suppose we define the zero-force distance as the maximum distance between breakpoint and endpoints, the critical location for the Lennard-Jones potential will be shifted to the right. Some curves in the neighborhood that should have attractive force will have repulsive force now. As a result, those fake repulsive force from the curve that is far away will decrease the effect of the repulsive force that is from the closer curves. In addition, it increases the cutoff distance for the attractive force. In other words, some curves that is very far away should not have pairwise force now still have attractive force, which will decrease the effect of the force that from the closer curves. In conclusion, the definition of the zero force distance based on the minimum distance between breakpoint and end points can define the cutoff distance for both repulsive and attractive force more precisely. As a result, it serves better for the purpose of maintaining cohesion and avoiding collision. To verify the above reasoning, experiments were conducted by first defining the zero-force distance as the minimum distance between breakpoint and end points, and then changing the definition of the zero-force distance to be the maximum distance between breakpoint and end points. The comparison of the results is shown in Figure 31.

Figure 31(a) shows the result with the zero-force distance being defined as minimum distance between the breakpoint and the end point, while Figure 31(b) is the result with the distance being defined as maximum distance. In Figure 31(a), when the first and second groups of curves getting too close, the algorithm push them away in a right amount of distance. Also, when the third group of curves is far away from the first two groups, the algorithm pulls them closer to the first two groups. However, in Figure 31(b), the algorithm pushes the first two curves far away which means the magnitude of the collective force is not defined appropriately. For the third group, the algorithm did not pull them closer. Thus, the minimum distance scheme is better than the maximum distance scheme in terms of maintaining cohesion among curves.

After determining the two central parameters for Lennard-Jones potential, the next critical issue is to calculate the collective forces for the curves which is presented in Section 4.3.2.

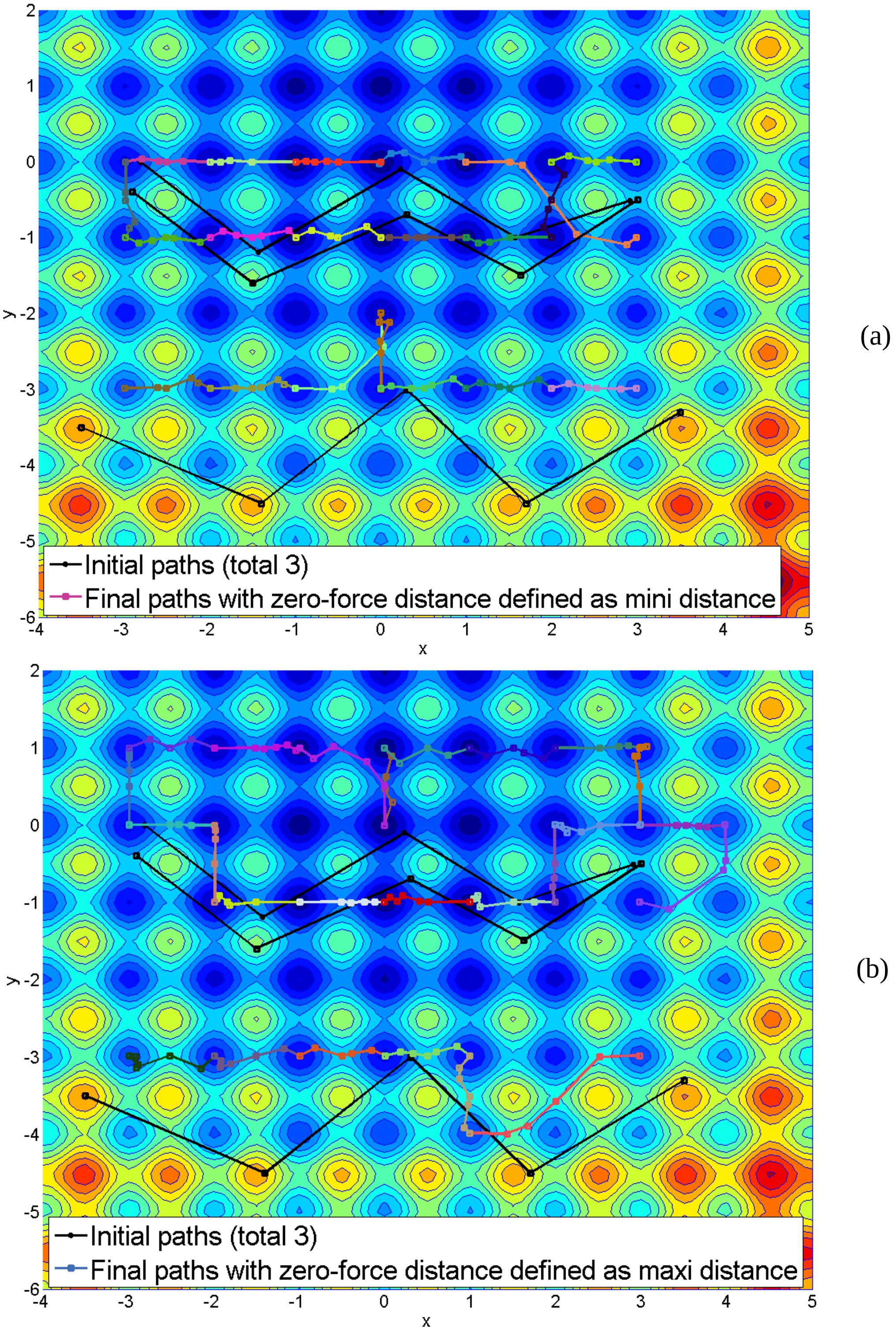
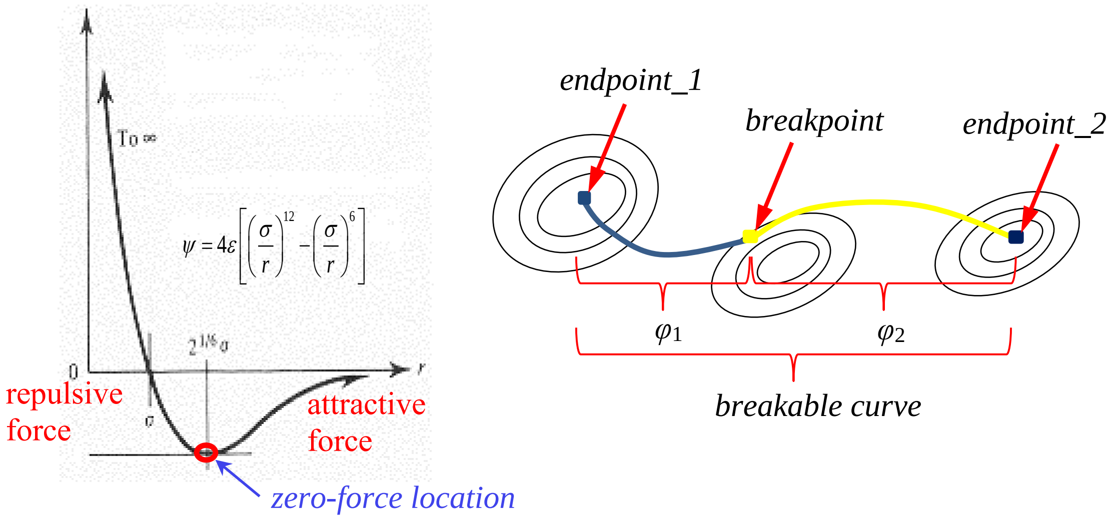


Figure 31: Comparison of the results for two different ways of zero-force distance definition (a) Zero-force distance is defined as the minimum distance between breakpoint and end point (b) Zero-force distance is defined as the maximum distance between breakpoint and end point



(a) Lennard-Jones potential (b) A breakable curve

Figure 32: An illustration of the definition for the distance at which the collective force between two curves is zero

4.3.2 Collective Force

The collective force for one curve is the weighted sum of the pairwise forces from its neighbor curves, which is defined as

$$F(\mathbf{x}) = \sum_{l=1}^{L} w_l f(\mathbf{x}) \tag{4.8}$$

in which $f(\mathbf{x})$ is the pairwise force between the curve and one of its neighbor curves, which is the gradient of the pairwise potential as

$$f(\mathbf{x}) = \frac{\partial \psi}{\partial r} \nabla r(\mathbf{x}) \tag{4.9}$$

in which r is the distance between two curves.

Based on the definition of the distance r, there are two general methods to define the pairwise force $f(\mathbf{x})$ at the k^{th} control point for the curve φ mentioned in Section 4.1. One is based on the Euclidean distance between the average position of two curves, and the other is based on the minimum Euclidean distance among the control points of two curves. In the average position based Euclidean distance method, for each curve or each group of curves, the average position of all the control points of the curve or the group of curves is first calculated. Then the distance r between two curves is defined as the Euclidean distance between their corresponding average positions. Based on the way to define the average position of one curve, the method includes three scenarios which are pairwise force between one group and other groups, pairwise force between one curve and other groups, and pairwise force between one curve and other curves. In the average position based Euclidean distance method, each intermediate control point within one curve has the same collective force. The details of the three scenarios are presented in Section 4.3.2.1.1, 4.3.2.1.2, and 4.3.2.1.3 respectively.

In the minimum Euclidean distance method, the distance r is defined as the minimum Euclidean distance between two intermediate control points on two different curves. Based on the way to define the minimum Euclidean distance, the method includes two scenarios which are 'global' minimum Euclidean distance and 'local' minimum Euclidean distance. In the 'global' minimum Euclidean distance method, the distance r is defined as the minimum distance between two curves. Each intermediate control point within one curve has the same collective force which are calculated using the global minimum distance r for the curve. Different from the 'global' minimum Euclidean distance method, the 'local' minimum Euclidean distance method defines the minimum distance for each intermediate control point within one curve separately. For each intermediate control point within one curve, the method finds the corresponding minimum distance from the control point to its neighbor curves. In other words, the collective force on each intermediate control point within one curve are the different. The details of these two methods are presented in Section 4.3.2.1 and 4.3.2.2, respectively.

The collective force calculated from Eqn. (4.8) will be infinite when $r(\mathbf{x})$ approaches zero. As a result, the collective force will dominant the search process and push two curves too far away from each other, which violates the cohesion criterion. On the other hand, when two curves are far away from each other, they should not interact with each other during search process. However, the force calculated from the pairwise force model is not zero, which violates the requirement. To solve the problem, we introduce two cut-off distances. One is introduced to keep the collective force at the same magnitude with the true potential force, which is defined as

$$r_L = c_{cut\_off}\sigma \tag{4.10}$$

in which c_{cut_off} is a constant. If $r(\mathbf{x}) < r_L$, the pairwise force is defined as the force calculated at $r(\mathbf{x}) = r_L$. The other is introduced to deactivate the interaction among curves which are far away from each other, which is defined as

$$r_U = 2.5\sigma \tag{4.11}$$

If $r(\mathbf{x}) > r_U$, the pairwise force is set to be zero. Hence, the pairwise force is defined as

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if } r(\mathbf{x}) > r_U \\ \dfrac{\partial \psi}{\partial r} \nabla r(\mathbf{x}) & \text{if } r_L < r(\mathbf{x}) < r_U \\ \dfrac{\partial \psi}{\partial r}(c_{cut\_off}\sigma)\dfrac{\mathbf{x} - \mathbf{x}_{t\_curve}}{c_{cut\_off}\sigma} & \text{if } r(\mathbf{x}) < r_L \end{cases} \tag{4.12}$$

4.3.2.1 Average Position based Euclidean Distance

In this method, the distance r is defined as the Euclidean distance between average positions of two curves. Specifically, for the curve φ that mentioned in Section 4.1, the $r_\varphi(\mathbf{x})$ is defined as

$$r_\varphi(\mathbf{x}) = \| \mathbf{x} - \mathbf{x}_\varphi^{ave} \| \tag{4.13}$$

in which $\mathbf{x}_\varphi^{ave}$ is the average position for the curve φ, which are defined as

$$\mathbf{x}_\varphi^{ave} = \frac{1}{M}\sum_{k=1}^{M} \mathbf{x}_{k,\varphi} \tag{4.14}$$

in which is M the total number of control points for curve φ; $\mathbf{x}$ in the Eqn (4.13) is the average position of all the control points for the curve in its neighbor groups. For example, the $r_\varphi(\mathbf{x})$ at the l^{th} curve in the curve φ's neighbor groups is defined as $r_\varphi(\mathbf{x}_l^{ave})$. The $\mathbf{x}_l^{ave}$ is the average position for the l^{th} curve, which is defined in the same way as $\mathbf{x}_\varphi^{ave}$. By plugging $\mathbf{x}_l^{ave}$ into Eqns (4.13), (4.9) and (4.8), the collective force for the curve φ is

$$F(\mathbf{x}_\varphi^{ave}) = \sum_{l=1}^{L} w_l \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}_l^{ave}) \tag{4.15}$$

As mentioned at the beginning of the Section 4.3.2, the method includes three scenarios which are pairwise force between one group and other groups, pairwise force between one curve and other groups, and pairwise force between one curve and other curves based on the definition of the average positions of one curve and its neighbor curves. To understand the definition of the average position of one curve, it is necessary to clearly define the scope of the curve. The average position of one curve is defined as the arithmetic mean of the positions of all the control points within the scope of the curve. Figure 33 and Table 18 illustrate the definition of the scope for the curve φ and the l^{th} curve mentioned in Section 4.1 within the three scenarios for the calculation of the collective forces for the curve section φ_1 and φ_2. In the first scenario in which the pairwise forces are calculated between one group and the other groups, the curve φ is defined as all the curves in the group that the curve section φ_1 and φ_2 belongs to. In the simple example in Figure 33, the φ is defined as the curve that connects the states A, B, C, and D. So the average position of the curve φ (i.e. $\mathbf{x}_{\varphi}^{ave}$) is the arithmetic mean of the positions for the control points on the sub-curve sections AB, BC, and CD. Each's neighbor group is treated as one single curve for the pairwise force calculation. In other words, the l^{th} curve in φ's neighbor groups is defined as all the curves in the l^{th} group. The neighbor curves for curve φ are defined as the curve that connects the states A, F, E, and D, and the curve that connects the states G, H, I, and J. Thus, the average position for the l^{th} curve (i.e. $\mathbf{x}_{l}^{ave}$) is the arithmetic mean of the positions for the control points on the sub-curve sections AF, FE, ED when $l=1$ (i.e. the first neighbor curve) and GH, HI, IJ when $l=2$ (i.e. the second neighbor curve).

In the second scenario in which the pairwise forces are calculated between one curve and the other groups, the curve φ is defined as either curve section φ_1 or φ_2 depending on the curve section that the calculated collective forces will be applied. If the calculated collective force will be applied to section φ_1, then the curve φ is defined as curve section φ_1; otherwise is defined as curve section φ_2. The l^{th} curve is defined as the same in the first scenario. In the third scenario in which the pairwise forces are calculated between one curve and the other curves, the curve φ is defined in the same way as in the second scenario. The algorithm calculates the pairwise forces between curve φ and each curve in φ's neighbor groups one by one. In other words, each curve in φ's neighbor groups is one of the l^{th} curve. In the example in Figure 33, the neighbor curves for curve φ are defined as the curves that connects the states A and F, the states F and E, the states E and D, the states G and H, the states H and I, and the states I and J. The detailed description on how to calculate the collective force for the three scenarios and their corresponding test results are presented in Section 4.3.2.1.1, 4.3.2.1.2, and 4.3.2.1.3 respectively.

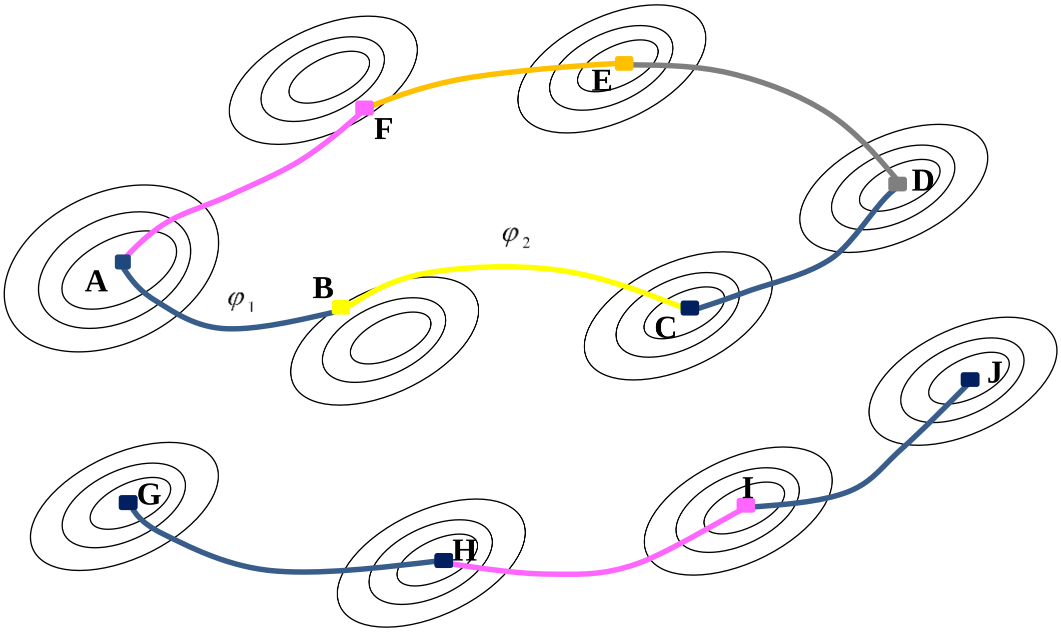


Figure 33: An illustration of the definition for the scope of curves in different scenarios

Table 18: Summary for the definition of curve scopes corresponding to the example in Figure 33

	curve φ	The l^{th} curve ($l = 1 \cdots L$)
Scenario_1 (group to group)	ABCD	AFED;GHIJ
Scenario_2 (curve to group)	AB or BC	AFED;GHIJ
Scenario_3 (curve to curve)	AB or BC	AF;FE;ED;GH;HI;IJ

4.3.2.1.1 Pairwise Force between One Group and Other Groups

In this scenario, the pairwise forces are calculated between two groups and each curve within one group has the same collective force. In other words, the algorithm calculates the pairwise forces by treating each group as one curve. The position of such curve is defined as the average position of all the control points in the group as in Eqn. (4.14). As in the simple example in Figure 33, to calculate the collective force for curve section φ_1, the $\mathbf{x}_{\varphi}^{ave}$ is defined as the average position for all the control points of the curve that connects the states A, B, C, and D. The number of curves in φ's neighbor groups is two, which means $L = 2$ in Eqn. (4.15) The $\mathbf{x}_{l}^{ave}$ is defined as the average position for all the control points of the curve that connects the states A, F, E, and D, and of the curve that connects the states G, H, I and J. The collective force for φ_1 in the example is

$$F(\mathbf{x}_{ABCD}^{ave}) = w_1 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{AFED}^{ave} \\ \varphi=ABCD}} + w_2 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{GHIJ}^{ave} \\ \varphi=ABCD}} \tag{4.16}$$

The collective force for the curve section φ_2 is the same as the one for the curve φ_1.

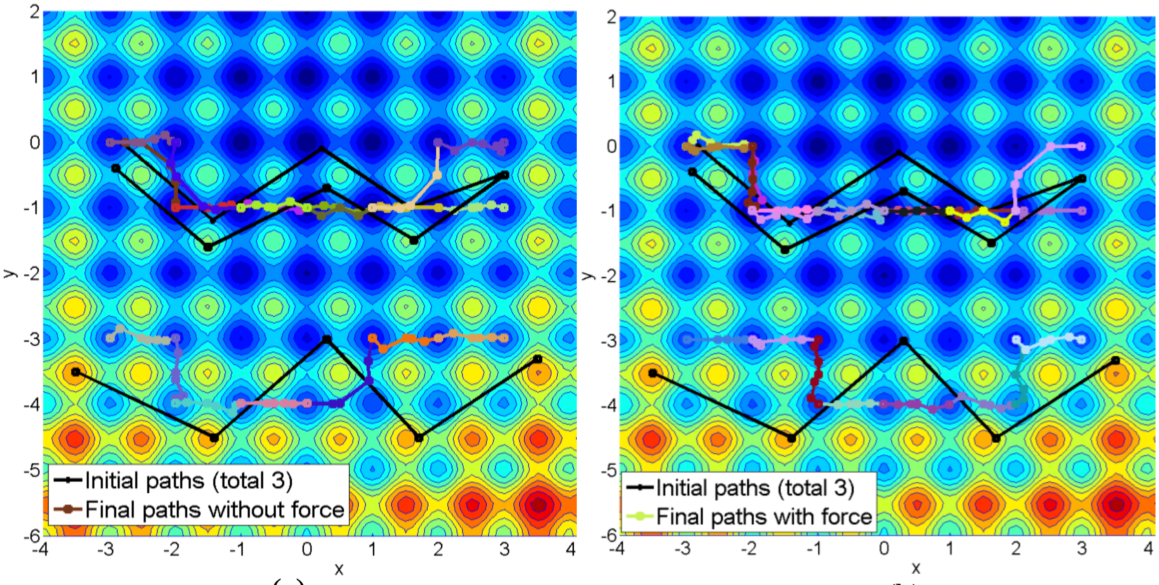
The algorithm is tested by the example of Rastrigin function which is a non-convex function frequently used to test global optimization algorithms. The function has a global minimum at $\mathbf{x} = (0, \cdots, 0)$ and uniformly distributed local minima as shown by a contour plot in Figure 34. As mentioned in Section 4.1, the algorithm uses groups of curves to search local minima and saddle points concurrently on a PES. Here three groups of curve are used. The initial curve for each group is a forth order Bézier curve $\varphi_j (j = 1, 2, 3)$. The initial positions for the three curves are listed in Table 19 and indicated in Figure 34 as 'initial path of group 1', 'initial path of group 2', and 'initial path of group 3'. The collective force for each curve is defined as the average force of the corresponding pairwise forces, i.e., the weights for the pairwise forces in Eqn. (4.15) are defined as 1. The cutoff coefficient c_{cut_off} for the Lennard-Jones potential in Eqn. (4.10) is defined as 1.0621 in the test.

As stated in Section 4.1, the central issue of the curve swarm algorithm is to maintain cohesion and avoid collision which is achieved by introducing a collective potential model to describe the collective forces among groups. To demonstrate the effectiveness of the collective potential model, we first run the experiment by setting the collective potential as zero, i.e., without applying the collective force to the intermediate control points. The selection of the initial curves in the test is based on the criterion that the first two groups of curves will collide with each other after several iterations in order to test the capability of avoiding collision of the algorithm and the third one will keep far away from the first two in order to test the capability of maintaining cohesion for the algorithm. Then with the same sets of initial curves, we run the experiment in which collective forces are applied to all the intermediate control points. The results for the experiments without and with collective force being applied to intermediate control points are showed in Figure 34 (a) and (b), separately.

The results show clearly that the collective forces which are calculated based on group to group pairwise forces are not effective in both collision avoidance and cohesion maintenance. In other words, the collective forces do not push two curves away when they are getting too close to each other and pull close when they are far away from each other. The reason is that using the arithmetic mean of positions at all the control points in one group to approximate the position of one curve is too rough. In other words, the distance between two groups of curves calculated based on average positions does not correctly capture the distribution of the curves on the PES. Thus, in the next section, we provide a different way to calculate the average position of the curves.

Table 19: Position for initial curves in each group (Rastrigin function)

Initial curve	Point 1	Point 2	Point 3	Point 4	Point 5
φ_1	(−2.81, 0)	(−1.43, −1.19)	(0.23, −0.1)	(1.57, −1)	(2.91, −0.51)
φ_2	(−2.9, −0.4)	(−1.5, −1.6)	(0.3, −0.7)	(1.63, −1.49)	(3, −0.5)
φ_3	(−3.5, −3.5)	(−1.4, −4.5)	(0.3, −3)	(1.7, −4.5)	(3.5, −3.3)



(a) (b)

Figure 34: Test results (a) without collective force (b) with group to group collective force applied to intermediate control points.

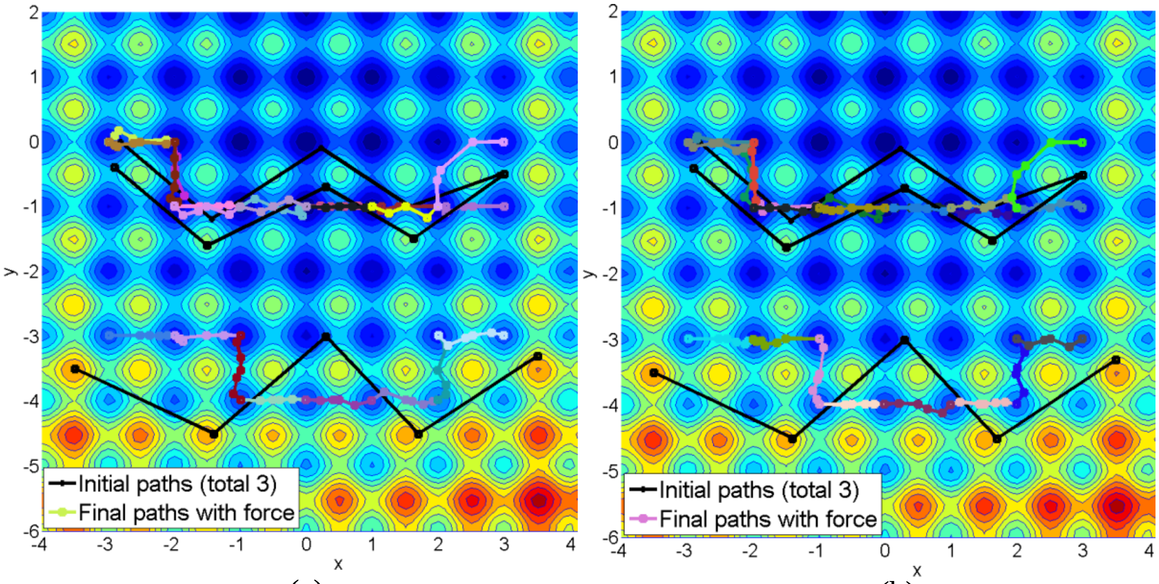
4.3.2.1.2 Pairwise Force between One Curve and the Other Groups

In this scenario, the pairwise forces are calculated between one curve and other groups. The difference between this scenario and the one in Section 4.3.2.1.1 lies in the definition of the curve φ that needs to be optimized. In this scenario, the curve φ is defined as the curve section that needs to be optimized instead of as the whole group of curves to which the curve section belongs to. As in the simple example in Figure 33, to calculate the collective force for curve section φ_1, the curve φ is defined as the curve section φ_1. In other words, the $\mathbf{x}_{\varphi}^{ave}$ in Eqn. (4.15) is defined as the average positions for all the control points of the curve that connects the states A and B. The number of curves in φ's neighbor group is two, which means $L = 2$ in Eqn. (4.15) The $\mathbf{x}_{l}^{ave}$ is defined as the average position for all the control points of the curve that connects the states A, F, E, and D, and of the curve that connects the states G, H, I, and J. The collective force for φ_1 in the example is

$$F(\mathbf{x}_{AB}^{ave}) = w_1 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Bigg|_{\substack{\mathbf{x}=\mathbf{x}_{AFED}^{ave} \\ \varphi=AB}} + w_2 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Bigg|_{\substack{\mathbf{x}=\mathbf{x}_{GHIJ}^{ave} \\ \varphi=AB}} \qquad (4.17)$$

The collective force for the curve section φ_2 is different with the one for the curve φ_1. To calculate the collective force for the curve section φ_2, the curve φ is defined as the curve section φ_2 i.e., the $\mathbf{x}_{\varphi}^{ave}$ in Eqn. (4.15) is defined as the average positions for all the control points of the curve that connects the states B and C. The definition for its neighbor curves are the same as the ones for curve φ_1. Hence, the collective force for φ_2 in the example is

$$F(\mathbf{x}_{BC}^{ave}) = w_1 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{AFED}^{ave} \\ \varphi=BC}} + w_2 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{GHIJ}^{ave} \\ \varphi=BC}} \qquad (4.18)$$



(a)

(b)

Figure 35: Test results (a) without group to group collective force (b) with curve to group collective force applied to intermediate control points.

To compare with the result of the first scenario in which the pairwise force is calculated between two groups, the algorithm is tested in the same way as did in Section 4.3.2.1.1. We choose the same three initial curves whose positions are listed in Table 19 as in the first scenario. The cutoff coefficient c_{cut_off} for the Lennard-Jones potential in Eqn. (4.10) is defined as 1.0621 in the test. The test result for the scenario with curve to group pairwise force is shown in Figure 35 (b). As a comparison, the result for the scenario with group to group pairwise force is listed in Figure 35 (a). The result in Figure 35 (b) is does not show any improvement compared with the one in Figure 35 (a). The reason is that the error involved in approximating the position of the neighbor curves using the average position at all the control points that group is too big. The distribution of the neighbor curves is not correctly captured. Thus, in the next section, we provide a more accurate way to approximate the positions of the neighbor curves

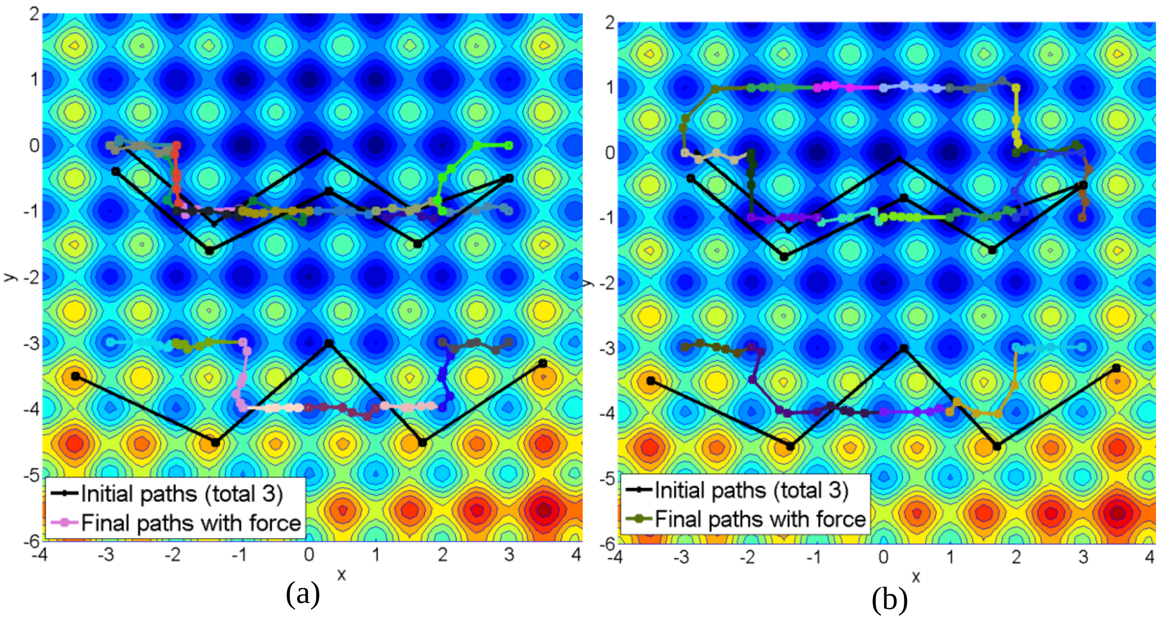
4.3.2.1.3 Pairwise Force between One Curve and the Other Curves

In this scenario, the pairwise forces are calculated between two curves. The difference between this scenario and the one in Section 4.3.2.1.2 lies in the definition of the l^{th} curve in the curve φ's neighborhood. In the scenario, each curve in φ's neighbor groups represents an independent curve i.e., the l^{th} curve. As in the simple example in Figure 33, to calculate the collective force for curve section φ_1, the curve φ is defined as the curve section φ_1. In other words, the $\mathbf{x}_{\varphi}^{ave}$ in Eqn. (4.15) is defined as the average positions for all the control points of the curve that connects the states A and B. The number of curves in φ's neighbor group is six, which means $L = 6$ in Eqn. (4.15) Each $\mathbf{x}_{l}^{ave}$ in Eqn. (4.15) is defined as the average position for all the control points of the curve that connects the states A and F, F and E, E and D, G and H, H and I, I and J. The collective force for φ_1 in the example is

$$\begin{aligned} F(\mathbf{x}_{AB}^{ave}) = w_1 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{AF}^{ave} \\ \varphi=AB}} + w_2 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{FE}^{ave} \\ \varphi=AB}} + w_3 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{ED}^{ave} \\ \varphi=AB}} \\ + w_4 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{GH}^{ave} \\ \varphi=AB}} + w_5 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{HI}^{ave} \\ \varphi=AB}} + w_6 \frac{\partial \psi}{\partial r} \nabla r_{\varphi}(\mathbf{x}) \Big|_{\substack{\mathbf{x}=\mathbf{x}_{IJ}^{ave} \\ \varphi=AB}} \end{aligned} \tag{4.19}$$

The collective force for the curve section φ_2 is different with the one for the curve φ_1. To calculate the collective force for the curve φ_2, the curve φ is defined as the curve section φ_2 i.e., the $\mathbf{x}_{\varphi}^{ave}$ in Eqn. (4.15) is defined as the average position for all the control points of the curve that connects the states B and C. The definition for its neighbor curves are the same as the ones for curve φ_1. Hence, the collective force for φ_2 in the example is

$$F(\mathbf{x}_{BC}^{ave}) = w_1 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{AF}^{ave} \\ \varphi=BC}} + w_2 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{FE}^{ave} \\ \varphi=BC}} + w_3 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{ED}^{ave} \\ \varphi=BC}}$$
$$+ w_4 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{GH}^{ave} \\ \varphi=BC}} + w_5 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{HI}^{ave} \\ \varphi=BC}} + w_6 \frac{\partial \psi}{\partial r} \nabla r_\varphi(\mathbf{x}) \bigg|_{\substack{\mathbf{x}=\mathbf{x}_{IJ}^{ave} \\ \varphi=BC}} \quad (4.20)$$



(a)

(b)

Figure 36: Test results (a) with curve to group collective force (b) with curve to curve collective force applied to intermediate control point.

To compare with the result of the second scenario in which the pairwise force is calculated between one curve and the other groups, the algorithm is tested in the same way as did in Section 4.3.2.1.1. We choose the same three initial curves whose positions are listed in Table 19 as in the first scenario. The cutoff coefficient c_{cut_off} for the Lennard-Jones potential in Eqn. (4.10) is defined as 1.0621 in the test. The test result for the scenario with curve-to-curve pairwise force is shown in Figure 36(b). As a comparison, the result for the scenario with curve-to-group pairwise force is shown in Figure 36(a). The result in Figure 36(b) is better than the one in Figure 36(a) in terms of avoiding collision. It indicates that the curve-to-curve pairwise force is a better way to describe the collective behavior among curves than the group-to-group pairwise force. The reason for this is that the error in approximating the position of a curve using the average position of its control points is smaller than the one in approximating the position of a group of curves using their average positions of all the control points of the curves in the group. However, the algorithm does not work effectively in terms of maintain cohesion. The algorithm pushes the first two groups too far away. Also, it does not pull the third group closer. The curve to curve scheme is not able to correctly describe the distribution of curves. Hence, in the next section, a new scheme is presented to solve this issue.

4.3.2.2 Minimum Euclidean Distance among Control Points

In this scenario, the pairwise force between two curves is calculated between two control points with minimum Euclidean distance. Since the introduction of pairwise forces calculated based on the methods in sections from 4.3.2.1.1 to 4.3.2.1.3 does not solve the cohesion problem, the minimum Euclidean distance scheme is introduced here to solve such problem. The key idea behind the minimum distance scheme is to push two curves away at their closest position between each other. Since it is computational expensive to locate the exact closest position between two curves using traditional distance based optimization methods, we simplify the problem by locating a pair of control points with minimum Euclidean distance on these two curves.

To better illustrate how to locate the control points with minimum distance, we may use a matrix form for the locations of the intermediate control points of each curve, with rows from one to n containing the n-dimensional coordinates for each intermediate control points on the curve. For example, for one curve with four intermediate control points is represented by the matrix

$$\begin{bmatrix} x_{11} & x_{21} & x_{31} & x_{41} \\ x_{12} & x_{22} & x_{32} & x_{42} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1n} & x_{2n} & x_{3n} & x_{4n} \end{bmatrix}$$

The number of columns is the total number of the intermediate control points for the curve. To locate the pair of control points with minimum distance between the newly created curve and one of its neighbor curves, we first represent the locations of their intermediate control points in two separate matrices with identical row dimensions. Starting with the first column of the matrix for the newly created curve, we calculate the Euclidean distance between this location and each location in the matrix of its neighbor curve. The process is then repeated for all the columns in the matrix of the newly created curve. The locations of the pair of control points that will be used to calculate the pairwise force between these two curves are determined by minimum distance. That is, if $[\mathbf{p}_1,\ldots,\mathbf{p}_m]$, in which $\mathbf{p}_m = [x_{m1},\ldots,x_{mn}]^T$ is the matrix for the newly created curves with m intermediate control points, and the matrix for its neighbor curve is $[\mathbf{q}_1,\ldots,\mathbf{q}_{m'}]$, in which $\mathbf{q}_{m'} = [x_{m'1},\ldots,x_{m'n}]^T$, then the distance d_{ij} between the i^{th} location in the matrix of the newly created curve and the j^{th} location in the matrix of its neighbor curve is $d_{ij} = |\mathbf{p}_i - \mathbf{q}_j|$, in which $1 \le i \le m$ and $1 \le j \le m'$.

After calculating all the pairwise distances, the locations of the minimum distance is determined by two different ways. One is 'global' minimum distance between two curves, and the other is 'local' minimum distance between two curves. Here, 'global' means the minimum distance is the minimum value of all combinations of d_{ij}. We locate only one pair of control points with 'global' minimum distance between two curves. The pairwise force between these two curves is calculated using the locations of this pair of control points. The pairwise force is then applied to each intermediate control point. In other words, each of the intermediate control point has the same pairwise force and as result the same collective force. On the other hand, 'local' means that each intermediate control point has its own corresponding minimum distance. For the i^{th} intermediate control point in the newly created curve, the corresponding minimum distance is defined as $\min_{1 \le j \le m'} d_{ij}$. The pairwise force for each intermediate control point is calculated separately using its own corresponding minimum-distance control point. In other words, each of the intermediate control point has different pairwise force and as result the different collective force. The detailed description and test results of the two schemes are presented in Section 4.3.2.2.1 and 4.3.2.2.2.

4.3.2.2.1 Global minimum distance between two curves

The scheme includes three steps which are 1) determining each pair of control points with 'global' minimum distance between one curve and each of its neighbor curves; 2) calculating the pairwise forces from each of its neighbor curves using the identified control points; 3) summing up the pairwise forces to obtain the collective force which is applied to each intermediate control point of the curve as defined by Eqn.(4.1). As mentioned at the beginning of the Section 4.3.2.2, the ‘global’ minimum distance is the minimum value of all combinations of d_{ij}. The locations of the pair of control points with ‘global’ minimum distance is determined by

$$(i_{new}, j_{neighbor}) = \arg \min_{\substack{1 \le i \le m \\ 1 \le j \le m'}} d_{ij} \tag{4.21}$$

in which i_{new} and $j_{neighbor}$ is the location of the control point with ‘global’ minimum distance on the newly created curve and its neighbor curve correspondingly.

To better illustrate the scheme, we take the simple example in Figure 37. The collective force for curve section φ_1 is the sum of the pairwise forces between the curve section AB and the curve sections from its neighbor groups, i.e. AF, FE, ED, GH, HI, and IJ. Here, I will illustrate the minimum distance scheme by the example of calculating the pairwise forces between the curve sections AB and AF. The curve section AB has five control points with p_1, p_2, and p_3 as its intermediate control points. The curve section AF has six control points with q_1, q_2, q_3, and q_4 as its intermediate control points. To find the pair of control points with minimum distance, we first construct the location matrices for curve AB and AF, which are $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ and $[\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4]$ respectively. The $\mathbf{p}_*$ and $\mathbf{q}_*$ are location vectors for intermediate control points. Starting from the first column $\mathbf{p}_1$ of the matrix for curve AB, the Euclidean distances between the location $\mathbf{p}_1$ and each column of the matrix for curve AF are calculated, which are d_{11}, d_{12}, d_{13}, d_{14} and as shown in Figure 37. The process is repeated for the rest of columns in matrix for curve AB which produces eight more distances that are d_{21}, d_{22}, d_{23}, d_{24}, d_{31}, d_{32}, d_{33} , and d_{34}. The two control points that have the distance equals to the ‘global’ minimum distance $\min\limits_{1 \le i \le 3, 1 \le j \le 4} d_{ij}$ are used to calculate the pairwise force between the curve AB and AF. Similarly, the procedure is applied to the rest of the curves FE, ED, GH, HI, and IJ to calculate the pairwise forces between the curve AB and each of those curves. After obtaining all the pairwise forces, the collective force is obtained by summing up all the pairwise forces. The collective force is then applied to each intermediate control point during the search process as stated in Section 4.1.

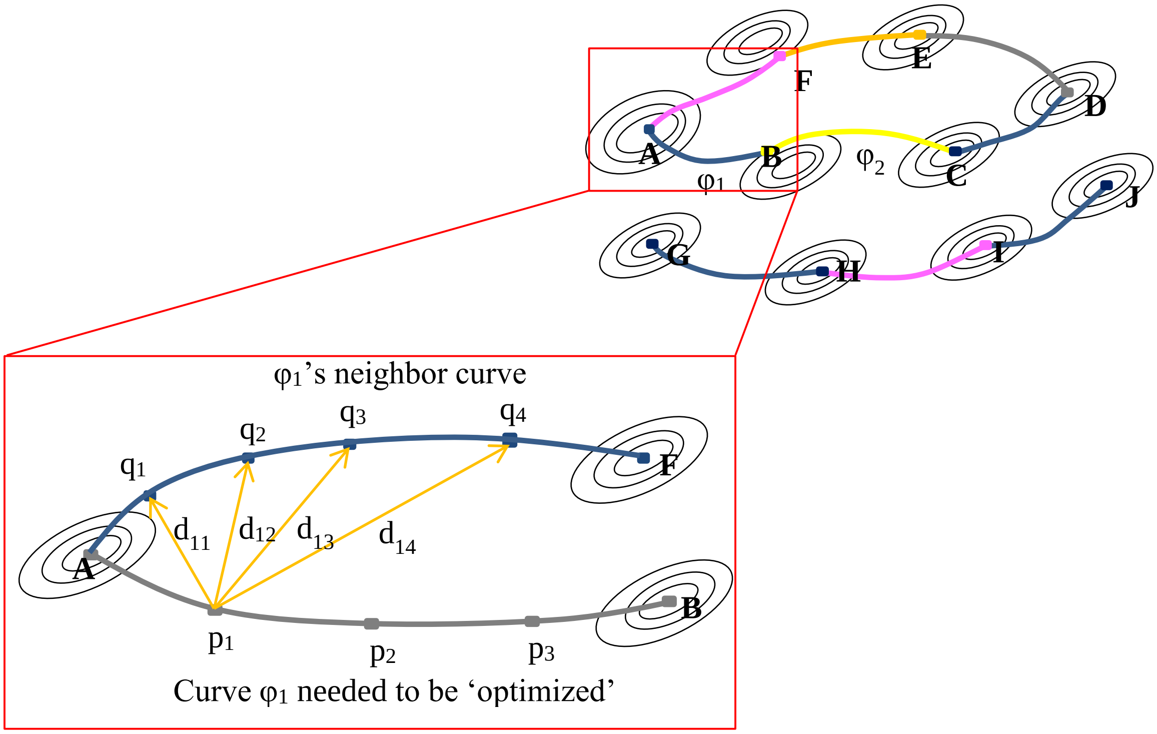
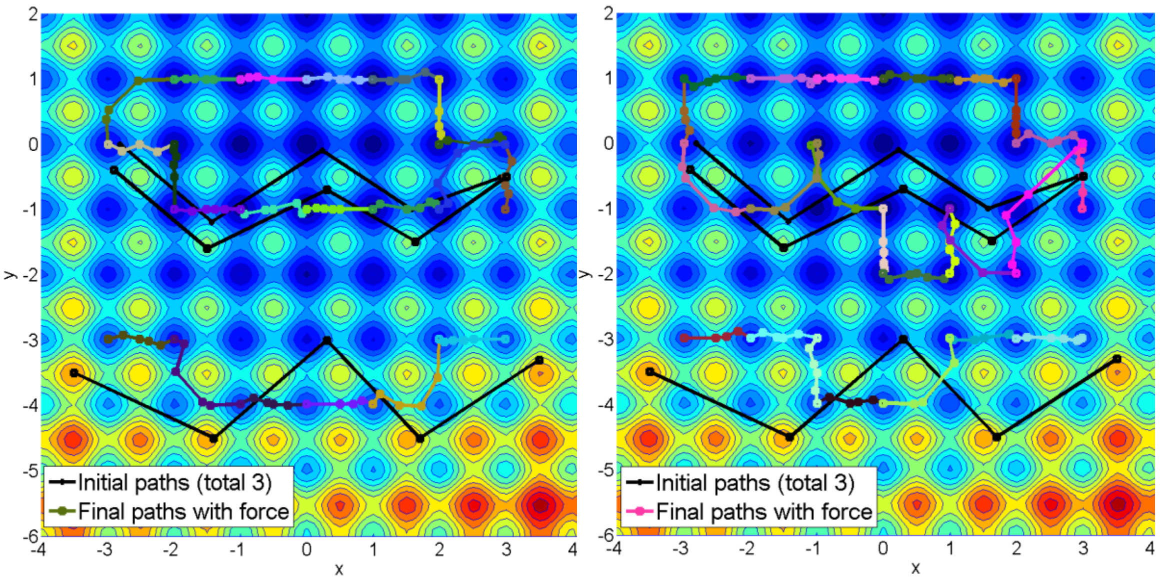


Figure 37: Illustration of the minimum distance scheme



(a) (b)

Figure 38: Test results (a) with curve to curve collective force (b) with global minimum distance based collective force.

The scheme is tested using the example of Rastrigin function. The parameter settings and the locations of initial curves are the same with the one in previous sections for the convenience of comparison. The result is compared with the one from the curve to curve scheme in Section 4.3.2.1.3. Figure 38(a) and (b) show the result from the curve to curve scheme and the one from the 'global' minimum distance scheme. Figure 38(b) does not show much improvement in terms of maintaining cohesion. The reason is that the 'global' minimum distance is not able to capture the correct distance between two curves at each intermediate control point. Hence, the 'local' minimum distance is proposed and presented in the next section.

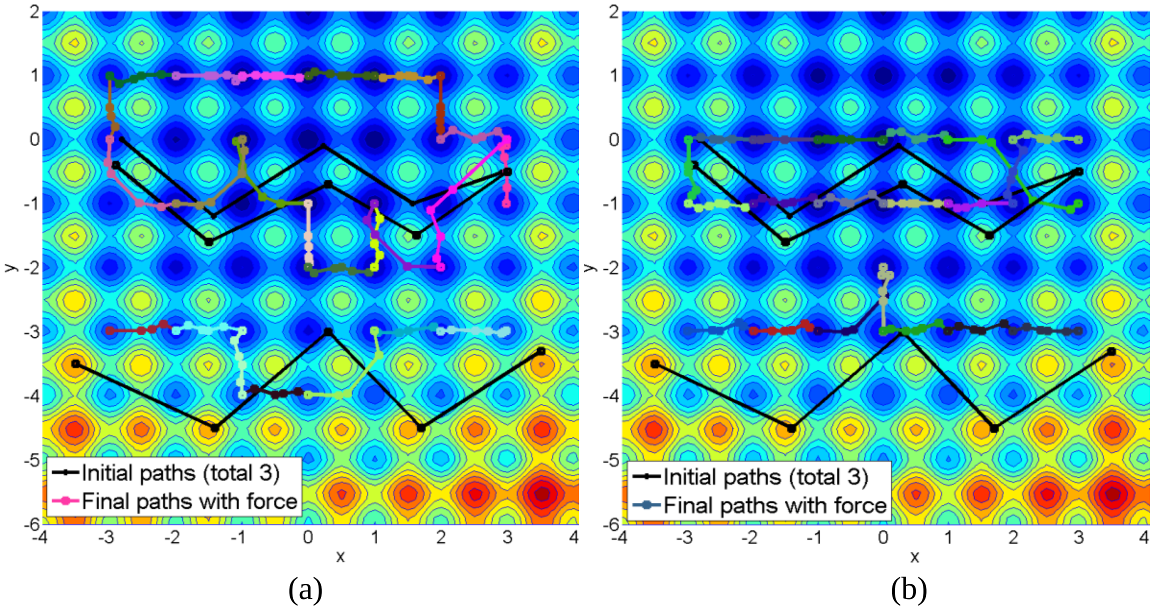
4.3.2.2.2 Local minimum distance for each control points

Similar to the 'global' minimum distance scheme in Section 4.3.2.2.1, the scheme includes three steps which are 1) determining the corresponding control points with 'local' minimum distance for each intermediate control point on the newly created curve; 2) calculating the pairwise forces from each of its neighbor curves using the identified control points; 3) summing up the corresponding pairwise forces to obtain the collective forces for each intermediate control point which is applied to the corresponding control points of the curve as defined by Eqn.(4.1). As mentioned at the beginning of the Section 4.3.2.2, the 'local' minimum distance indicates that each intermediate control point has its own corresponding minimum-distance control point on each of its neighbor curves. For the i^{th} intermediate control point with the index i_{new} in the newly created curve, the corresponding minimum-distance control point with the index $j_{neighbor}$ on one of its neighbor curves is defined as

$$(i_{new}, j_{neighbor}) = \min_{\substack{i=i_{new} \\ 1\le j\le m'}} d_{ij} \tag{4.22}$$

To better illustrate the scheme, we take the simple example in Figure 37 as did in Section 4.3.2.2.2. The collective force for curve section φ_1 is the sum of the pairwise forces between the curve section AB and the curve sections from its neighbor groups, i.e. AF, FE, ED, GH, HI, and IJ. Here, I will illustrate the 'local' minimum distance scheme by the example of calculating the pairwise forces between the curve section AB and AF. The curve section AB has five control points with p_1, p_2, and p_3 as its intermediate control points. The curve section AF has six control points with q_1, q_2, q_3, and q_4 as its intermediate control points. As stated at the beginning of this section, each intermediate control point has its own collective force. In this example, we need to calculate the collective force for control points p$_1$, p$_2$, and p$_3$ respectively. To find the control points with local minimum distance for control points p$_1$, p$_2$, and p$_3$ respectively, we first construct the location matrices for curve AB and AF, which are $[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]$ and $[\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4]$ respectively. The $\mathbf{p}_*$ and $\mathbf{q}_*$ are location vectors for intermediate control points. Starting from the first column $\mathbf{p}_1$ of the matrix for curve AB, the Euclidean distances between the location $\mathbf{p}_1$ and each column of the matrix for curve AF are calculated, which are d_{11}, d_{12}, d_{13}, and d_{14} as shown in Figure 37. The process is repeated for the rest of columns in matrix for curve AB which produces eight more distances that are d_{21}, d_{22}, d_{23}, d_{24}, d_{31}, d_{32}, d_{33}, and d_{34}. The corresponding local minimum distance control point for p$_1$ is the one with distance equals to $\min_{i=1, 1 \le j \le 4} d_{ij}$. Such control point is used to calculate the pairwise force between the curve AB and AF at control point p$_1$. Same for the control point p$_2$, and p$_{3,}$ the corresponding local minimum distance control points are the ones with distance equals to $\min_{i=2, 1 \le j \le 4} d_{ij}$ and $\min_{i=3, 1 \le j \le 4} d_{ij}$, respectively. Those two control points are used to calculate the pairwise force between the curve AB and AF at control points p$_2$, and p$_3$ respectively. Similarly, the procedure is applied for the rest of the curves FE, ED, GH, HI, and IJ to calculate the pairwise forces between the curve AB and each of those curves at the control points p$_1$, p$_2$, and p$_3$ on curve AB respectively. At the end, the corresponding pairwise forces are summed up to obtain the collective forces at the control points p$_1$, p$_2$, and p$_3$ respectively.

The scheme is tested using the example of Rastrigin function. The parameter settings and the locations of initial curves are the same with the one in previous sections for the purpose of result comparison. Figure 39(a) and (b) shows the results from the global minimum distance scheme and local minimum distance scheme, respectively. Both of the global and local minimum distance scheme effectively avoid collision. However, only the local minimum distance scheme maintains cohesion. The minimum distance scheme is able to correctly capture the distribution of curves. Thus, the minimum distance scheme is adopted in the algorithm to calculate the compare wise force among curves using Lennard-Jones potential. To test the robustness, convergence, scalability of the algorithm, more tests are conducted which are described in Section 4.4 and 4.5.



(a) (b)

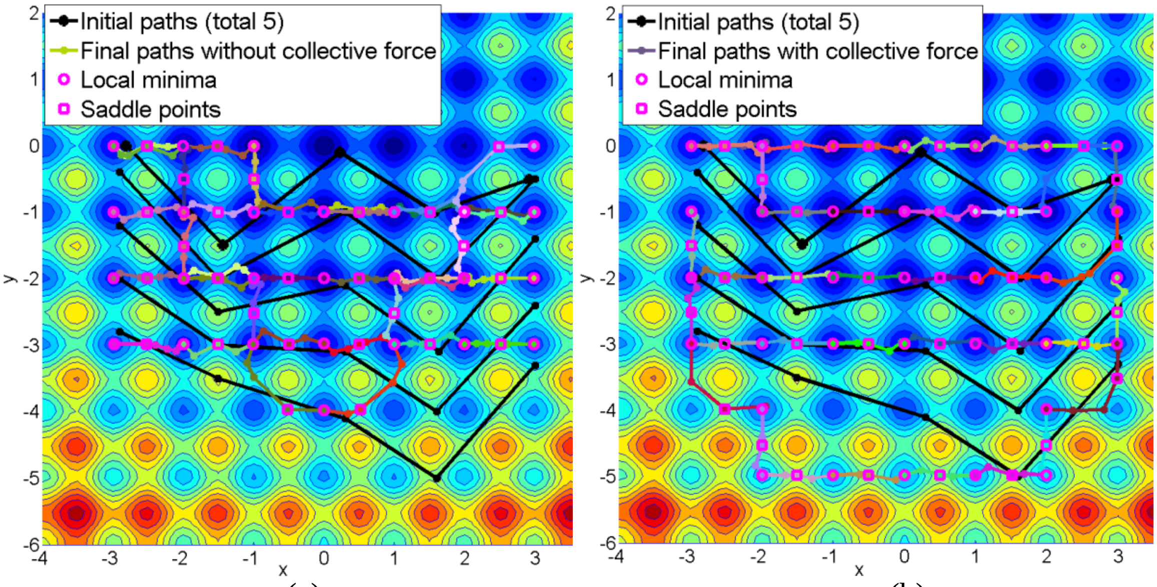
Figure 39: Test results with (a) with global minimum distance (b) with local minimum distance based collective force.

4.4 Implementation and Demonstration

The algorithm is demonstrated by examples of Rastrigin function, Schwefel function, and the diffusion process of the hydrogen atoms in the FeTi system and the pure iron system. The detailed description of those functions and processes are presented in Section 3.4. The algorithm is implemented in MATLAB. Based on the test results in Section 4.3.2, the collective force in the following examples are calculated using the scheme of local minimum distance for each control point which is presented in Section 4.3.2.2.2.

4.4.1 Test Results for Rastrigin Function

In Section 4.3.2.2.2, we test the local minimum distance scheme using Rastrigin function. Here, the algorithm is tested with more initial curves. All the computational settings are the same with the ones in Section 4.3.2.2.2. The initial positions for the initial curves are listed in Table 19. Similarly, the algorithm's capability to exhaustively search local minima and saddle points within an area is demonstrated by comparing the results from the curve swarm algorithm and the ones from the concurrent search algorithm without applying collective force. Figure 40(a) shows that two curves (e.g. the first and second curve, and the third and fourth curve) may duplicate the search efforts and find the same result when there is no communication among curves. Figure 41(b) shows that the collective force introduced in the curve swarm algorithm pushed those two curves away. As a result, the curve swarm search algorithm locates 35 local minima and 35 saddle points which is more than the ones (26 local minima and 29 saddle points) located by concurrent search algorithm without applying collective forces. The number of local minima and saddle points which are identified by the concurrent search algorithm without applying collective force and curve swarm search algorithm are listed in Table 20.



(a) (b)

Figure 40: Test results (a) without collective force (b) with global minimum distance based collective force applied to the intermediate control points

Table 20: Coordinates of the control points for the five initial curves (Rastrigin function)

Initial curve	Point 1	Point 2	Point 3	Point 4	Point 5
1	(−2.81, 0)	(−1.43, −1.49)	(0.23, −0.1)	(1.57, −1)	(2.91, −0.51)
2	(−2.9, −0.4)	(−1.5, −2.)	(0.3, −1)	(1.63, −2)	(3, −0.5)
3	(−2.9, −1.2)	(−1.5, −2.5)	(0.3, −2.1)	(1.63, −3.1)	(3, −1.4)
4	(−2.9, −2)	(−1.5, −3)	(0.3, −3.1)	(1.6, −4)	(3, −2.4)
5	(−2.9, −2.8)	(−1.5, −3.5)	(0.3, −4.1)	(1.6, −5)	(3, −3.3)

Table 21: The number of identified local minima and saddle points (Rastrigin function)

	Number of local minima	Number of saddle points
Concurrent search algorithm without applying collective force	26	29
Curve swarm search algorithm	35	35

4.4.2 Test Results for Schwefel Function

Here, we test the algorithm with Schwefel function which has a more complicate energy landscape than Rastrigin function. There are a total of five initial curves which are the curves in black with asterisk marker in Figure 41. The coordinates of the control points on the initial curves are included in Table 22.

Similarly, the algorithm’s capability to exhaustively search local minima and saddle points within an area is demonstrated by comparing the results from the curve swarm algorithm and the ones from the concurrent search algorithm without applying collective force. Figure 41 (a) shows that two curves (e.g. the first and second curve, and the fourth and fifth curve) may duplicate the search efforts and find the same result when there is no communication among curves. Figure 41(b) shows that the collective force introduced in the curve swarm algorithm pushed those two curves away. As a result, the curve swarm search algorithm locates more local minima and saddle points than the concurrent search algorithm without applying collective forces. The number of local minima and saddle points which are identified by the concurrent search algorithm without applying collective force and curve swarm search algorithm are listed in Table 23.

Table 22: Coordinates of the control points for the initial curves (Schwefel function)

Initial curve	Point 1	Point 2	Point 3	Point 4	Point 5
1	(−225.3, 256)	(−89, 100)	(49, 360)	(185.6, 115)	(320, 375)
2	(−225.3, 158)	(−89, 2)	(49, 262)	(185.6, 17)	(320, 277)
3	(−225.3, 60)	(−89, −96)	(49, 164)	(185.6, −81)	(320, 179)
4	(−225.3, −38)	(−89, −194)	(49, 66)	(185.6, −179)	(320, 81)
5	(−225.3, −136)	(−89, −292)	(49, −32)	(185.6, −277)	(320, −17)

Table 23: The number of identified local minima and saddle points (Schwefel function)

	Number of local minima	Number of saddle points
Concurrent search algorithm without applying collective force	27	27
Curve swarm search algorithm	34	35

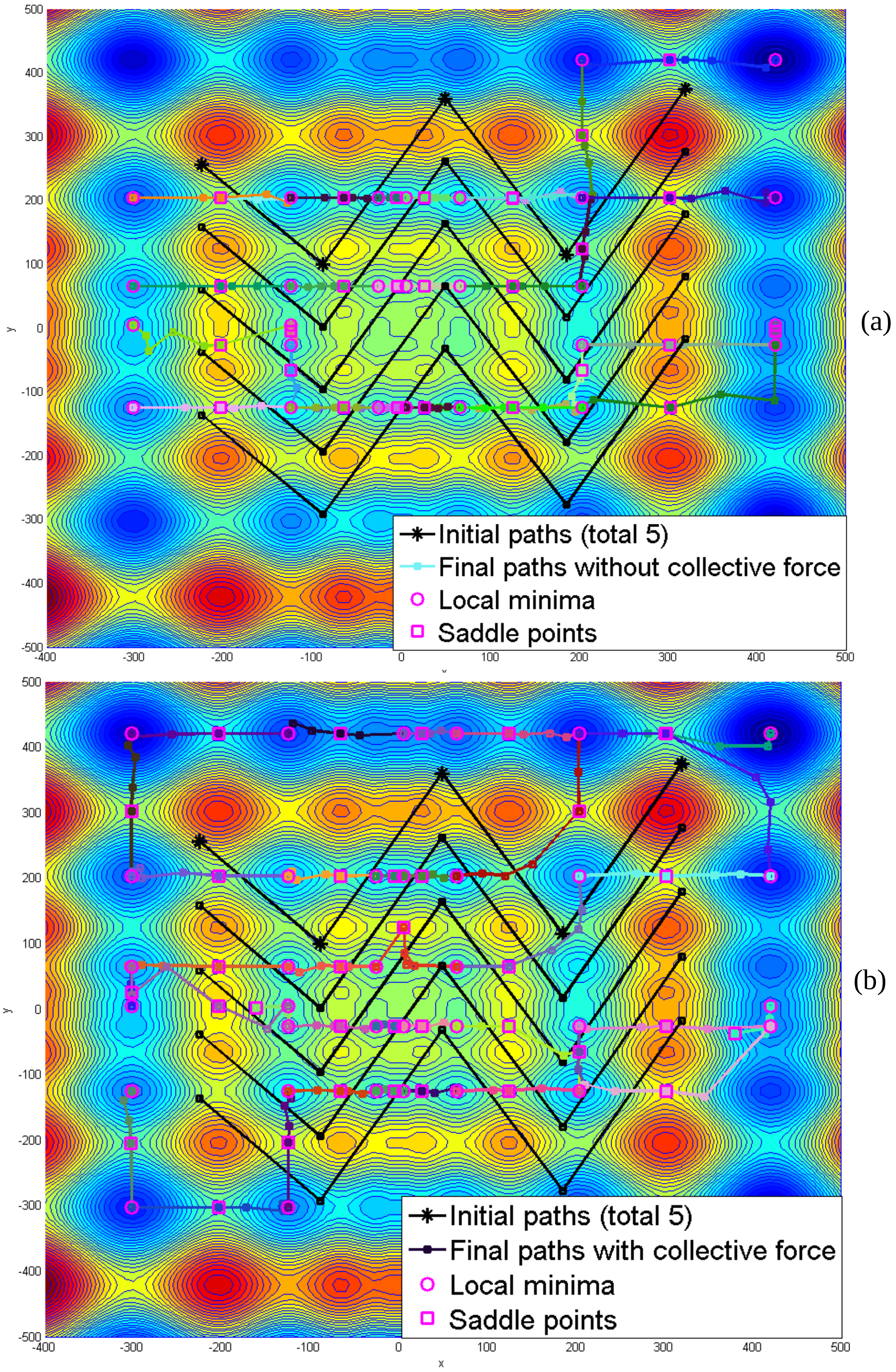


Figure 41: Test result Schwefel function (a) concurrent search algorithm without applying collective force (b) curve swarm search algorithm

4.4.3 Test Results for FeTi-H System

In this section, the curve swarm search algorithm is used to study the hydrogen diffusion process in the Iron-Titanium system. The detailed description about the FeTi-H can be found in Section 3.4.5.

4.4.3.1 Computational Details

To save the computational time, here three initial curves are used in the test. The initial structures for the three initial curves are the ones in Figure 15(a-c) with the fractional coordinates being listed in Table 11(a-c). The final structures for all the initial curves are set to be β phase FeTiH which shown in Figure 14(b) with its fractional coordinates being listed in Table 12. To maintain consistence with the previous studies, here we refer the three transitions as FeTiH_tran_a, FeTiH_tran_b, and FeTiH_tran_c.

Similarly, the total energy of the system and the forces on each atom are calculated based on DFT calculation using VASP. The projector augmented wave potentials, specifically LDA potentials, and a $4\times2\times2$ k-point sampling scheme are used here. The unit cells for both initial and final structures are defined as $a = 2.956$ Å, $b = 4.543$ Å, and $c = 4.388$ Å. The procedure described in Figure 29 and Table 17 is applied to the three initial curves to locate multiple local minima and saddle points. The collective force among curves are calculated using 'local' minimum distance scheme. The cutoff coefficient c_{cut_off} for the Lennard-Jones potential in Eqn. (4.10) is defined as 1.085.

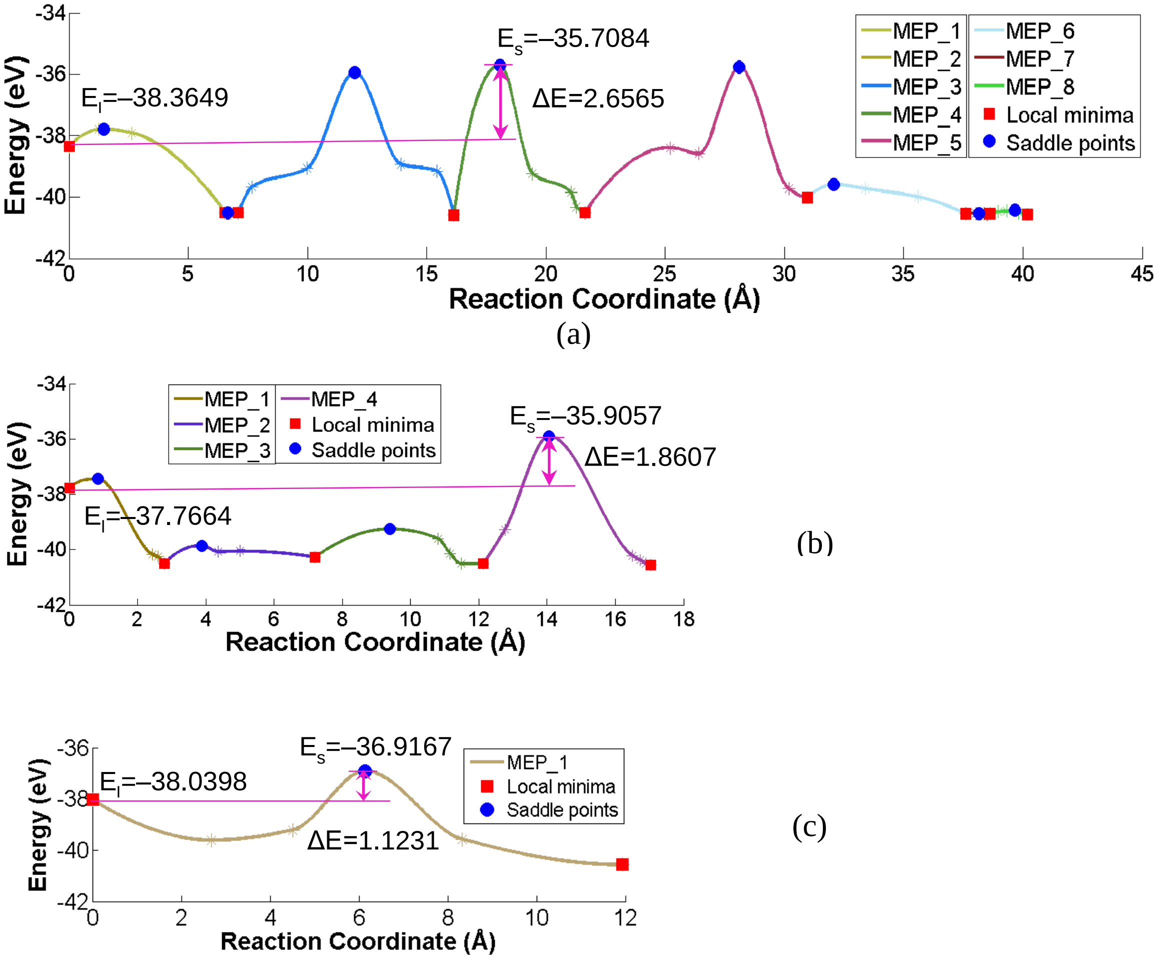
4.4.3.2 Computational Results and Discussion

The located minimum energy paths (MEPs) for the three groups of curves are plotted in terms of total energy (eV) with respect to the reaction coordinates which are shown in Figure 42. Similarly, the E_I and E_S are the total energy at the initial states and transition states (i.e. saddle points with highest energy or global saddle points along the transition path). The ΔE represents the activation energy in eV. The square marker in red are the local minima and the round marker in blue are the saddle points. The asterisk markers are intermediate images along the MEPs. Each curve section separated by the square marker in different colors represents a sub-stage transition path. For the first group, the algorithm locates eight MEPs with nine local minima and eight saddle points which are shown in Figure 42(a). The global saddle point along the transition FeTiH_tran_a is the one on the fourth curve with a total energy of –35.7084 eV. The total energy for the initial state is –38.3649 eV. The activation energy for the transition is the energy difference between the global saddle point and the initial state which is 2.6565 eV for FeTiH_tran_a.

For the second group, the algorithm locates four MEPs with five local minima and four saddle points which are shown in Figure 42(b). The total energy of the initial state is –37.7664 eV and the global saddle point is –35.9057 eV which is the one on the fourth curve. The activation energy for the FeTiH_tran_b is 1.8607 eV. For the third group, the algorithm locates only one MEPs with two local minima and one saddle points with a total energy of –36.9167 eV which is shown in Figure 42(c). The total energy for the initial state is –38.0398 eV and the activation energy for FeTiH_tran_c is 1.1231 eV. The test results are summarized in Table 24.

It is worth to notify that even with the same initial curves, the results obtained by the curve swarm search algorithm are different with the ones by the concurrent search algorithm. The reason is based on the fact that there are multiple transition paths for the transition from one initial state to the final state as illustrated in Figure 28. Due to the introduction of the collective forces in the curve swarm search algorithm, the same initial curve may end up with different final MEPs in curve swarm search algorithm compared with the one in the concurrent search algorithm. With more groups of initial curves, the curve swarm search algorithm will provide more information about the landscape of the transition in a systematic way. Even though the concurrent search algorithm can provide more information as well given many trials are done with different initial curves, it is not systematic and computational expensive since it may end up duplicate exploration in the same paths. In addition, there is high chance they may miss some search area since the search on the surface is pure random. The curve swarm search algorithm is designed to solve these two problems by using groups of curves which are manipulated by the collective forces.

Figure 43-Figure 45 shows the detailed transition of the atomic configurations along the MEP for transition FeTiH_tran_a, FeTiH_tran_b and FeTiH_tran_c. By comparing the detailed transition of the atomic configurations shown in Figure 43-Figure 45 and the ones in Figure 17, Figure 18, and Figure 20, it clearly shows that the transition paths located by the curve swarm and concurrent search algorithm are different.



(a)

(b)

(c)

Figure 42: Minimum energy paths from curve swarm search algorithm for hydrogen diffusion in the FeTiH structure starting from different initial structures with hydrogen locating at (a) the Fe-Fe bridge along the first primitive lattice vector (b) Fe-Fe bridge along the third primitive lattice vector (c) center of the (001) surface

Table 24: Summary the test results by curve swarm algorithm on FeTi-H system

	# of MEPs	# of local minima	# of saddle points	Energy for initial state E_I (eV)	Energy for saddle point E_s (eV)	Activation energy ΔE (eV)	Experiment ΔE (eV)
FeTiH_tran_a	8	9	8	–38.3649	–35.7084	2.6565	1±0.05
FeTiH_tran_b	4	5	4	–37.7664	–35.9057	1.8607	
FeTiH_tran_c	1	2	1	–38.0398	–36.9167	1.1231	

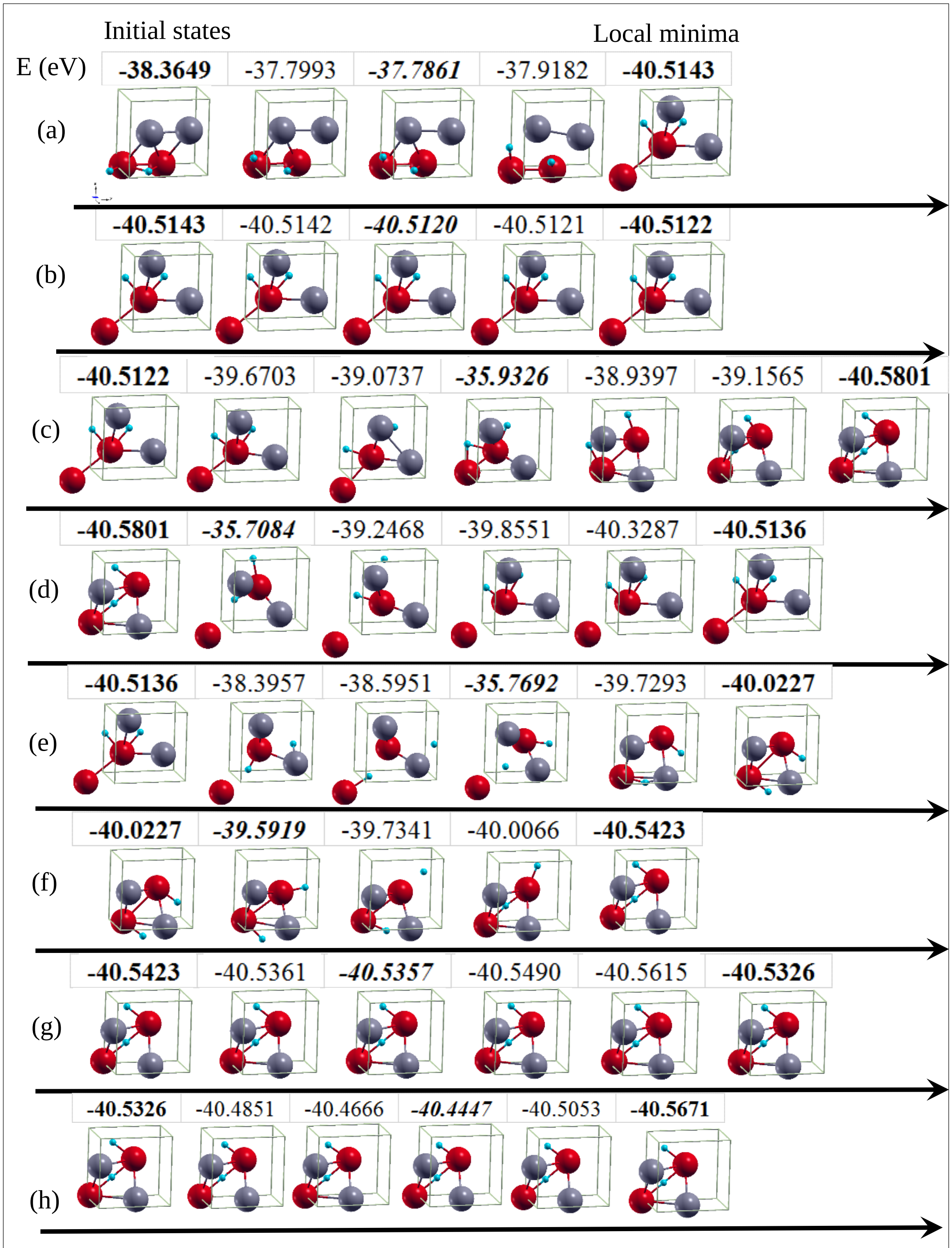


Figure 43: Detailed atomic configuration transition of hydrogen atoms in FeTi-H for FeTiH_tran_a (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 (e) MEP_5 (f) MEP_6 (g) MEP_7 (h) MEP_8 obtained by curve swarm algorithm

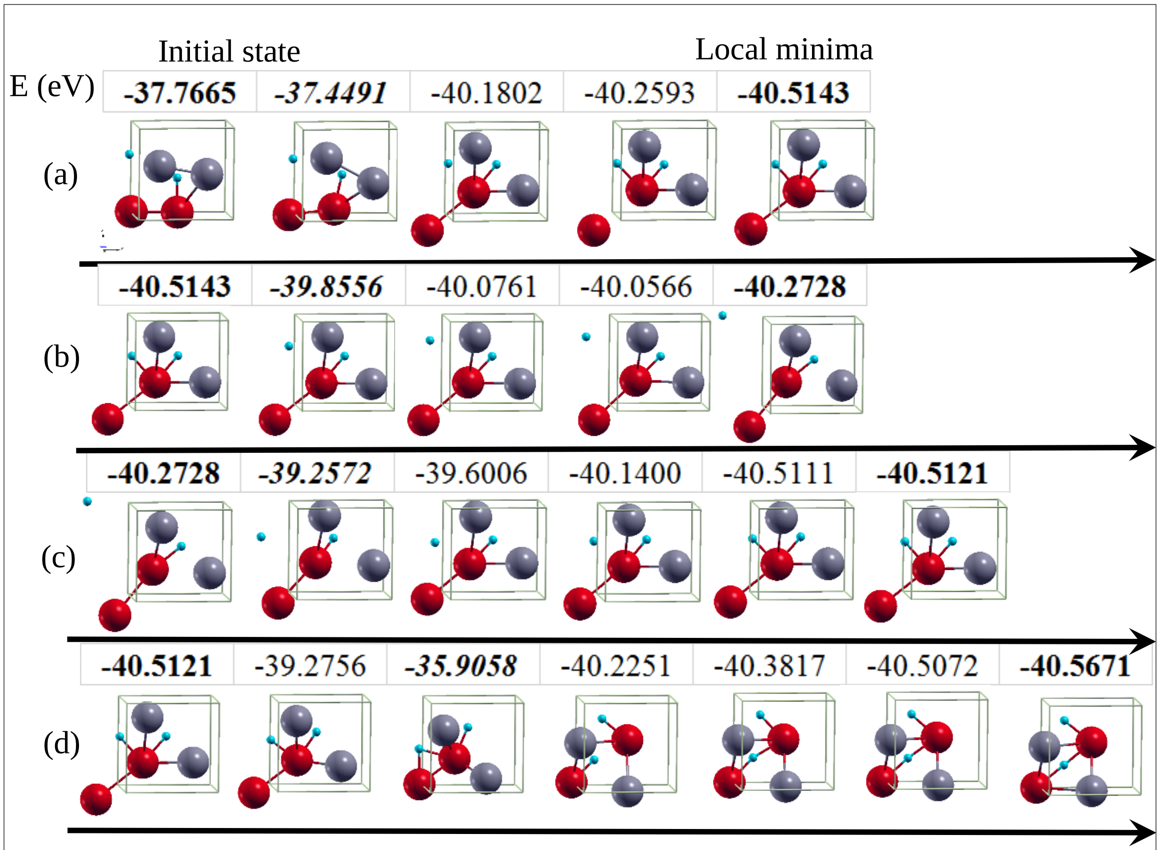


Figure 44: Detailed atomic configuration transition of hydrogen atoms in FeTi-H for FeTiH_tran_b (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 obtained by curve swarm algorithm

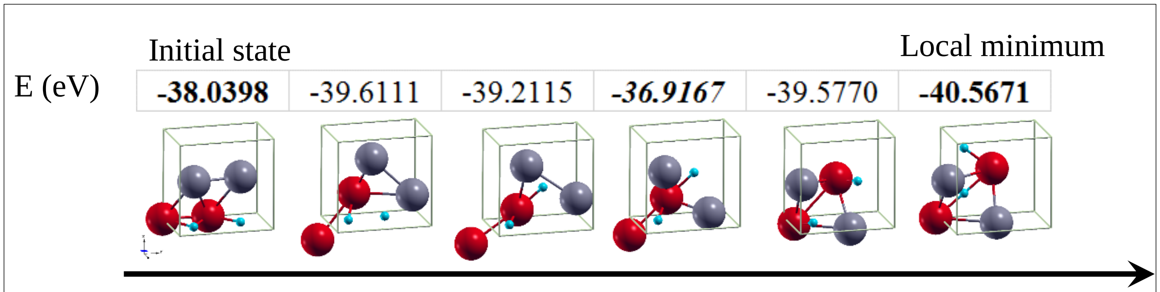


Figure 45: Detailed atomic configuration transition of hydrogen atoms in FeTi-H for FeTiH_tran_c obtained by curve swarm algorithm

4.4.4 Test Results for Fe_8-H System

In this section, the curve swarm search algorithm is used to study the hydrogen diffusion process in the pure Iron system. The detailed description about the FeTi-H can be found in Section 3.4.6.

4.4.4.1 Computational Details

Here, three initial curves are used in the test. For comparison purpose, we study the diffusion process from the same initial structure to the final structures as we did in the demonstration of the concurrent search algorithm. The initial structures for the three initial curves are the same one as shown in Figure 22 with the fractional coordinates being listed in Table 14. The final structures are the ones as shown in Figure 23(a-c) with its fractional coordinates being listed in Table 15. To maintain consistence with the previous studies, here we refer the three transitions as Fe_8H_tran_a, Fe_8H_tran_b, and Fe_8H_tran_c.

Similarly, the total energy of the system and the forces on each atom are calculated based on DFT calculation using VASP. The projector augmented wave potentials, specifically LDA potentials, and a $2\times4\times2$ k-point sampling scheme are used here. The unit cell for both initial and final structures are defined as $a=5.72$ Å, $b=2.86$ Å, and $c=5.72$ Å. The procedure described in Figure 29 and Table 17 is applied to the three initial curves to locate multiple local minima and saddle points. The collective force among curves are calculated using ‘local’ minimum distance scheme.

4.4.4.2 Computational Results

The located minimum energy paths (MEPs) for the three groups of curves are plotted in terms of total energy (eV) with respect to the reaction coordinates which are shown in Figure 46. For the first group, the algorithm locates four MEPs with five local minima and four saddle points which is shown in Figure 46(a). The global saddle point along the transition Fe_8H_a is the one on the second curve with a total energy of –74.8022 eV. The total energy for the initial state is –74.9868eV. The activation energy for the transition is the energy difference between the global saddle point and the initial state which is 0.1846 eV for Fe_8H_a.

For the second group, the algorithm locates four MEPs with five local minima and four saddle points which is shown in Figure 46(b). The total energy of the initial state is –74.9868 eV and the global saddle point is –74.7037 eV which is the one on the third curve. The activation energy for the Fe_8H_b is 0.2831 eV. For the third group, the algorithm locates two MEPs with three local minima and two saddle points which is shown in Figure 46 (c). The global saddle point is the one on the second curve with a total energy of –74.8576 eV. The total energy for the initial state is –74.9868 eV and the activation energy for Fe_8H_c is 0.1292 eV. The test results are summarized in Table 25. The total energy for the initial structure here is higher than the one obtained from the concurrent search algorithm in Section 3.4.6. For the curve swarm search algorithm, to reduce computational cost, a higher break condition (EDIFF=10^{-4}) is used, while the EDIFF is set to be 10^{-5} in the example in Section 3.4.6.

Figure 47-Figure 49 shows the detailed transition of the atomic configurations along the MEP for transition Fe_8H_a Fe_8H _b and Fe_8H _c.

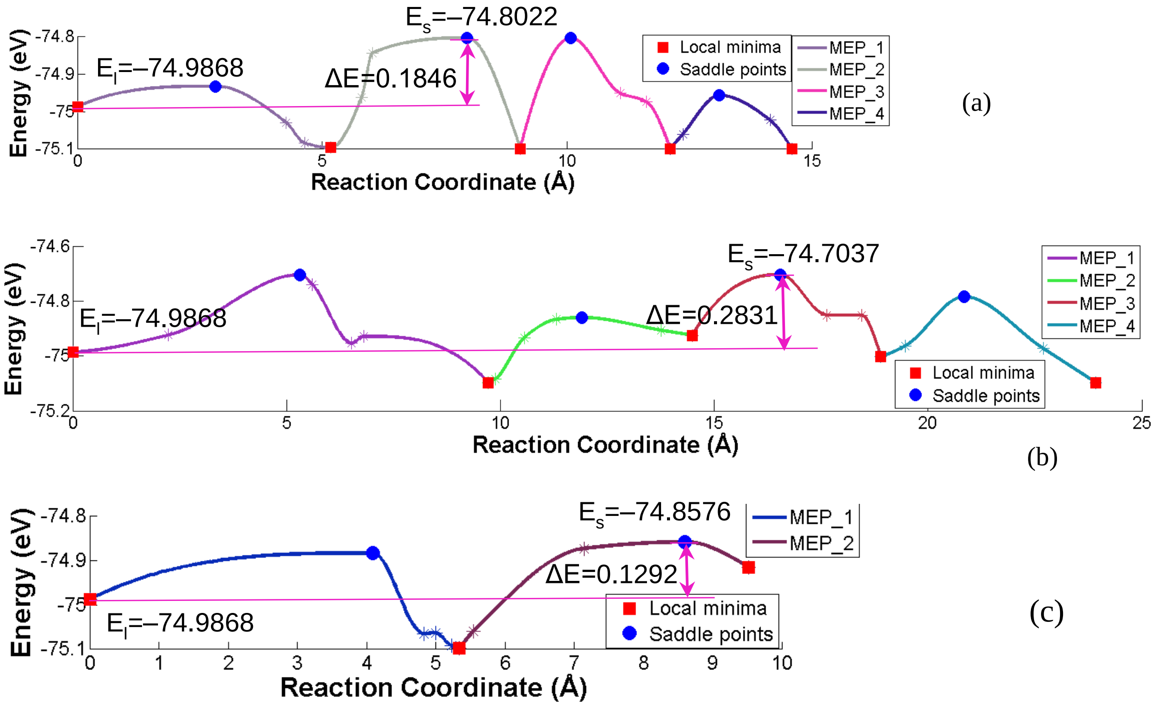


Figure 46: Minimum energy paths from curve swarm search algorithm for for hydrogen diffusion in Fe_8H structure starting from the initial structure to final structures with hydrogen residing at (a) tetrahedral site on the (100) surface (b) octahedral site on the (001) surface (c) octahedral site on the (100) surface

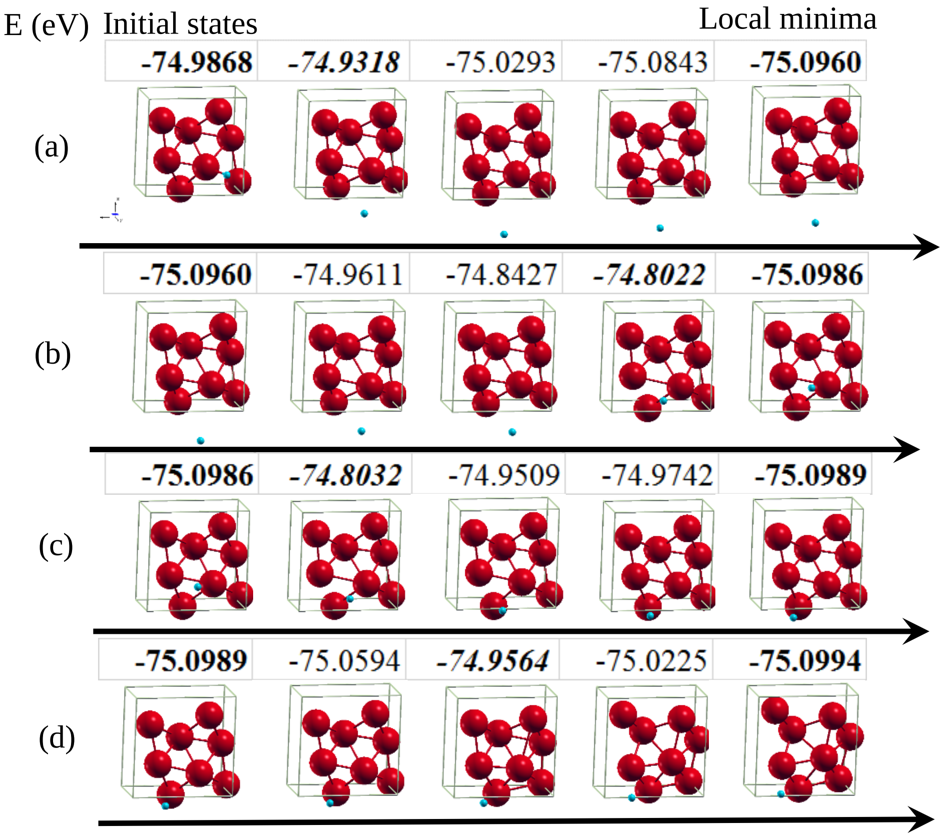


Figure 47: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_a (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 obtained by curve swarm search algorithm

Table 25: Summary of the results on Fe_8H system using curve swarm search algoirthm

	# of MEPs	# of local minima	# of saddle points	Energy for initial state E_I (eV)	Energy for saddle point E_s (eV)	Activation energy ΔE (eV)	Experiment ΔE (eV)
Fe_8H_tran_a	4	5	4	–74.9868	–74.8022	0.1846	0.035~0.142
Fe_8H_tran_b	4	5	4	–74.9868	–74.7037	0.2831	
Fe_8H_tran_c	2	3	2	–74.9868	–74.8576	0.1292	

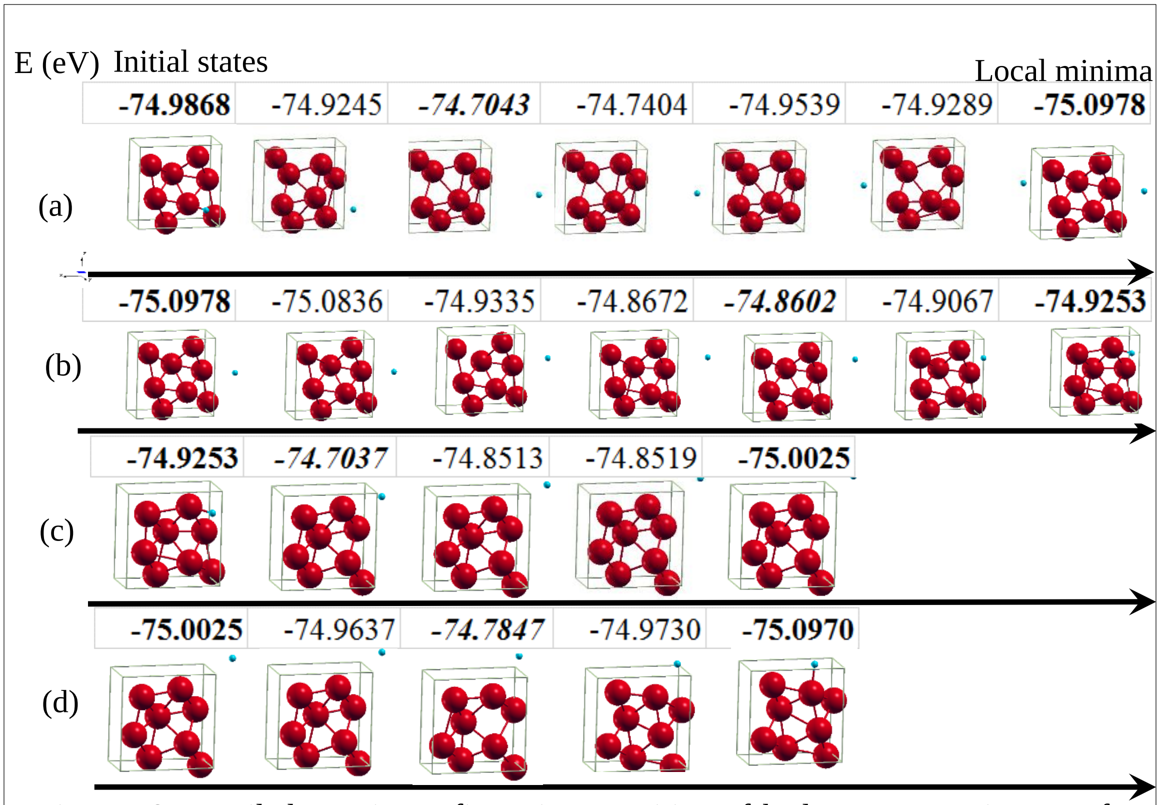


Figure 48: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_b (a) MEP_1 (b) MEP_2 (c) MEP_3 (d) MEP_4 obtained by curve swarm search algorithm

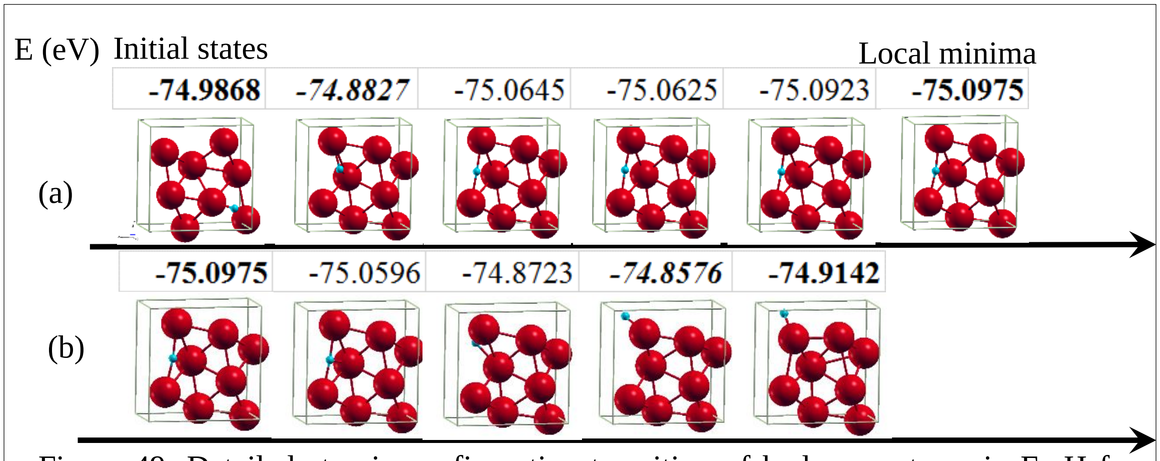


Figure 49: Detailed atomic configuration transition of hydrogen atoms in Fe_8H for Fe_8H_tran_c (a) MEP_1 (b) MEP_2obtained by curve swarm search algorithm

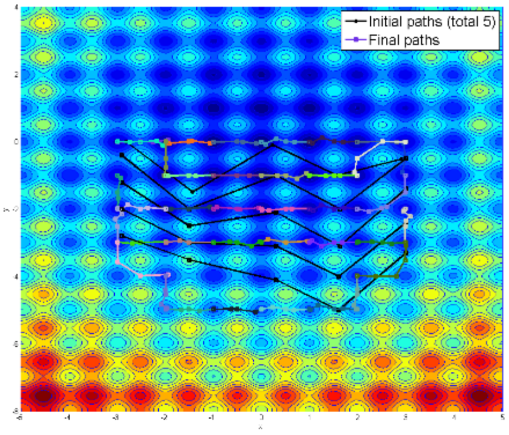
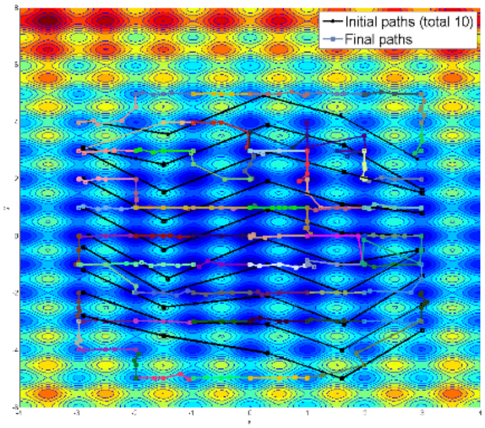
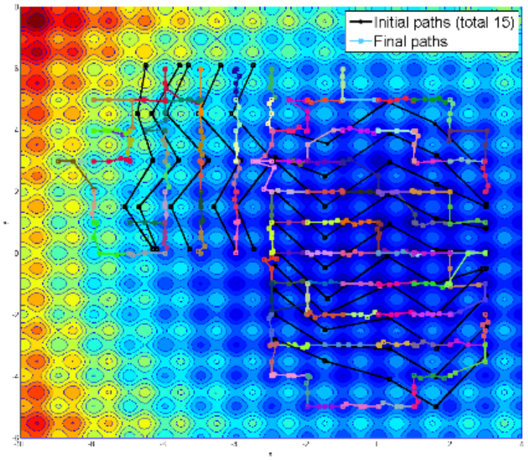
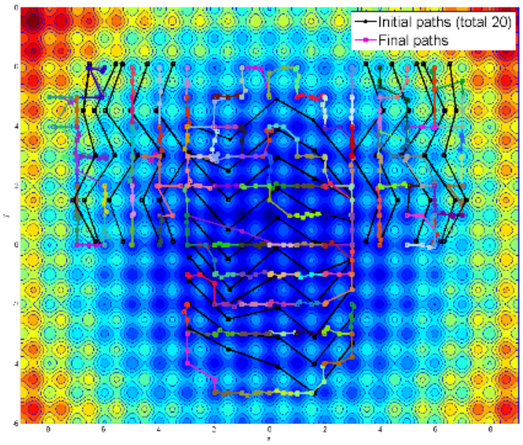
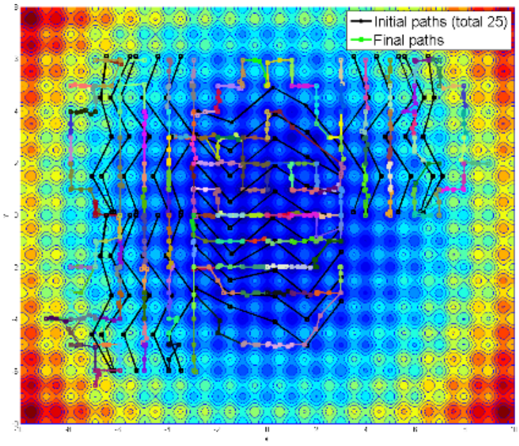
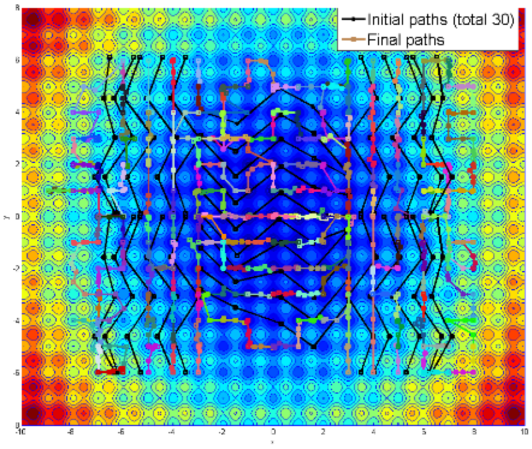
4.5 Discussion

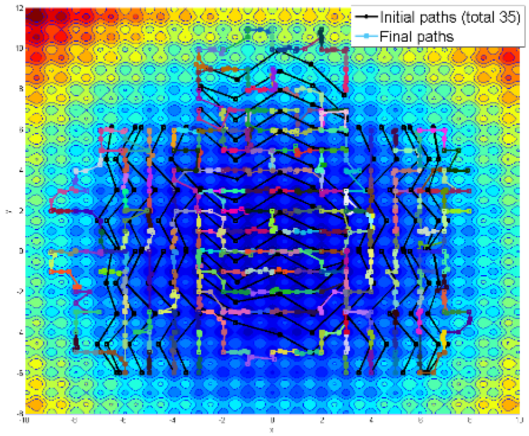
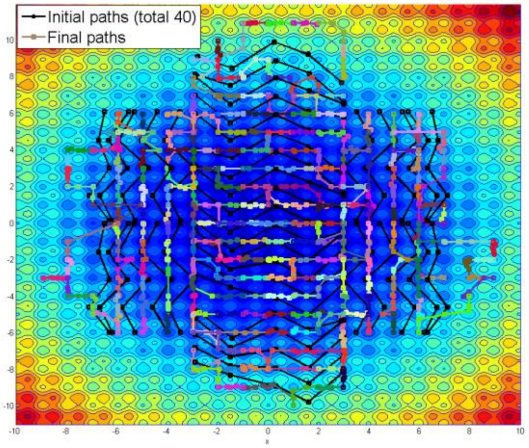
Section 4.4 demonstrates that the curve swarm search algorithm is able to thoroughly search the local minima and saddle points within one area. The proposed collective potential model works well in terms of maintaining cohesion and avoiding collision. Since computational cost and convergence are always the concerns for searching algorithms, in this section, we conduct the scalability and convergence test of the algorithm on Rastrigin function. The detailed description of the tests are presented in Section 4.5.1 and 4.5.2.

4.5.1 Scalability Test

The scalability of the curve swarm search algorithm is tested by gradually increasing the number of the initial curves on the PES. The number of the curves and the corresponding CPU time in minutes are shown in Table 26. Starting with five initial groups, we gradually increase the number of groups to ten, fifteen till forty. For two different initial curves, the required total running time for each curve may be different if the length of this two curves or the landscape that this two curves cross are different. In this case, the characteristic of the curve and the landscape of the PES together with the number of curves will determine the total running time. However, for scalability test, we should set up the experiment that the total running time is determined only by the number of curves. To eliminate the effect of the characteristic of the curve and the landscape of the PES, we choose Rastrigin function with uniformly distributed local minima and saddle points to conduct the experiment. Also, each initial curve has relative the same length as shown in those figures in Table 26. Those curves in black are initial curves and the curves in colors are located transition paths. Figure 50 shows a near quadratic relationship between the total CPU time and the total number of initial curves (i.e. number of groups) which is acceptable in terms of computational efficiency.

Table 26: Detailed results of the scalability test

Number of curves	Total CPU time(min)	Number of curves	Total CPU time (min)
5	21.67	10	52.93
			
15	102.31	20	132.77
			
25	203.48	30	339.38
			

35	439.28	40	451.31
			

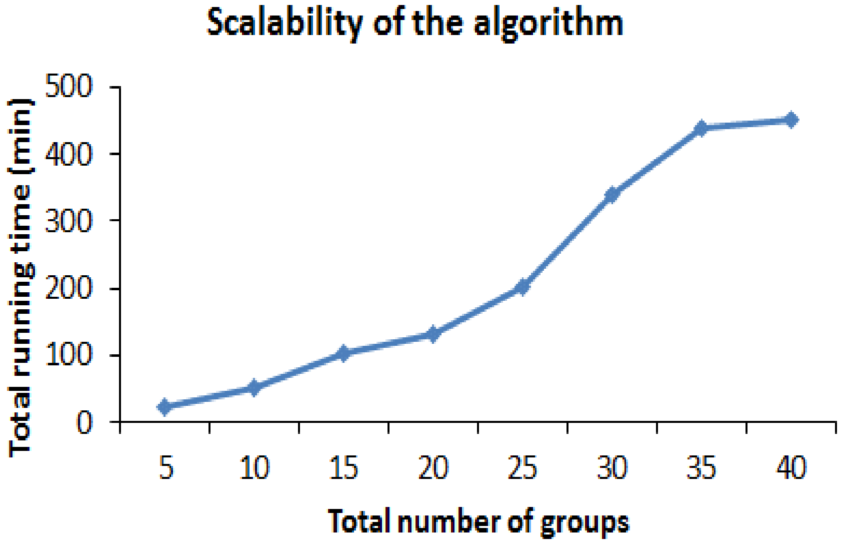
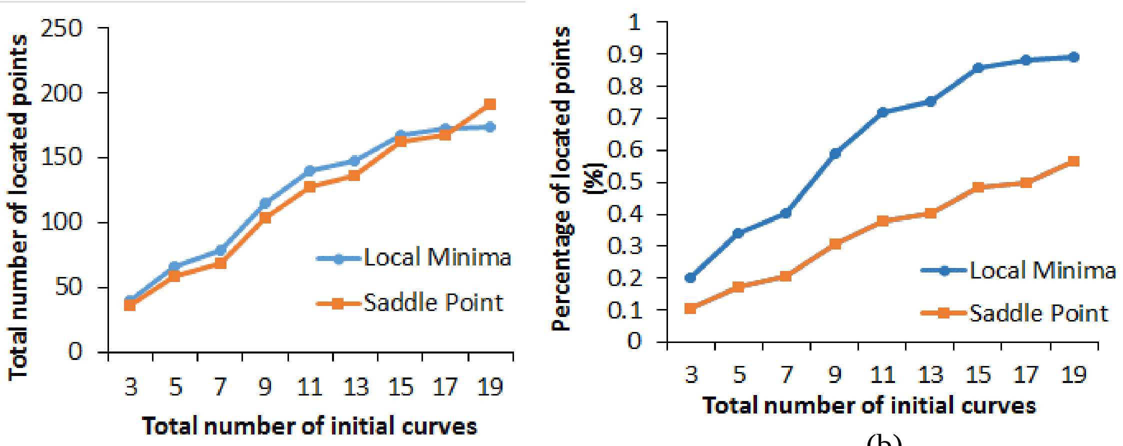


Figure 50: The total CPU running time with respect to the total number of groups.

4.5.2 Convergence Test

The convergence rate of the algorithm is also tested by gradually increasing the number of initial curves within a fixed area. Similarly, for the convergence test, we need to eliminate the effect of the characteristic of the curve and the landscape of the PES, hence, we choose Rastrigin function as the test function and each initial curve has the same length. Also, the area is set to be $x \in [-6.5 \;\; 6.5]$ and $y \in [-7.5 \;\; 7.5]$. The located saddle points and local minima which are outside this area are not considered in the total number of located points. The total number of local minima and saddle points within this area is 195 and 338 respectively. For each number of initial curve, the total number of located local minima and saddle points is recorded. We start the test by using three initial curves, and then gradually increase the number of initial curves to five till nineteen with the same space.

The change of the total number of located local minima and saddle points with respect to the number of initial curves is plotted in Figure 51(a) in which the yellow curve with square marker is for saddle points and the light blue one with round marker for local minima. It shows that the total number of located local minima converges when the number of initial curves approaches to 17. However, the total number of located saddle points keeps increasing which means it needs more initial curves to locate all the saddle points in the area. Also, the percentage of located points with respect to the total number of initial curves is plotted as shown in Figure 51(b). It shows that the percentage of located local minima converges when the total number of initial curves approaches 17 which is consistent with the results shown in Figure 51(a). The percentage of located local minima is 88.2% when the total number of initial curves is 17, and 89.2% when the number of curves is 19. Similarly, for saddle point, more initial curves are needed in order to converge.



(a)

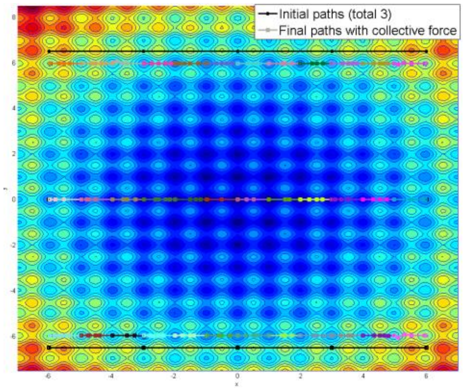
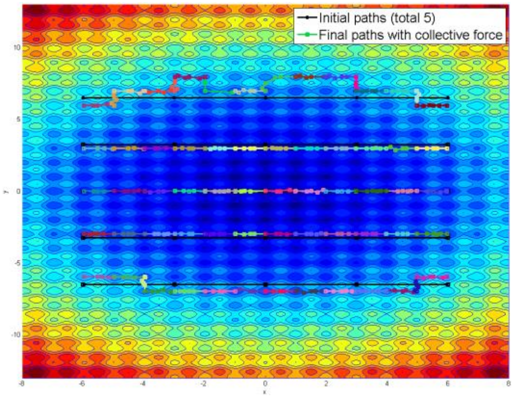
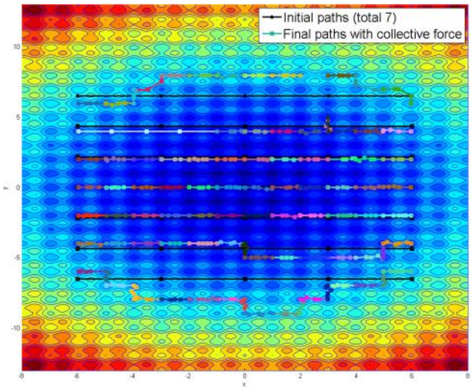
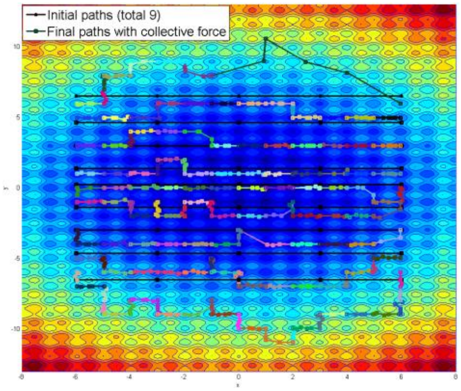
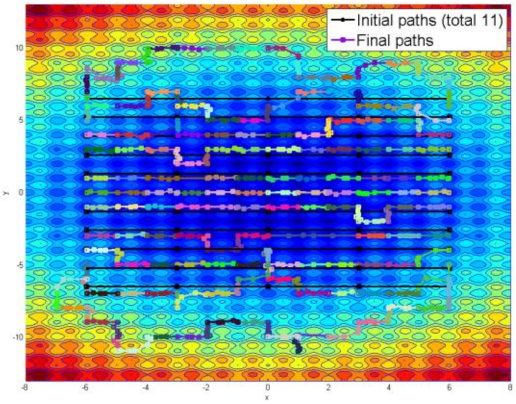
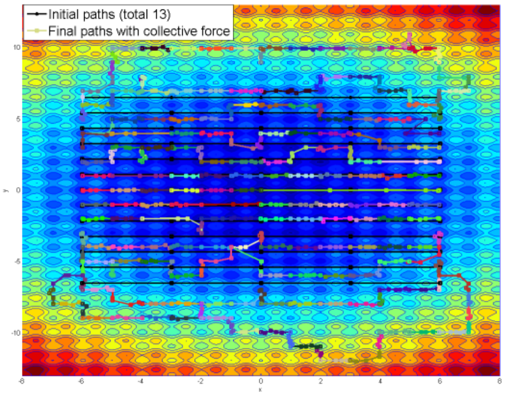
(b)

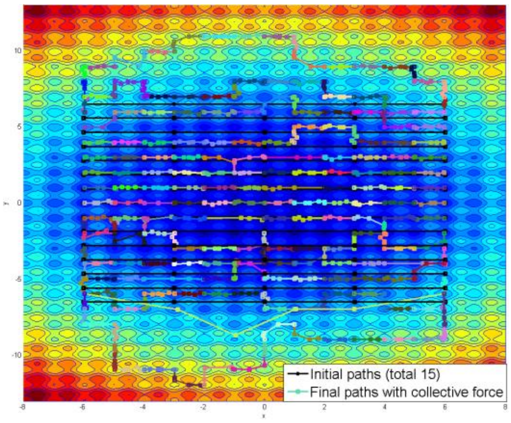
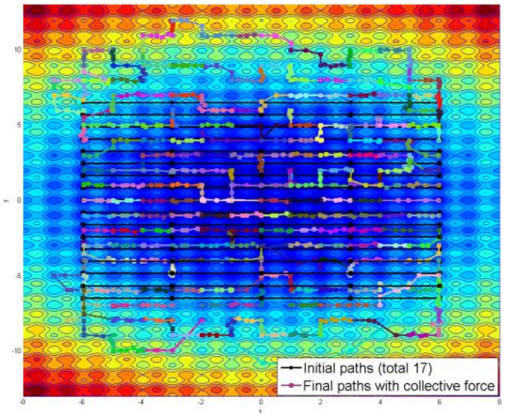
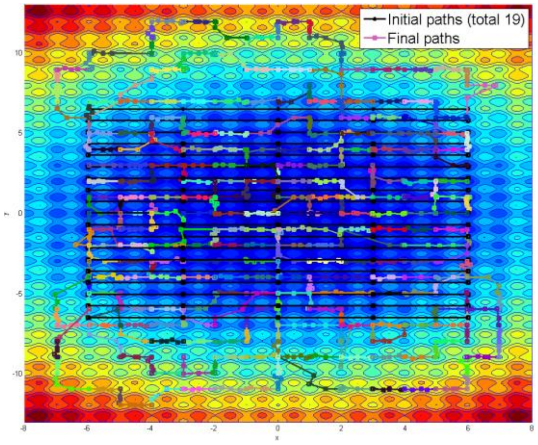
Figure 51: The relationship between the total number of initial curves and (a) total number of located local minima and saddle point (b) percentage of located points

Figure 51(b) also shows that the percentage of the located saddle points is much lower than the one of the local minima. The percentage of the located saddle points is

56.8% when the number of initial curves approaches 19. This is due to the distribution of the local minima and saddle points on the surface defined by Rastrigin function as shown in Table 27.

Table 27: Detailed results of the convergence test

<table>
<tr><th>Number of curves</th><th>Located local minima out of 195</th><th>Located saddle points out of 338</th><th>Number of curves</th><th>Located local minima out of 195</th><th>Located saddle points out of 338</th></tr>
<tr><td>3</td><td>39</td><td>36</td><td>5</td><td>66</td><td>59</td></tr>
<tr><td colspan="3">
</td><td colspan="3">
</td></tr>
<tr><td>7</td><td>79</td><td>69</td><td>9</td><td>115</td><td>104</td></tr>
<tr><td colspan="3">
</td><td colspan="3">
</td></tr>
<tr><td>11</td><td>140</td><td>128</td><td>13</td><td>147</td><td>136</td></tr>
<tr><td colspan="3">
</td><td colspan="3">
</td></tr>
</table>

<table>
<tr><td>15</td><td>167</td><td>163</td><td>17</td><td>172</td><td>168</td></tr>
<tr><td colspan="3">
</td><td colspan="3">
</td></tr>
<tr><td>19</td><td>174</td><td>192</td><td></td><td></td><td></td></tr>
<tr><td colspan="3">
</td><td colspan="3"></td></tr>
</table>

CHAPTER V
EFFICIENT AND ROBUST SADDLE POINT SEARCH

Simulating phase transformation of materials at the atomistic scale requires the knowledge of saddle points on the potential energy surface (PES). In the existing first-principles saddle point search methods, the requirement of a large number of expensive evaluations of potential energy, e.g. using density functional theory (DFT), limits the application of such algorithms to large systems. Thus, it is meaningful to minimize the number of functional evaluations as DFT simulations during the search process. Furthermore, model-form uncertainty and numerical errors are inherent in DFT and search algorithms. Robustness of the search results should be considered. In this chapter, a new search algorithm based on stochastic Kriging is presented to search local minima and saddle points on a PES efficiently and robustly. Different from existing searching methods, the algorithm keeps a memory of searching history by constructing surrogate models and uses the search results on the surrogate models to provide the guidance of future search on the PES. The surrogate model is also updated with more DFT simulation results. The algorithm is demonstrated by the examples of LEPS potential, Rastrigin and Schwefel functions with a multitude of minima and saddle points. At the end, the algorithm is applied to study the real material system.

5.1 Overview of the Saddle Point Search Algorithm Using Kriging Method

In this section, how the stochastic Kriging method is used to construct the metamodel and integrated with the concurrent search algorithm is described. The goal is to improve the efficiency and robustness of the searching algorithm. Different from the classical Kriging methods, which select the point with maximum square error as the next sampling point to improve the accuracy of the surrogate model, here a new sampling scheme is developed to serve the purpose of locating local minima and saddle points.

Similar to the original concurrent search algorithm, the general process of the algorithm includes three stages. The major difference is that the single transition pathway search is integrated with the Kriging method. The climbing process which aims to locate saddle points also incorporate kriging. The overall flowchart of using the Kriging method for saddle point search algorithm is shown in Figure 52.

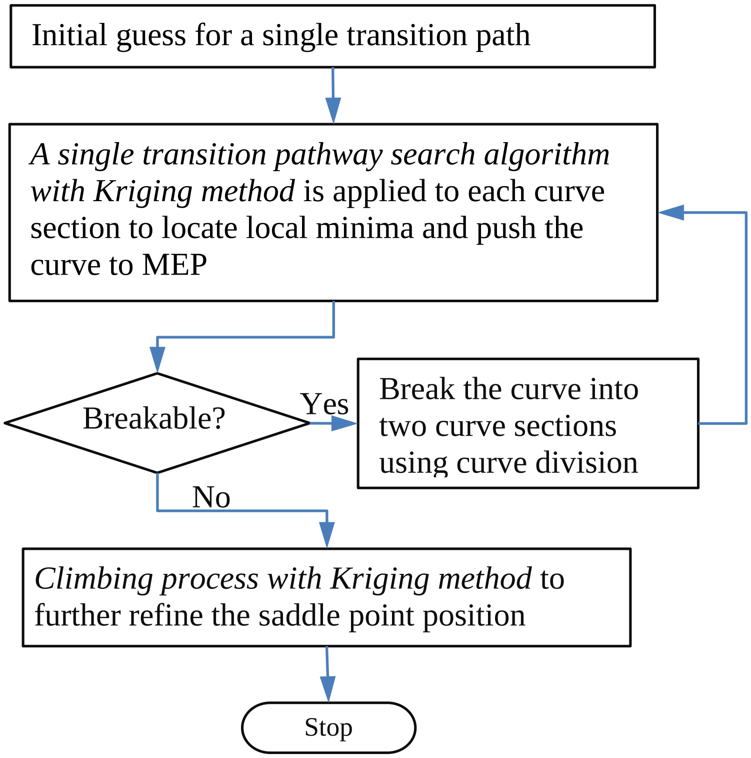

Initial guess for a single transition path
A single transition pathway search algorithm with Kriging method is applied to each curve section to locate local minima and push the curve to MEP
Breakable?
Yes
Break the curve into two curve sections using curve division
No
Climbing process with Kriging method to further refine the saddle point position
Stop

Figure 52: Flowchart for concurrent search algorithm with Kriging method

5.2 New Local Sampling Scheme for the Searching Algorithm

The major function of the surrogate model is to help decide the next sampling point in the searching process. In classical Kriging methods, the next sampling point is chosen as the one with the maximum square error. Since our primary goal is not to reduce the prediction error as fast as possible, a new sampling scheme is developed for searching local minima, which uses the criterion of choosing a sampling point with the minimum functional value around end control points.

For interpolation methods such as Kriging, the density of sample points within a region determines the accuracy of the model prediction for that region. Thus sampling needs to occur in the regions where local minima and saddle points are most likely located. The algorithm first updates the positions of control points using the real model with underlying physics, which however is computationally expensive. After collecting a predefined number of sample points which should be at least larger than the required number of design sites to construct the Kriging model, the algorithm then constructs a surrogate model using those sample points. Then the searching process is switched to the surrogate model. Since the surrogate model involves approximation error, there is no need to actually find the local minima or saddle points on the surrogate surface. After one iteration of search which includes moving both the end control points along each conjugate gradient direction and the intermediate control points along their corresponding conjugate directions, the algorithm switches the functional evaluation back to the real model. Then the searching process continues with sampling more points that are closer to the region of interest. This ‘real-surrogate-real’ iteration continues till the searching process converges.

To further improve the efficiency of the searching process, the new local sampling scheme is developed for the region of local minima which are located by the two end control points in our algorithm. On the surrogate model, for those two end control points, the algorithm determines their new positions by locating the positions with lowest energy using different search schemes which are presented in Section 5.6.1, 5.6.2, 5.6.3, and 5.6.4.

Since error is involved in surrogate model, it is important to constrain the searching area within the same basin where the starting position is. Otherwise, it will lead to jumping back and forth between two different energy basins. Consequently, it increases the computational time to converge. Here, the boundary of the searching area is defined as follows. First, the center of the searching area is set to be the current location of the end control point. Second, the area is a hypercube with the side length calculated as

$$a = \sqrt{2}c \min(d_{pre}, d_{neighbor}) \tag{5.1}$$

in which $c \in (0,1]$ is a constant; d_{pre} is the distance between the positions of the end point in current and previous iteration, i.e. $d_{pre} = \| \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)} \|$ where i is the index for the iteration; and $d_{neighbor}$ is the distance between the end point and its neighboring control point, i.e. $d_{pre} = \| \mathbf{x}_{end}^{(i)} - \mathbf{x}_{neighbor}^{(i)} \|$. The definition of the hypercube length assures that the new end control point will not jump to the position which is far away from the closest local minimum. In addition, it prevents the formation of possible loops at the end of the curve. The following two subsections present how the proposed sampling scheme and Kriging are integrated into a single transition pathway search and climbing process.

5.3 Single Transition Pathway Search with Kriging Model

In the concurrent search algorithm, the local minimum is identified using the conjugate gradient method and the path is refined by moving the intermediate control points along conjugate direction, both of which require a large number of functional evaluations for the physical model. To reduce the number of functional evaluations, the new algorithm developed in this paper keeps track of all information about the searching history (i.e. the functional values and gradients at the control points in all previous iterations) and incorporate it into the surrogate construction and searching process by the Kriging method. The algorithm embodies the search history into the construction and refinement of the surrogate model. The memory is then used to help decide the future search.

The general process for the single transition pathway search algorithm with Kriging is as follows. First, the algorithm updates the positions of the end points based on the real model using the conjugate gradient method and the positions of intermediate control points by moving them along conjugate directions until a predefined number of functional evaluations is reached. During this searching process, the functional evaluations in the line search along conjugate directions are conducted based on the real model until the number of evaluations reaches a threshold level. When more functional evaluations are required in the line search, the surrogate model is used instead so that the associated computational cost can be reduced. Second, a surrogate model is constructed or updated using all available sample points and functional values. The whole searching process on the surrogate model is based on the evaluation of the surrogate. Third, when the searching process switches from the real model to the surrogate model, the positions of the end control points are updated with the sample points with the minimum functional values within the corresponding sample areas on the surrogate model. The intermediate control points move along the conjugate directions with all the functional evaluations conducted on the surrogate model. Fourth, after one searching iteration on the surrogate model, the searching process switches back to the real model except that the second portion of the functional evaluations during each line search are still performed on the surrogate model. The overall flowchart of single transition pathway searching with the Kriging method is shown in Figure 53.

5.4 Line Search with Kriging Model

Line search is one of the critical components for the searching algorithm as it involves in all stages of the searching process. During the searching of local minima and MEP, line search is used to determine the minimum or maximum along each conjugate direction. Hence, the line search is widely used to search the optimum in a particular direction. For large systems with many degrees of freedom, a large number of line searches are required since a line search is applied to each conjugate direction. Therefore introducing Kriging will significantly improve the efficiency of the algorithm with reduced numbers of functional evaluations during line searches.

During the line search, the second portion of functional evaluations are conducted on the surrogate model. The general process for the line search is as follows. First, the algorithm conducts a couple of trial searches to determine an appropriate step length, which is based on the real model. Second, after the step length is determined, the algorithm refines the position of the control point along the direction using the determined step length. In this step, functional evaluations are also conducted on the real model. Third, the algorithm refines the surrogate model based on the results from the first and second steps and then continues updating the position of the point with functional evaluations being conducted on the surrogate model until the predefined stop criteria are satisfied.

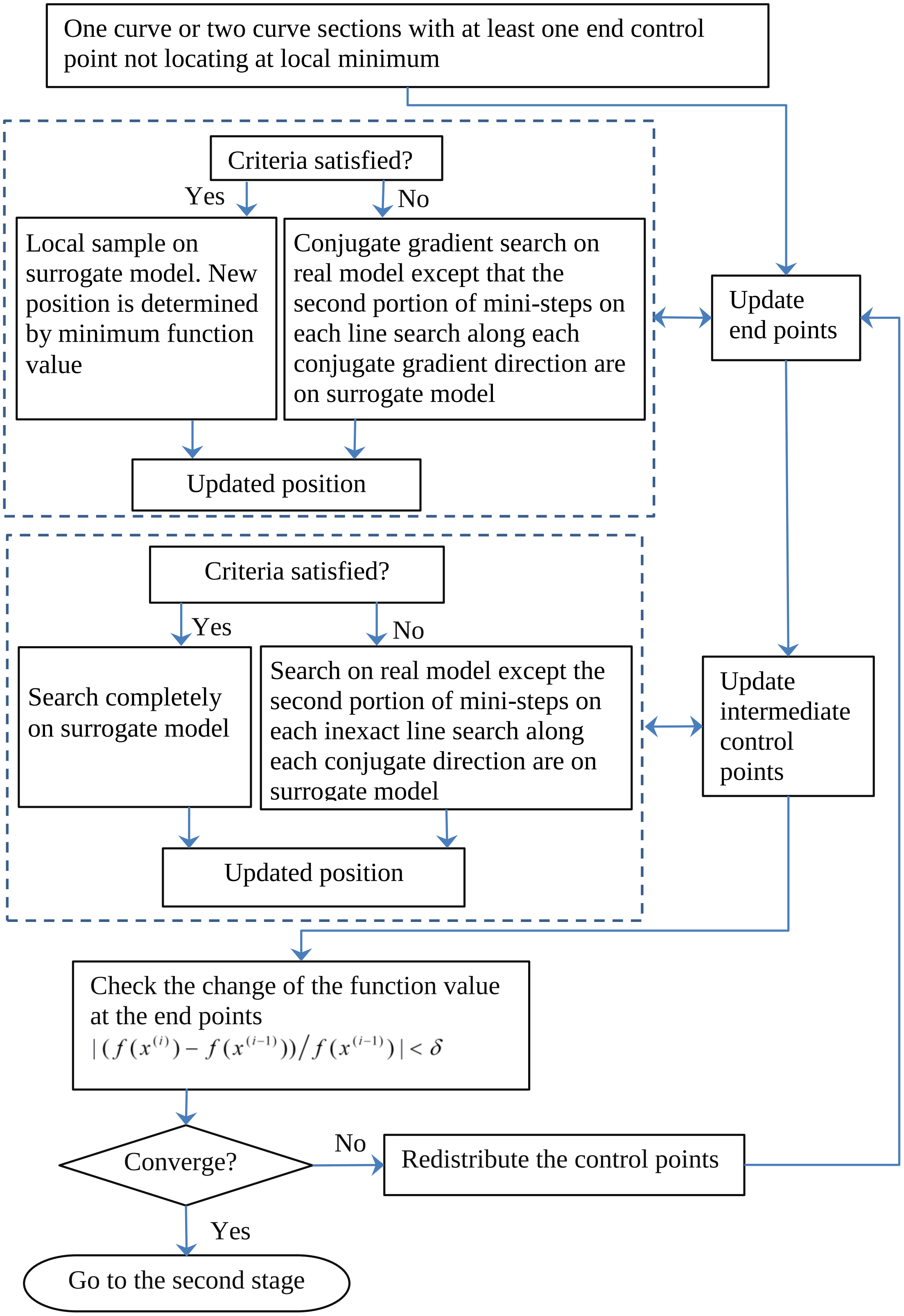

One curve or two curve sections with at least one end control point not locating at local minimum
Criteria satisfied?
Yes
No
Local sample on surrogate model. New position is determined by minimum function value
Conjugate gradient search on real model except that the second portion of mini-steps on each line search along each conjugate gradient direction are on surrogate model
Updated position
Update end points
Criteria satisfied?
Yes
No
Search completely on surrogate model
Search on real model except the second portion of mini-steps on each inexact line search along each conjugate direction are on surrogate model
Updated position
Update intermediate control points
Check the change of the function value at the end points
$|(f(x^{(i)})-f(x^{(i-1)}))/f(x^{(i-1)})|<\delta$
Converge?
No
Redistribute the control points
Yes
Go to the second stage

Figure 53: Flowchart for single transition pathway search algorithm with Kriging method

5.5 Climbing Process with Kriging Model

During the climbing stage in the concurrent searching algorithm, the control point with the maximum energy for each path climbs up to the saddle position, while all the other intermediate control points are minimized along their corresponding conjugate directions with positive eigenvalue. Integrated with Kriging, the new climbing process is as follows. First, the algorithm lets the control point with the maximum energy climb up along a conjugate direction and minimize all of the intermediate control points along their corresponding conjugate directions with positive eigenvalues. During the process, the construction of the conjugate directions is conducted on the real model except that some functional evaluations during the line search are conducted using the surrogate model. Second, after the number of iterations reaches a threshold, the algorithm refines the surrogate model and the whole searching process is conducted on the refined surrogate model.

5.6 Efficiency Demonstration with Universal Kriging Method Using Different Local Search Methods

Section 5.2 presents a new local sampling scheme to improve the efficiency of locating the local minima. In addition, to further improve the efficiency, a jumping scheme is developed to further improve the efficiency of locating the local minima on the surrogate model. Here, we present different methods to realize the jumping scheme. Each method is demonstrated using at least one of the well-known LEPS potential, Rastrigin function, and Schwefel function. We use these simple functions to help visualize the searching process and results. The Rastrigin and Schwefel functions are among the most used benchmark functions for global optimization, as defined in Table 6. The implementation is done using MATLAB and the Kriging model is constructed with DACE toolbox [216]. In this experiment, the basis polynomial function is set to be the second order and the correlation function is Gaussian correlation function.

The experiments are designed to test the accuracy and efficiency of the algorithm. The general process for the demonstration is as follows. First, we test the accuracy of the algorithm by setting a set of initial positions and test the capability of locating the local minima and saddle points. Second, we test the efficiency of the algorithm by comparing the average number of functional evaluations required to locate one transition path between the new searching algorithm with Kriging and the original concurrent searching algorithm. Since the most time-consuming portion of the saddle point searching algorithm is the functional evaluations on the real model, we use the total average number of functional evaluations as the metric to compare the efficiency in the examples.

5.6.1 Jumping Scheme by Using Uniform Sampling Method around the End Control Points

In this method, for each end control point, the algorithm draws samples uniformly within its local region which is defined by Eqn.(5.1). The sample point with the smallest functional value is selected as the next sampling point for the end control point.

5.6.1.1 Test Result for LEPS potential

The LEPS potential model describes a reaction involving three atoms A, B, and C whose motions are restricted along a straight line. This function is a benchmark function for saddle point search algorithms. Here, we use two different initial positions to test how well the new algorithm works. Figure 54(a) shows the located local minima and corresponding saddle points by using concurrent searching algorithm, while Figure 54(b) shows the results by the integrated Kriging searching algorithm. By comparing the results in Figure 54(a) and (b), it clearly shows that the new algorithm maintains a good accuracy of locating the local minima and the saddle point. The final path identified by the algorithm is very similar to the ones identified by the concurrent search algorithm.

Table 28 shows that the total number of functional evaluations for the new algorithm is much lower than the one for the concurrent search algorithm.

5.6.1.2 Test Result for Rastrigin Function

We choose a PES defined by Rastrigin function with multiple local minima and saddle points to demonstrate the capability of locating multiple saddle points. The Rastrigin function is a non-convex function which is frequently used to test global optimization algorithms. The function has a global minimum at $\mathbf{x} = (0, \cdots, 0)$ and local minima are uniformly distributed as shown by the contour plot in Figure 55. We test the new algorithm using four sets of initial positions. The results in Figure 55 show that the algorithm accurately locates local minima and saddle points. The identified final paths are also consistent with the ones identified by the concurrent search algorithm.

The total numbers of functional evaluations for the four initial positions are shown in Table 29. To minimize the influence of the initial positions on the complex functional landscape, four different initial positions are applied and the total average number of functional evaluations per final path (N/path) is shown in Table 29. The result in Table 29 shows that in average the number of functional evaluations required to locate one final path for the new algorithm with integrated Kriging is less than half of the one for the concurrent search algorithm. The algorithm works very well for the Rastrigin function.

5.6.1.3 Test Result for Schwefel Function

A third example to test the algorithm is Schwefel function which has a relatively non-uniform PES as shown in Figure 56. We follow the same procedure as for the Rastrigin function. The results are shown in Figure 56 and Table 30. The total average number of function evaluations for the new algorithm is 326 which is lower than the one (461) for the concurrent search algorithm.

Table 28: Test results for LEPS potential

Algorithm		Pos_1	Pos_2	Total	N/path
Con	N_f	2104	6264	8368	4184
	N_p	1	1	2	
Krig	N_f	643	941	1584	792
	N_p	1	1	2	

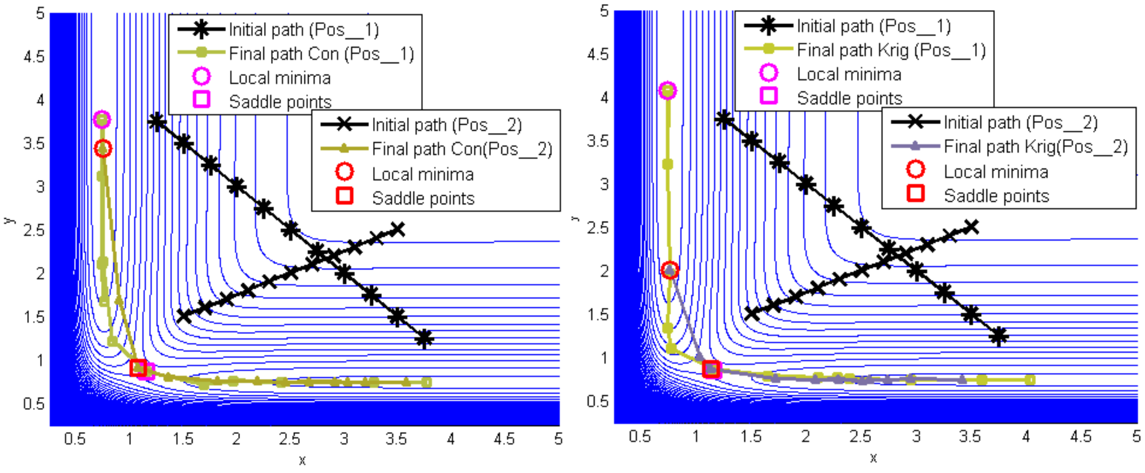
Note: "Con" represents the concurrent search algorithm; "Krig" indicates the new search algorithm integrated with kriging; "Pos_*" means the initial position * which are shown in Figure 54; N_f is the number of function evaluation at each initial position; N_p is the number of identified final paths; N/path is the average number of function evaluation to locate one final path.

Table 29: Test results for function Rastrigin

Algorithm		Pos_1	Pos_2	Pos_3	Pos_4	Total	N/path
Con	N_f	4289	4186	2120	3452	14047	502
	N_p	8	8	6	6	28	
Krig	N_f	1594	1841	1437	795	5667	218
	N_p	8	7	7	4	26	

Table 30: Test results for function Schwefel

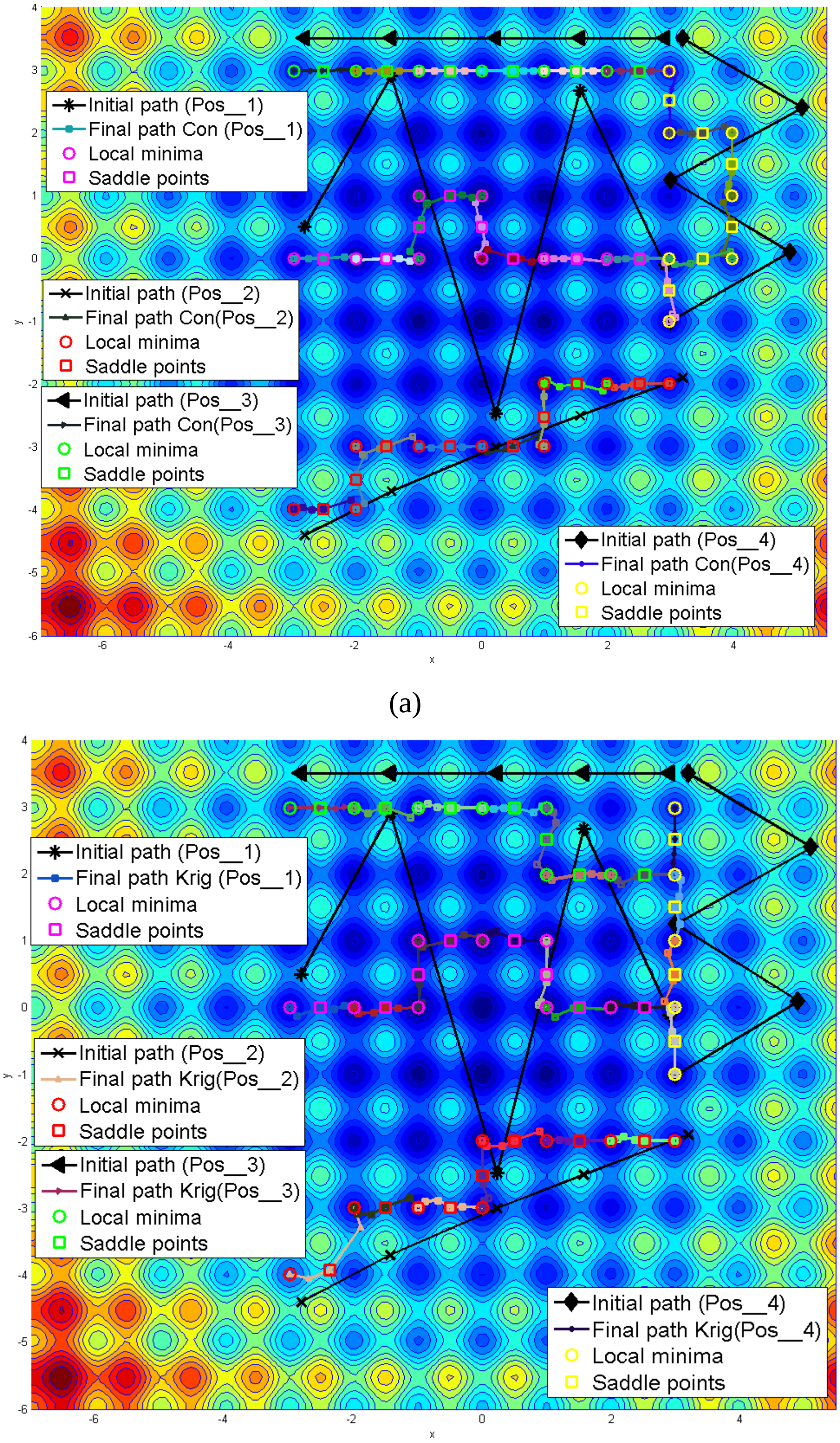
Algorithm		Pos_1	Pos_2	Pos_3	Pos_4	Total	N/path
Con	N_f	3037	3799	2742	2875	12453	461
	N_p	7	7	6	7	27	
Krig	N_f	2380	1937	1785	4028	10130	326
	N_p	7	8	8	8	31	



(a)

(b)

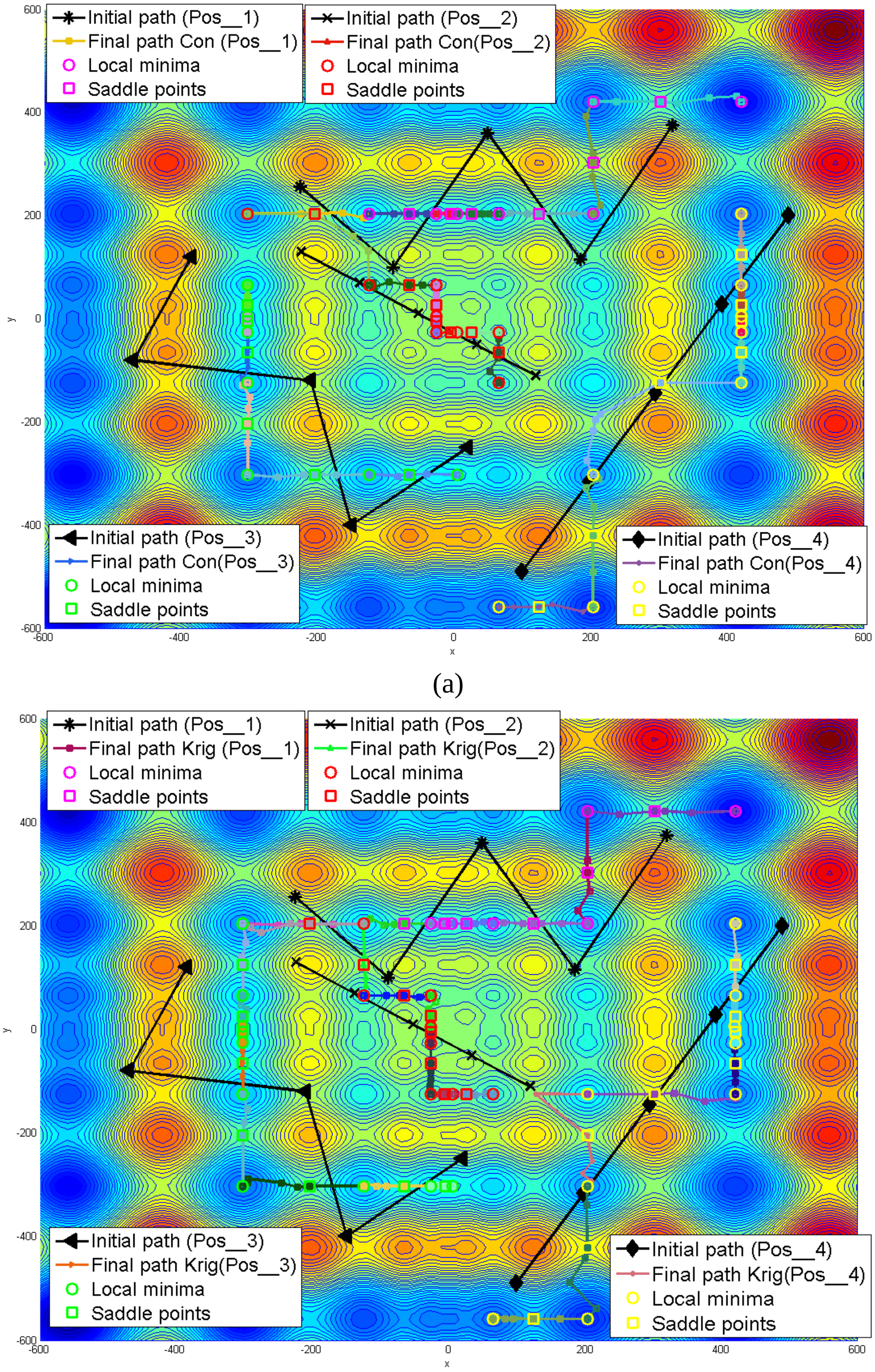
Figure 54: test results for LEPS potential (a) concurrent searching algorithm (b) integrated kriging search algorithm based on concurrent searching algorithm



(a)

(b)

Figure 55: test results for Rastrigin function (a) concurrent searching algorithm (b) integrated kriging search algorithm based on concurrent searching algorithm



(a)

(b)

Figure 56: test results for Schwefel function (a) concurrent searching algorithm (b) integrated kriging search algorithm based on concurrent searching algorithm

5.6.2 Jumping Scheme by Using Random Sampling around the End Control Points

The test results in Section 5.6.1 show that the new algorithm with uniform sampling method significantly reduces the number of functional evaluations compared to concurrent search algorithm. However, the uniform sampling on the surrogate model requires $(N\_samp)^n$ number of functional evaluations for n-dimensional surface with N_samp number of sample points in each direction on the surrogate model. The scalability of the computational time of the functional evaluation on the surrogate model is not good. Here, a new jumping scheme by using random sampling is proposed to solve the scalability issue. In this method, for each end control point, the algorithm randomly sample a predefined number of points within its local region which is defined in Eqn.(5.1). The sample point with the smallest functional value is selected as the next sampling point for the end control point.

The test results on the function Rastrigin and function Schwefel are summarized in Table 31 and Table 32 in which t is the total CPU running time in hours. For the different jumping methods, we consider both the number of functional evaluations on the real model and the efficiency of surrogate model evaluations. Thus, the total CPU time is also included here as a criterion for efficiency test. It shows that the random sampling scheme needs more functional evaluations than the uniform sampling for Rastrigin function but requires less functional evaluations for Schwefel function. Overall, those two jumping methods have the same level of efficiency in terms of function evaluations on both the real model and the surrogate model. Figure 57 and Figure 58 visualize the located local minima and saddle points using integrated Kriging method with random sampling for function Rastrigin and function Schwefel, respectively. It clearly shows that random sampling method maintains the same accuracy level of locating local minima and saddle points with the uniform sampling method.

5.6.3 Constrained Local Search on the Surrogate Model

In this method, for each end control point, the genetic algorithm (GA) is used to search the position with lowest function value on the surrogate model within the area defined in Eqn.(5.1). The located position is the next sampling point for the end control point. The test results on the Rastrigin function are summarized in Table 31. For the second and forth initial curves, the algorithm diverges during the local minima search. For the first and third initial curves, the algorithm requires twice number of function evaluations compared with the uniform sampling random sampling methods. A major part of functional evaluations are conducted during the optimization process of the initial curves, in which two local minima are located by the two end control points of the initial curve. The reason is that the GA search algorithm locates the constrained local minimum instead of finding an approximate one on the surrogate model which involves big error due to insufficient number of data points at the beginning of the search process. The surrogate model is refined when there are new data points. As a result, the local minimum on the surrogate model keeps changing which leads to the convergence problem.

5.6.4 Local Search with hybrid Line Search

Here, the conjugate gradient method is used to search local minima on the surface. In the above three jumping methods, the local minima search is switching between the real model and surrogate model. In other words, in one iteration, the search process is on the real model, and then in the other iteration, the search process is all on the surrogate model. However, for the local search with hybrid line search, the whole search process is on the real model except that the second half of the functional evaluations of the line search along each conjugate gradient direction are done on the surrogate model. The test results on the function Rastrigin and function Schwefel are summarized in Table 31 and Table 32. It shows that both the number of functional evaluations and the total CPU time per path are reduced significantly compared with the previous three jumping methods. The visualized results in Figure 59 and Figure 60 shows that the hybrid line search method maintains the same level of accuracy in terms of locating the local minima and saddle points.

Table 31: Summary of the test results for different jumping methods (Rastrigin function)

Jumping scheme		Pos_1	Pos_2	Pos_3	Pos_4	Total	N/path & t(h)/path
Uniform sampling	N_f	1594	1841	1437	795	5667	218
	N_p	8	7	7	4	26	
	t(h)	0.4	0.513	0.285	0.055	1.253	0.048
Random sampling	N_f	1810	1637	1269	995	5711	228.44
	N_p	8	7	6	4	25	
	t(h)	0.58	0.43	0.2	0.11	1.32	0.053
Constrained local search	N_f	2926	N/A	3429	N/A	6355	530
	N_p	6	N/A	6	N/A	12	
	t(h)	4.46	N/A	4.74	N/A	9.2	0.76
Local search with hybrid line search	N_f	1625	1303	1030	1224	5182	192
	N_p	8	7	6	6	27	
	t(h)	0.39	0.2	0.11	0.19	0.89	0.033

Table 32: Summary of the test results for different jumping methods (Schwefel function)

Jumping scheme		Pos_1	Pos_2	Pos_3	Pos_4	Total	N/path & t(h)/path
Uniform sampling	N_f	2380	1937	1785	4028	10130	326
	N_p	7	8	8	8	31	
	t(h)	0.55	0.28	0.069	0.379	1.278	0.041
Random sampling	N_f	1063	1830	1662	2071	6626	228
	N_p	5	7	8	9	29	
	t(h)	0.08	0.487	0.24	0.525	1.332	0.046
Local search with hydrid line search	N_f	781	1336	1091	1296	4504	187
	N_p	5	6	6	7	24	
	t(h)	0.03	0.18	0.074	0.121	0.405	0.017

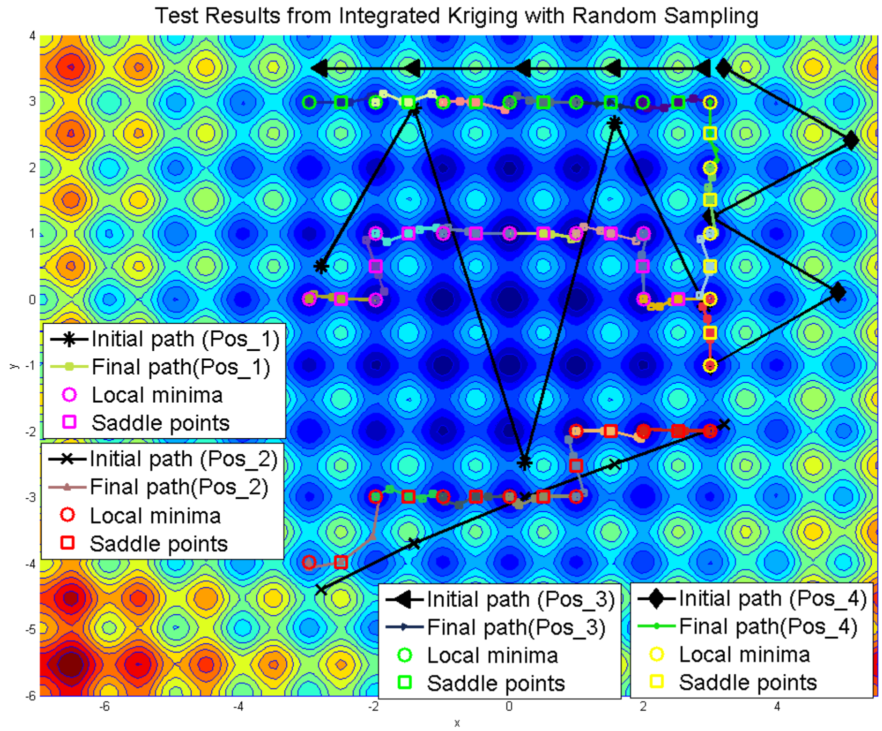


Figure 57: test results for Rastrigin function using integrated Kriging methods with random sampling method

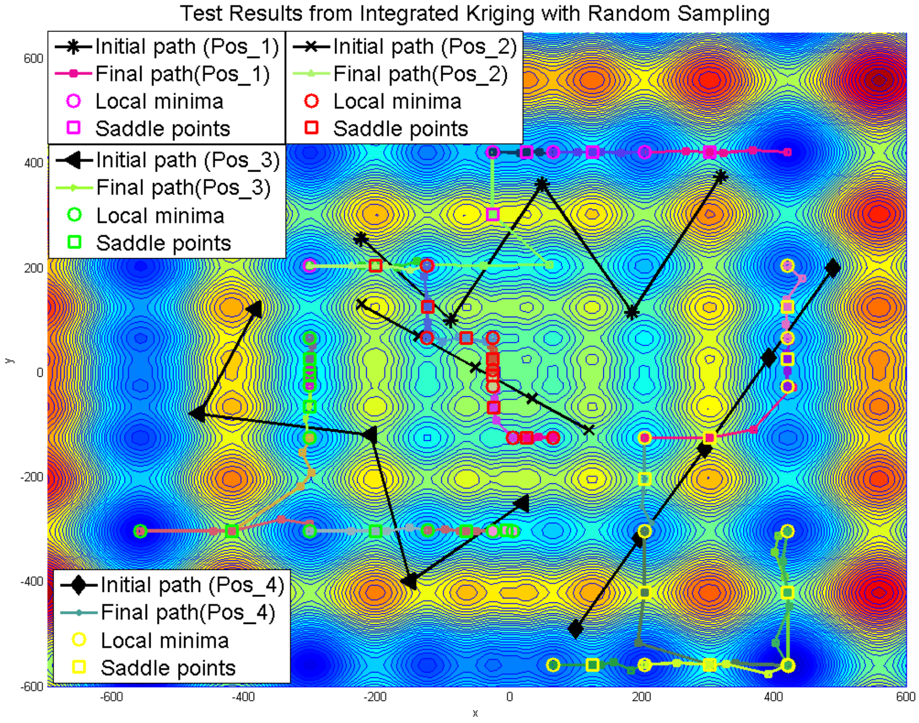


Figure 58: test results for Schwefel function using integrated Kriging methods with random sampling method

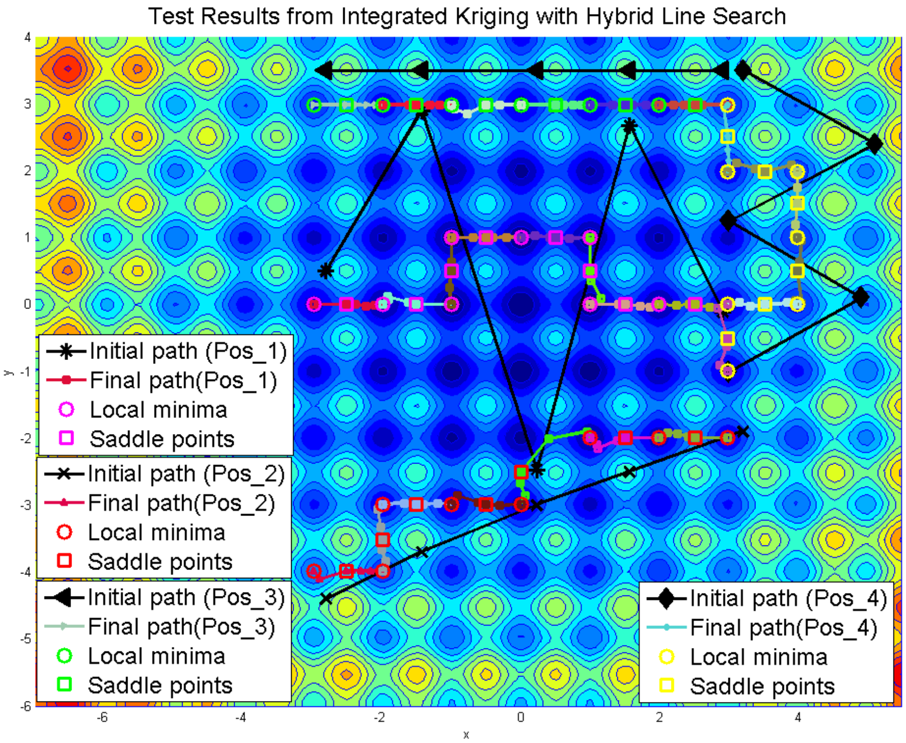


Figure 59: test results for Rastrigin function using integrated Kriging methods with hybrid line search

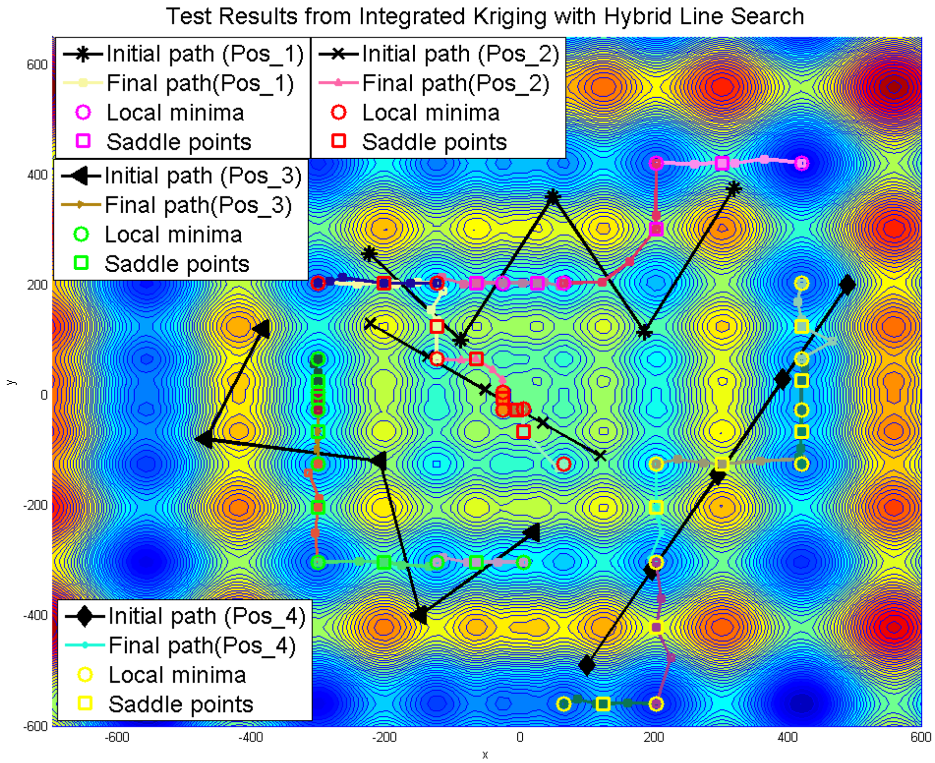


Figure 60: test results for Schwefel function using integrated Kriging methods with hybrid line search

5.7 Robust Saddle Point Search using Kriging Models of Hidden Gaussian Process

To capture the uncertainties from DFT calculation associated with the inputs, a generalized Kriging model is developed to construct the surrogate model to approximate the real PES model. The generalized Kriging model

$$\begin{aligned} z(\mathbf{x}) &= y(\mathbf{x}) + \gamma(\mathbf{x}) \\ y(\mathbf{x}) &= \mathbf{f}(\mathbf{x})^T \boldsymbol{\beta} + \varepsilon(\mathbf{x}) \end{aligned} \tag{5.2}$$

regards metamodel $y(\mathbf{x})$ as hidden and the value of $y(\mathbf{x})$ is inferred from observable $z(\mathbf{x})$, where $\gamma(\mathbf{x}) \sim \mathrm{GP}(0, \tau^2\mathbf{s}(\mathbf{x}))$ is a Gaussian process with zero mean and covariance $\tau^2\mathbf{s}(\bullet)$ in addition to $\varepsilon(\mathbf{x}) \sim \mathrm{GP}(0, \sigma^2\mathbf{r}(\mathbf{x}))$. $\varepsilon(\bullet)$ captures model-form uncertainty associated with the metamodel, whereas $\gamma(\bullet)$ represents uncertainty associated with input data. It is assumed that the two Gaussian processes are independent. $\mathbf{f}(\mathbf{x})^T \boldsymbol{\beta}$ is the mean part of $y(\mathbf{x})$ where $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \ldots, f_p(\mathbf{x})]^T$ is a vector of p basis polynomial functions, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_p]^T$ are the corresponding unknown regression coefficients.

Similar to composite Gaussian processes [189, 217], it is straightforward to derive

$$\boldsymbol{\beta} = (\mathbf{F}^T(\sigma^2\mathbf{R} + \tau^2\mathbf{S})^{-1}\mathbf{F})^{-1}\mathbf{F}^T(\sigma^2\mathbf{R} + \tau^2\mathbf{S})^{-1}\mathbf{Z} \tag{5.3}$$

and

$$\hat{z}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \hat{\boldsymbol{\beta}} + (\sigma^2\mathbf{r}(\mathbf{x}) + \tau^2\mathbf{s}(\mathbf{x}))^T(\sigma^2\mathbf{R} + \tau^2\mathbf{S})^{-1}(\mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}}) \tag{5.4}$$

by minimizing the mean squared error of the predictor of z, where **Z** is the vector of *m* observed values. **S** is the covariance matrix associate with the input data at different design sites.

For independent sampling, the errors $\gamma(\mathbf{x})$ associate with input data are assumed to be uncorrelated at different design sites i.e. $S_{ij}(\mathbf{x}_i, \mathbf{x}_j) = 0$ if $\mathbf{x}_i \neq \mathbf{x}_j$. Hence, the covariance matrix for $\gamma(\mathbf{x})$ is a $m \times m$ diagonal matrix with the diagonal elements being the variance at the corresponding design sites divided by τ^2 i.e. $\mathbf{S} = diag\{\tau_1^2/\tau^2, \tau_2^2/\tau^2, \cdots, \tau_m^2/\tau^2\}$ for m data points and $\mathbf{s}(\mathbf{x}) = 0$ at the unknown point (i.e. trial point). Thus, the prediction at the unknown point becomes

$$\hat{z}(\mathbf{x}) = \mathbf{f}(\mathbf{x})^T \hat{\boldsymbol{\beta}} + \mathbf{r}(\mathbf{x})^T (\mathbf{R} + \tau^2 \mathbf{S} / \sigma^2)^{-1} (\mathbf{Z} - \mathbf{F}\hat{\boldsymbol{\beta}}) \quad (5.5)$$

where $\mathbf{r}(\mathbf{x}) = [R(\theta_\varepsilon, \mathbf{x}, \mathbf{x}_1), \ldots, R(\theta_\varepsilon, \mathbf{x}, \mathbf{x}_m)]^T$ is the covariance at position $\mathbf{x}$, $\mathbf{R}$ is the $m \times m$ correlation matrix at the m design sites (i.e. input data points) associate with the Gaussian process $\varepsilon(\bullet)$, $\mathbf{F}$ is a $m \times p$ matrix of polynomial values $\mathbf{F} = [\mathbf{f}_1(\mathbf{x}), \ldots, \mathbf{f}_m(\mathbf{x})]^T$. When the mean structure $\mathbf{f}(\mathbf{x})^T \hat{\boldsymbol{\beta}}$ is constructed using the zero order polynomials, the Kriging model of hidden process reduces to the stochastic Kriging model [188].

The Kriging model with hidden Gaussian process cannot be integrated into the search process as described in Sections 5.2-5.5. The reason is that the surrogate model using Kriging model of hidden Gaussian process incorporates the uncertainties associated with the DFT calculation. However, the total energy for the real model based on the computer experiments is deterministic as long as all the parameter settings are fixed. In other words, the real model does not incorporate such uncertainties. Thus, there is convergence problem if the Kriging model with hidden Gaussian process is integrated directly. To solve this issue, we divided the robust saddle point search into two stages. In the first stage, the surrogate model is constructed using universal Kriging method. The search process is the same with the one described in Sections 5.2-5.5. The local search with hybrid line search is used in the local search. After the saddle point position converged on each sub curves, the algorithm moves to the second stage. In this stage, the algorithm first construct a surrogate model using Kriging model with hidden Gaussian process. Then starting from the converged saddle point position, the algorithm refines the saddle point position by searching on the stochastic surrogate model. The flow chart for the efficient and robust saddle point search is shown in Figure 61. To construct the stochastic surrogate model, the variance at each sampling points should be provided in addition to the ones required by the universal Kriging method.

In this algorithm, the first stage serves the purpose to locate the local minima and provide enough sampling points around the critical areas which in this case is the saddle point position. Thus, the stochastic surrogate model constructed based on those sampling points is accurate enough to predict the saddle point in the second stage.

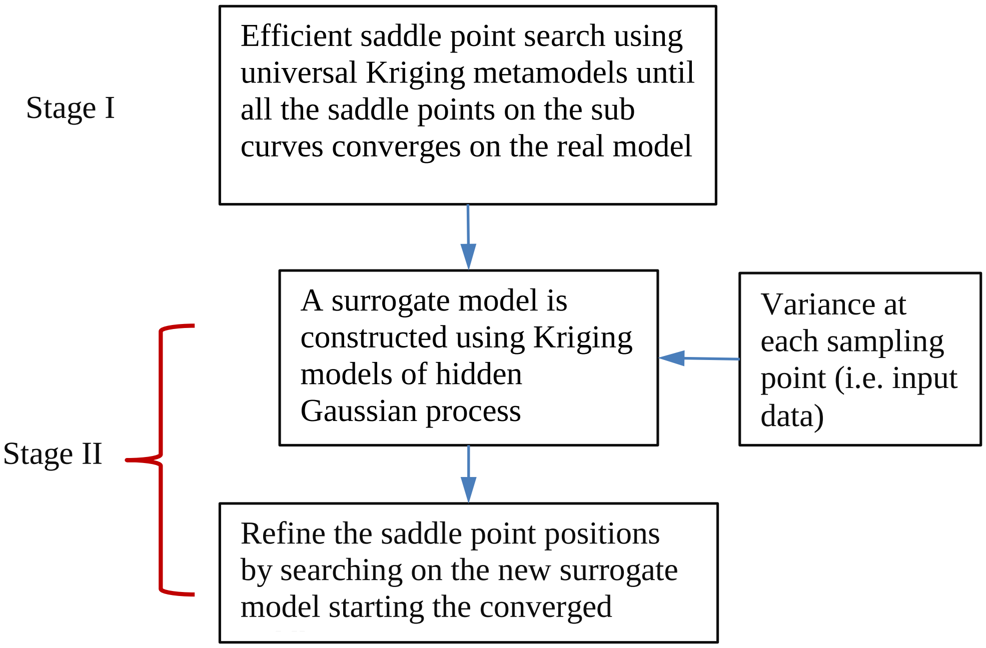

Stage I
Efficient saddle point search using universal Kriging metamodels until all the saddle points on the sub curves converges on the real model
A surrogate model is constructed using Kriging models of hidden Gaussian process
Variance at each sampling point (i.e. input data)
Stage II
Refine the saddle point positions by searching on the new surrogate model starting the converged

Figure 61: Flowchart for the efficient and robust saddle point search

5.8 Demonstration for the Robust Saddle Point Search Method

The robust saddle point search method is demonstrated using the Rastrigin and Schwefel function. All the settings are the same with the ones in 5.6.4. The variance is assumed to be 1.25 at all design sites, which means $\tau^2\mathbf{S} = diag(1.25, 1.25, \cdots)_{m\times m}$, where m is the number of design sites. For the real material system, the variance can be obtained by running DFT experiments with different settings (e.g. k-points, pseudo potentials, thresholds for the cutting off energy, or different simulation packages such as VASP, Quantum Espresso [218], etc.). For example, the variance for the total energy of a material system at a certain position can be obtained by conducting the following experiments.

1) Run the DFT simulations at that position with different k-point sampling and record the total energy of the system corresponding to the different k-point sampling.
2) Run the DFT simulations with different pseudo potentials. In those experiments, the value of the k-point sampling is set to be the one at which the total energy converges. Then record the total energy of the system corresponding to different pseudo potentials.

3) Run the DFT simulations using different cutting off energy and record the corresponding total energy of the system.
4) Run the DFT simulations using different simulation packages such as VASP, Quantum Espresso and so on. Record the total energy of the system corresponding to different simulation packages.
5) Calculate the variance of the total energy based on the data obtained from the above experiments. The calculated variance is the input for the algorithm.

For each design site, the same procedure can be applied to find the variance at that design site. After the variance at each design site is obtained, the matrix $\tau^2\mathbf{S}$ is defined with its diagonal elements being the variance at the corresponding design site. To save computational cost, it is reasonable to assume that the variances at each design sites are the same for the real material system. Therefore, the matrix $\tau^2\mathbf{S}$ can be obtained by conducting the experiments at only one design site.

5.8.1 Test Results for Rastrigin Function

In the test, the efficient saddle point search algorithm with universal Kriging models is first applied to the initial curves to search all the local minima and saddle points on the PES. After the saddle points converge, the robust saddle point search algorithm with hidden Krigign models is applied to the converged saddle points to refine their positions with the consideration of the uncertainties associated with the uncertainty from DFT. The positions of the located saddle points are shown in Figure 63(a) in which the horizontal axis is the x coordinates and the vertical axis is the y coordinates for the saddle point. Since the resolution is too low to clearly show the difference between the saddle point positions located by using efficient saddle point search algorithm and the ones refined by robust saddle point search algoirhtm, four examples of the saddle points (i.e. saddle_1, saddle_2, saddle_3 and saddle_4) are selected to zoom in which are shown in Figure 63 (b)-(e). The circle dots in the center of the rectangular boxes are the saddle points located by efficient saddle point search and the corners of the rectangular boxes are the saddle point positions located by the robust saddle point search. It clearly shows that the saddle point positions are shifted due to the uncertainties associated with the input data. The area of the rectangular boxes at each point indicates how sensitive the saddle point positions could be with respect to the input uncertainty from DFT. The larger the area is, the more sensitive the saddle point position is.

The total energy at each saddle point of Pos_1 is shown in Figure 62(a) in which the horizontal axis is the index for each saddle point and the vertical axis is the total energy. Figure 62(b)-(f) are the detailed views at each saddle point. The cross markers are the locations of the saddle points without uncertainty which are located by the efficient saddle point search algorithm and the ends of the error bar are the positions of the saddle with uncertainty which are located by the robust saddle point search. The length of the error bar at each point indicates how sensitive the results could be with respect to the input uncertainty. The larger the length is, the more sensitive the result is at that point.

The positions and the total energy for the rest of the saddle points without and with robust saddle point search are listed in Table 33. The total time for the efficient and robust saddle points at different positions are summarized in Table 35. It shows that the saddle point refinement process using the hidden Kriging surrogate model does not add much extra running time.

5.8.2 Test Results for Schwefel Function

Similar to Rastrigin function, first the efficient saddle point search algorithm is applied to the initial curves. Then the robust saddle point search algorithm is used to refine the located saddle point positions. Figure 64 shows that the positions at the four saddle points are shifted after incorporating the uncertainties from the input data into the search process. Figure 64(b)-(e) show that the first saddle point position (saddle_1) is very sensitive to the uncertainties in the y direction, the second saddle point position (saddle_2) is very sensitive to the uncertainties in the x direction, the third one (saddle_3) is less sensitive to the uncertainties than the others, and the fourth one (saddle_4) is equal sensitive in both x and y directions.

The effect of the uncertainties to the total energy at each saddle point for Pos_1 is shown in Figure 65. Similarly, the length of the error bar indicates how sensitive the total energy could be with respect to the uncertainties. The positions and the total energy for the rest of the saddle points without and with robust saddle point search are listed in

Table 34. The total CPU time in Table 35 shows that the refinement process does not require much time.

Table 33: comparison between the located saddle point without and with robust saddle point search for Rastrigin function at Pos_1

Saddle #	Position		Total energy	
	without	with	without	with
1	(–2.5127,0)	(–2.5121, –0.0185)	26.2818	26.1913
2	(-1.5046,0)	(-1.5083, -0.0112)	22.2596	22.1877
3	(-0.9950, 0.5026)	(-0.9959, 0.5024)	21.2462	21.2733
4	(-0.5027, 0.9950)	(-0.5027, 1.0004)	21.2462	21.2388
5	(0.5027, 0.9950)	(0.5023, 1.0089)	21.2462	21.2164
6	(0.9946, 0.5049)	(0.9934, 0.5028)	21.2452	21.2506
7	(1.5076, 0)	(1.5086, -0.0053)	22.2615	22.2338
8	(2.5127,0)	(2.5116, -0.0093)	26.2818	26.2542

Table 34: comparison between the located saddle point without and with robust saddle point search for Schwefel function at Pos_1

Saddle #	Position		Total energy	
	without	with	without	with
1	(-203.8975, 203.6150)	(-203.8320, 203.4696)	837.9463	837.9246
2	(-65.5231, 203.961)	(-65.6708, 202.9309)	699.7277	699.3544
3	(25.8693, 203.7916)	(27.0224, 203.2168)	660.334	660.1138
4	(203.6463, 301.8220)	(203.6786, 302.5962)	936.4825	936.4834
5	(302.4806, 420.9720)	(302.4599, 420.7908)	719.5250	719.6925

Table 35: Summary of the total CPU time (h) for the robust saddle point search

	Pos_1	Pos_2	Pos_3	Pos_4
Rastrigin function	0.452	0.267	0.135	0.235
Schwefel function	0.073	0.242	0.099	0.195

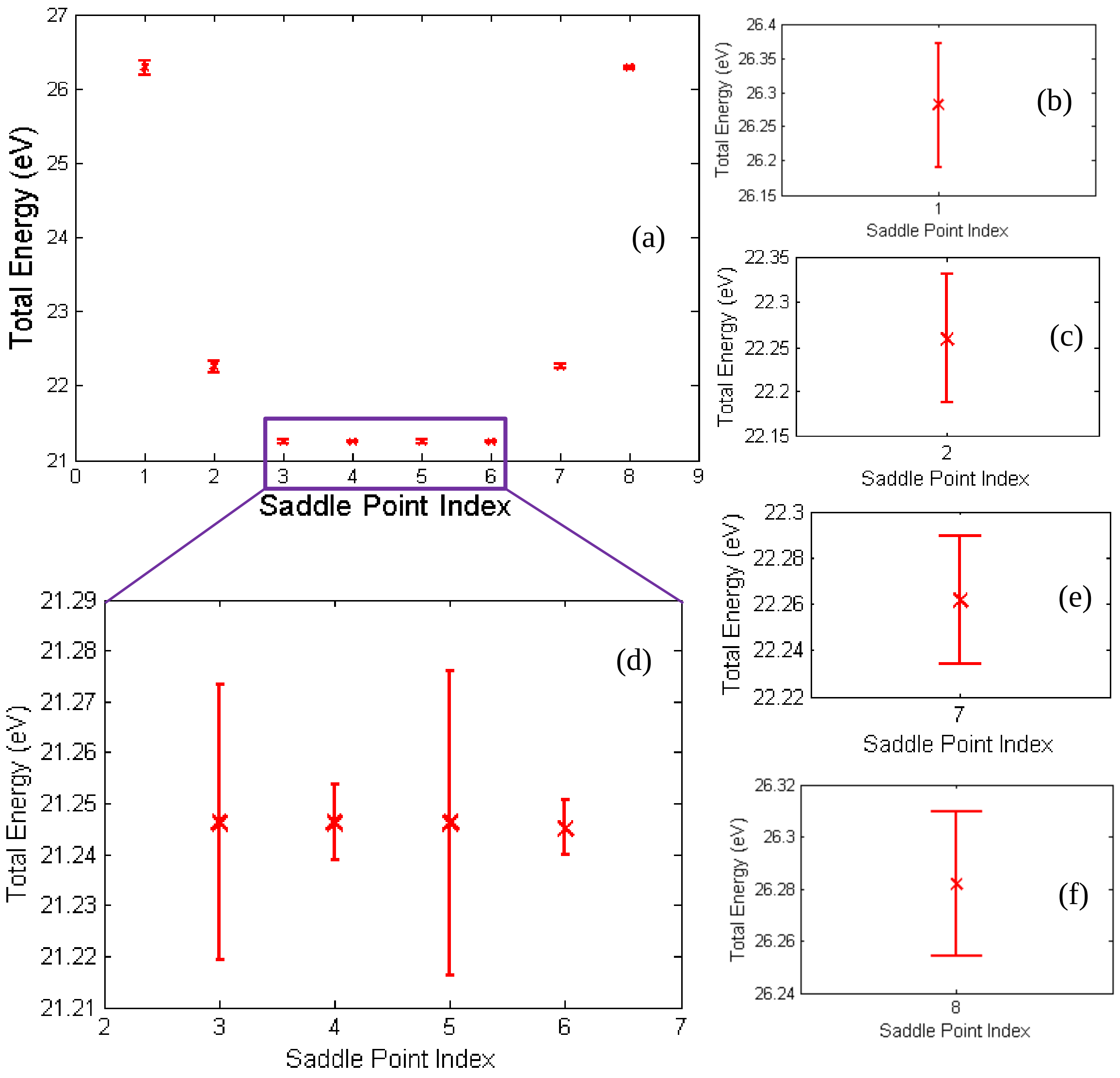


Figure 62: comparison of the total energy at the saddle points for Pos_1 located by the efficient saddle point search algorithm without (cross position) and with (the end of the error bar) robust saddle point search algorithm (Rastigin function)

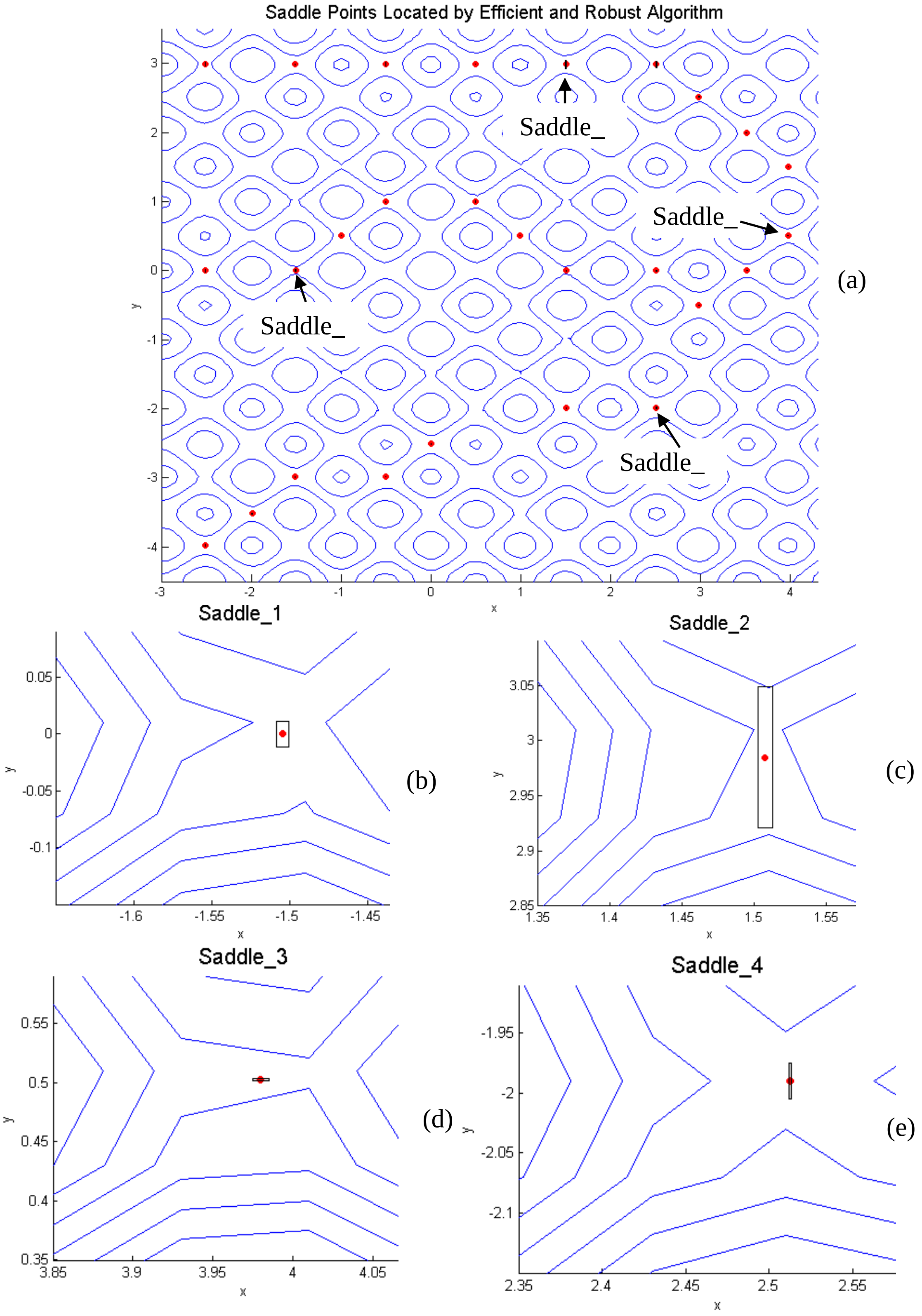


Figure 63: comparison of the saddle points position for Pos_1 located by the efficient saddle point search algorithm without (circle points) and with (the corner of the rectangular error bar) robust saddle point search algorithm (Rastigin function)

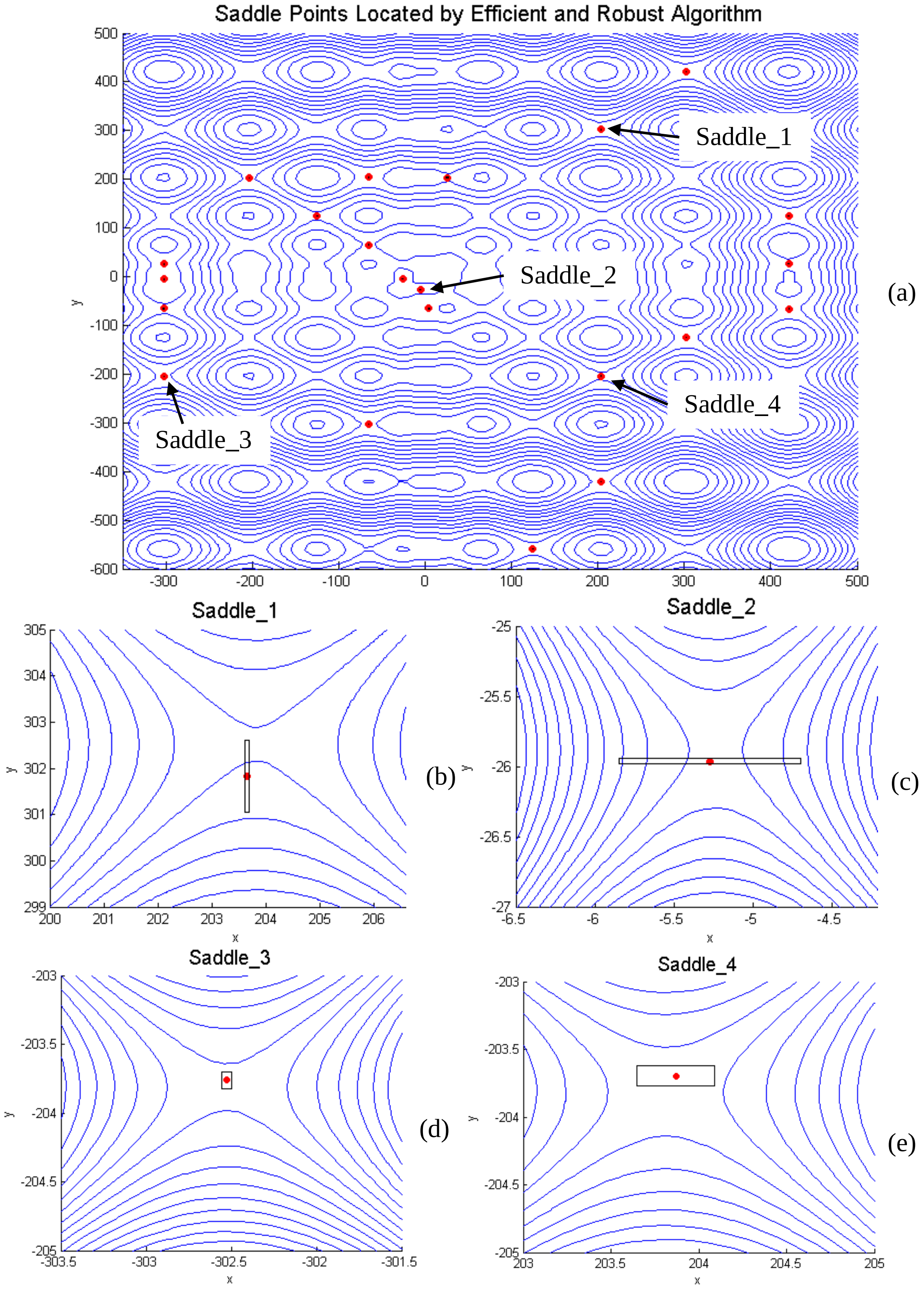


Figure 64: comparison of the saddle points position for Pos_1 located by the efficient saddle point search algorithm without (circle points) and with (the corner of the rectangular error bar) robust saddle point search algorithm (Schwefel function)

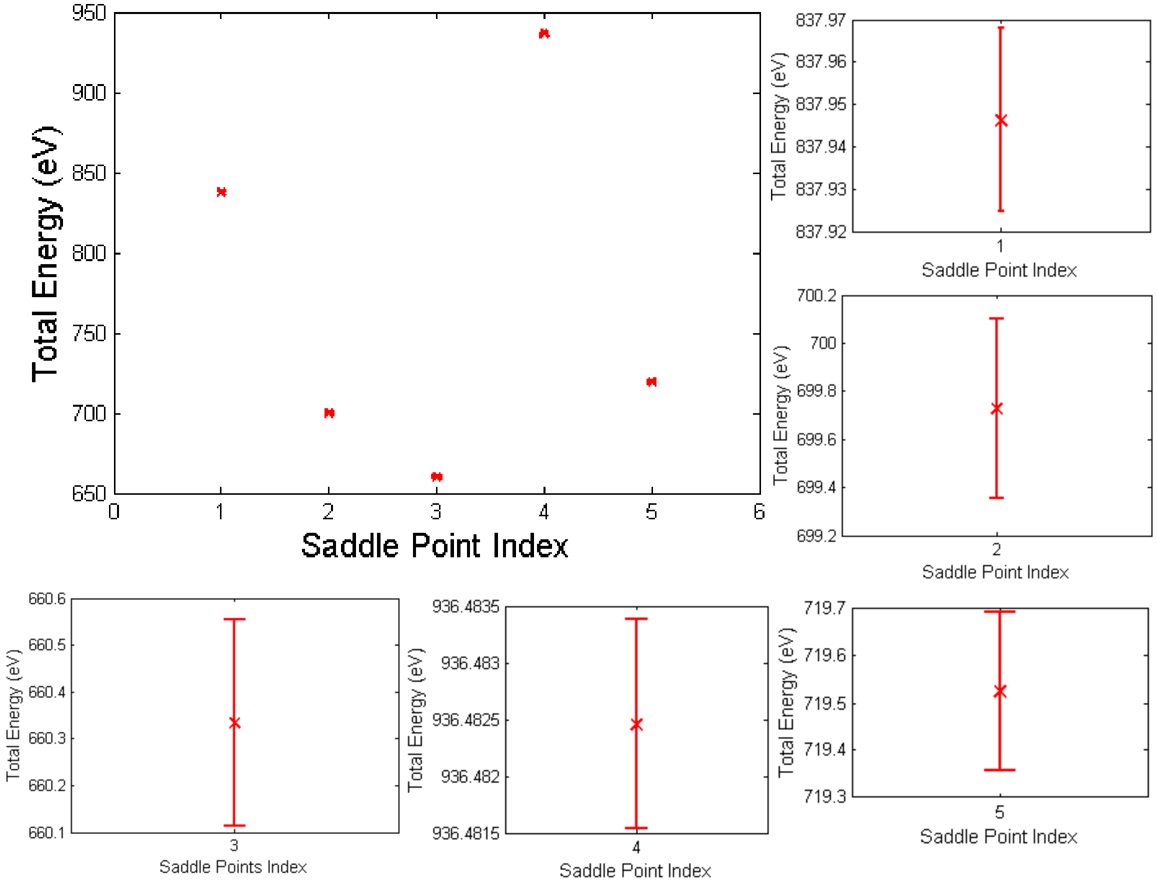


Figure 65: comparison of the total energy at the saddle points for Pos_1 located by the efficient saddle point search algorithm without (cross position) and with (the end of the error bar) robust saddle point search algorithm (Schwefel function)

5.9 Sensitivity Analysis

Here, the sensitivity of the results with respect to different variances at the design sites is studied. The value of the variance at the design sites indicates how much uncertainties are involved. The bigger the variance is, the larger the uncertainties are involved at the design sites. The expertiment is set up as follows. The robust saddle point search algorithm with hidden Kriging models is used to search the saddle points at the same position many times with different variance (from o.5 to 2.3) at the design sites. We select one saddle point and record the total energy at that point with repect to different variances. The results are plotted in Figure 66. The horizontal axis is the variance at the design sites and the vertical axis is the total energy at the selected saddle point with respect to different variances. The cross marker is the total energy at the selected saddle point without uncertainty and the end point of the error bar is the total energy at the selected saddle point with uncertainty from the input data. It clearly shows that when the variance increases, the error between the total energy with and without the certainty increases as well. This indicates that the robust saddle point search algorithm with hidden Kriging models captures the uncertainties very well. It also shows that when the variance is small which is less than one in this case, the results is relatively unaffected by the uncertainty. However, when the variance approaches to 2.3, the result will be significantly affected by the uncertainties involved in the input data. Thus, it is very important to incorporate such uncertainties to the searh process.

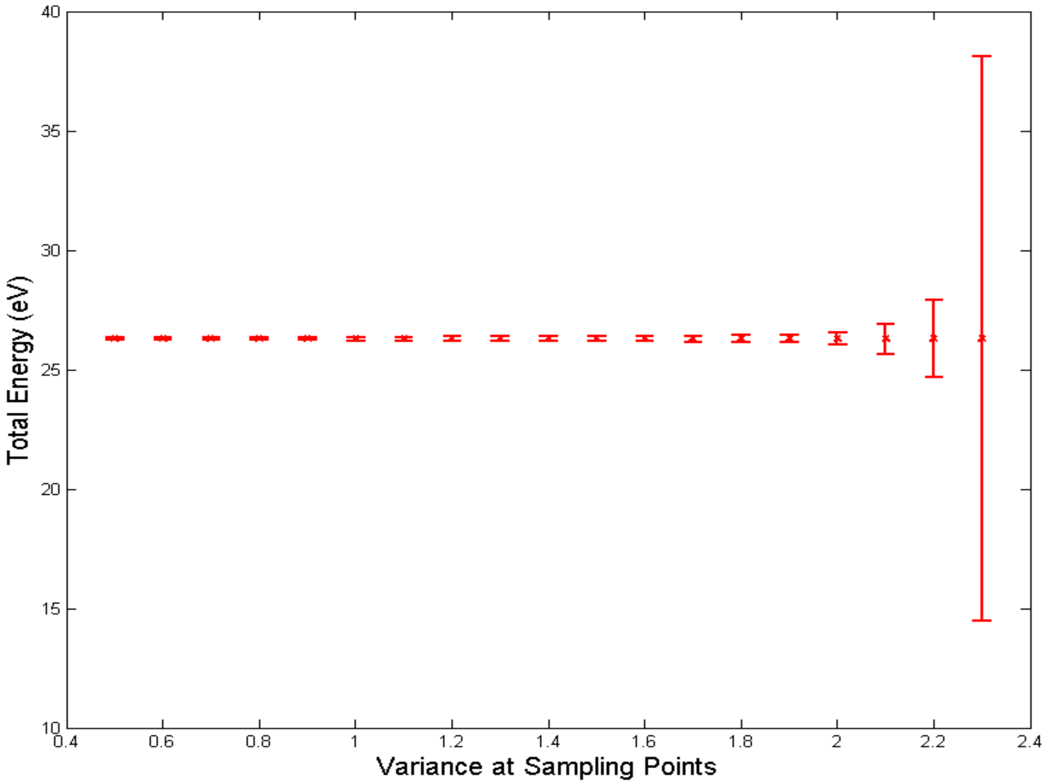


Figure 66: sensitivity test with respect to different variance at design sites using Rastrigin function

CHAPTER VI
CONCLUSIONS AND FUTURE WORKS

This chapter summarizes the methodologies and contributions of this dissertation, and proposes the future work.

6.1 Summary for the Dissertation

The ultimate goal of the dissertation is to contribute to the creation of new modeling and simulation mechanisms and tools for design of new materials for better and faster product innovation. Specifically, we developed three new saddle point search methods to provide a global view of energy landscape efficiently and robust estimation of activation energy. First, a concurrent search algorithm is developed to locate multiple local minima and saddle points along one transition path. The transition path is modeled using a parametric Bézier curve. Each control point on the curve represents an image along the transition path. The local minima are located by minimizing the two end control points of the Bézier curve using the conjugate gradient method, while the saddle points are located by moving the intermediate control points with highest energy on each sub path along conjugate directions. The algorithm includes three stages which are a single transition pathway search, multiple transition pathway search, and climbing process. In the first stage, the algorithm optimizes the initial curve to locate two end control points as well as pushes the curve to minimum energy path. If the optimized curve crosses an extra energy basin, the algorithm moves to the second stage in which the optimized curve is divided into two curve sections at a breakpoint by a curve subdivision scheme. The breakpoint is then minimized in the same way as did to the two end control points in the first stage to locate the extra local minimum. Similarly, the intermediate control points of these two curve sections are updated the same way as did to the intermediate control points of the initial curve. If one of the optimized sub curve crosses extra energy basin, the sub curve is divided into two curves. The process continues until all the sub curves only cross two adjacent energy basins. Then the algorithm moves to the third stage in which the intermediate control point with highest energy on each sub curves climbs up to locate the saddle point.

Second, a curve swarm search algorithm is developed to exhaustively search the local minima and saddle points within one area which provides a global view of the landscape of the PES. Unlike the concurrent search algorithm, the curve swarm search algorithm uses multiple groups of curves to locate multiple multi-stage transition paths within a search area, instead of one multi-stage transition path. The algorithm is inspired by the concept of flocking that describes the collective behavior of many interacting particles. It treats all newly created curves from one initial curve as one group, with their end points connected together. For each curve in one group, its two end control points are minimized to locate two local minima, while intermediate control points is updated to refine the shape of the curve. If extra minima are found along the curve, it is broken into multiple curve segments to form a multi-stage transition path. During the search process, each group communicates with its neighbor groups to avoid collision and maintain cohesion. To simulate the communication activities among curves, a collective potential model which is a function of distance between two curves is devised here. During the search process, the algorithm calculates the collective force of one curve from its neighbors based on the collective potential model. The collective force is then applied to each intermediate control point of the curve. Thus, each intermediate control point is driven by a weighted sum of the collective force and a parallel component of the true potential force along the conjugate direction.

Third, an efficient and robust search algorithm using Kriging models of hidden Gaussian process is developed to improve the efficiency of the saddle point search and robustness of the activation energy estimation. A Kriging surrogate model is introduced to approximate the real model to reduce the number of expensive evaluations of potential energy which is the most time consuming part in first principle simulation and the major reason that limits the application of the existing saddle point search algorithms to large systems. In addition, the Kriging model with hidden Gaussian process uses two Gaussian processes to capture both the model-form uncertainty in Kriging model and the input data errors that are inherent in DFT and search algorithms. Different from existing search methods, the algorithm keeps a memory of searching history by constructing surrogate models and uses the search results on the surrogate models to provide the guidance of future search on the PES. The surrogate model is also updated with more DFT simulation results. The algorithm starts searching local minima and saddle points by evaluating each function value using DFT calculation and records the information (i.e. position and the corresponding value of total energy and gradient) at each updated position. After a predefined number of iterations, the algorithm constructs a surrogate model which represents the true PES using kriging method with hidden Gaussian process based on the data points collected in the previous searching iterations. Then all the function evaluations during the searching process are conducted using the surrogate model which is much cheaper in terms of computational cost. Since the surrogate model involves uncertainties, the algorithm needs more data to update the surrogate model to better approximate the true PES. Thus, after a predefined number of searching iteration on the surrogate model, the function evaluation is conducted again using DFT calculation. This ‘real-surrogate-real’ model continues until the saddle point position converges.

6.2 Contributions

The novel contributions of the dissertation are highlighted as follows.

- A concurrent saddle point search algorithm is developed which is able to dynamically locate multiple local minima and saddle points along one transition path with multiple transition stages. A new approach based on parametric Bézier curves to represent transition paths is developed which enables an efficient images redistribution scheme based on degree elevation and reduction. A curve subdivision scheme based on the gradient and total energy information at each intermediate images is developed which makes the search algorithm dynamically insert images on the fly during the searching process.
- A curve swarm search algorithm based on the concept of flocking is developed to exhaustively explore the local minima and saddle points within one area. Thus, it provide a global view of the PES landscape. A collective potential model is developed to simulate the communication

activities among curves which effectively avoid collision and maintain cohesion during the search process.

- An efficient and robust saddle point search method using Kriging models is developed which significantly reduces the number of the expensive functional evaluations based DFT calculation. In addition, a Kriging model with hidden Gaussian process is developed to incorporate the uncertainties from both the surrogate model itself and the input data associated with model-form error and numerical error in DFT calculations. Thus, it improves the robustness of the activation energy estimation.

6.3 Future Work

In the curve swarm algorithm, the collective potential is defined as a weighted sum of pair potentials between two groups of curves. In the current scheme, the weights are assigned to be equal, which does not consider the physical implication of PES's. If the distance between two groups is very large or there exists a third group between them, a slight difference about the collective behaviors could be observed. The equal weights however do not model such differences. In the future, the model could be improved by optimizing the weight function. For example, the weights could be a function of the distance between two groups. The longer the distance is, the lower the weight should be.

The results shows that the integrated Kriging method of hidden Gaussian process works well with the simple examples such as Rastrigin function. In the future work, we will apply the robust saddle point search algorithm to large material systems with higher dimensions. The variance for the material system will be obtained by calculating the total energy for one structure using different k-point sampling, pseudopotential, and cut-off energies. In addition, the algorithm is only implemented with concurrent search algorithm. In the future, we will extend the algorithm to the curve swarm algorithm.

REFERENCES

[1] Olson, G. B., 1997, "Computational design of hierarchically structured materials," Science, 277(5330), pp. 1237-1242.

[2] Chemical Industry Vision2020 Technology Partnership, 2003, "Chemical Industry R&D Roadmap for Nanomaterials by Design: From Fundamental to Function," Available at http://www.ChemicalVision2020.org.

[3] National Science and Technology Council, 2011, "Materials genome initiative for global competitiveness "http://www.whitehouse.gov/sites/default/files/microsites/ostp/materials_genome_initiative-final.pdf.

[4] Zalba, B., Marín, J. M., Cabeza, L. F., and Mehling, H., 2003, "Review on thermal energy storage with phase change: materials, heat transfer analysis and applications," Appl. Therm. Eng., 23(3), pp. 251-283.

[5] Raoux, S., and Wuttig, M., 2009, Phase change materials: science and applications, Springer.

[6] Khudhair, A. M., and Farid, M. M., 2004, "A review on energy conservation in building applications with thermal storage by latent heat using phase change materials," Energy Convers. Manage., 45(2), pp. 263-275.

[7] Pasupathy, A., Velraj, R., and Seeniraj, R., 2008, "Phase change material-based building architecture for thermal management in residential and commercial establishments," Renewable and Sustainable Energy Reviews, 12(1), pp. 39-64.

[8] Tyagi, V. V., and Buddhi, D., 2007, "PCM thermal storage in buildings: a state of art," Renewable and Sustainable Energy Reviews, 11(6), pp. 1146-1166.

[9] Farid, M. M., Khudhair, A. M., Razack, S. A. K., and Al-Hallaj, S., 2004, "A review on phase change energy storage: materials and applications," Energy Convers. Manage., 45(9), pp. 1597-1615.

[10] Kenisarin, M., and Mahkamov, K., 2007, "Solar energy storage using phase change materials," Renewable and Sustainable Energy Reviews, 11(9), pp. 1913-1965.

[11] Wuttig, M., 2005, "Phase-change materials: towards a universal memory?," Nature materials, 4(4), pp. 265-266.

[12] Wuttig, M., and Yamada, N., 2007, "Phase-change materials for rewriteable data storage," Nature materials, 6(11), pp. 824-832.

[13] Al Hallaj, S., and Selman, J., 2000, "A Novel Thermal Management System for Electric Vehicle Batteries Using Phase‐Change Material," J. Electrochem. Soc., 147(9), pp. 3231-3236.

[14] Johnston, J. H., and Dodds, M., 2011, "The development of a flexible, re-useable thermal buffering and insulating liner for packaging temperature sensitive products," Appita Journal: Journal of the Technical Association of the Australian and New Zealand Pulp and Paper Industry, 64(2), p. 153.

[15] El Feninat, F., Laroche, G., Fiset, M., and Mantovani, D., 2002, "Shape memory materials for biomedical applications," Adv. Eng. Mater., 4(3), p. 91.

[16] Shim, H., McCullough, E., and Jones, B., 2001, "Using phase change materials in clothing," Textile Research Journal, 71(6), pp. 495-502.

[17] Salaün, F., Devaux, E., Bourbigot, S., and Rumeau, P., 2010, "Development of phase change materials in clothing part I: Formulation of microencapsulated phase change," Textile Research Journal, 80(3), pp. 195-205.

[18] Yeomans, J. M., 1992, Statistical mechanics of phase transitions, Oxford University Press.

[19] Papon, P., Leblond, J., and Meijer, P. H. E., 2006, The physics of phase transitions: concepts and applications, Springer.

[20] Voter, A. F., and Doll, J. D., 1985, "Dynamical corrections to transition state theory for multistate systems: Surface self‐diffusion in the rare‐event regime," J. Chem. Phys., 82, p. 80.

[21] Bell, S., and Crighton, J. S., 1984, "Locating transition states," J. Chem. Phys., 80, p. 2464.

[22] Schlegel, H. B., 1987, "Optimization of equilibrium geometries and transition structures," Advances in Chemical Physics; Ab Initio Methods in Quantum Chemistry-I, p. 249.

[23] Mckee, M. L., and Page, M., 1993, "Computing reaction pathways on molecular potential energy surfaces," Reviews in Computational Chemistry, Volume 4, pp. 35-65.

[24] Schlegel, H. B., 1995, "Geometry optimization on potential energy surfaces," Modern Electronic Structure Theory, D. R. Yarkony, ed., World Scientific, Singapore, pp. 459-500.

[25] Henkelman, G., Jóhannesson, G., and Jónsson, H., 2002, "Methods for finding saddle points and minimum energy paths," Theoretical Methods in Condensed Phase Chemistry, pp. 269-302.

[26] Schlegel, H. B., 2003, "Exploring potential energy surfaces for chemical reactions: an overview of some practical methods," J. Comput. Chem., 24(12), pp. 1514-1527.

[27] Olsen, R., Kroes, G., Henkelman, G., Arnaldsson, A., and Jónsson, H., 2004, "Comparison of methods for finding saddle points without knowledge of the final states," J. Chem. Phys., 121, p. 9776.

[28] Alhat, D., Lasrado, V., and Wang, Y., "A Review of Recent Phase Transition Simulation Methods: Saddle Point Search," ASME.

[29] Lasrado, V., Alhat, D., and Wang, Y., "A Review of Recent Phase Transition Simulation Methods: Transition Path Search," ASME.

[30] Kennedy, J., and Eberhart, R., "Particle swarm optimization," Proc. Neural Networks, 1995. Proceedings., IEEE International Conference on, IEEE, pp. 1942-1948.

[31] Olfati-Saber, R., 2006, "Flocking for multi-agent dynamic systems: Algorithms and theory," Automatic Control, IEEE Transactions on, 51(3), pp. 401-420.

[32] Eyring, H., and Polanyi, M., 1931, "Uber einfache Gasreaktionen," Zeitschrift für physikalische Chemie B, 12, p. 279.

[33] Laidler, K. J., and King, M. C., 1983, "Development of transition-state theory," The Journal of Physical Chemistry, 87(15), pp. 2657-2664.

[34] Vineyard, G. H., 1957, "Frequency factors and isotope effects in solid state rate processes," Journal of Physics and Chemistry of Solids, 3(1-2), pp. 121-127.

[35] Truhlar, D. G., and Garrett, B. C., 1980, "Variational transition-state theory," Accounts of Chemical Research, 13(12), pp. 440-448.

[36] Miller, W. H., Handy, N. C., and Adams, J. E., 1980, "Reaction path Hamiltonian for polyatomic molecules," The Journal of chemical physics, 72, p. 99.

[37] Chen, L., Ying, S., and Ala-Nissila, T., 2002, "Finding transition paths and rate coefficients through accelerated Langevin dynamics," Physical Review E, 65(4), pp. 042101(042101-042104).

[38] Hilderbrandt, R. L., 1977, "Application of Newton-Raphson optimization techniques in molecular mechanics calculations," Computers & Chemistry, 1(3), pp. 179-186.

[39] Cerjan, C. J., and Miller, W. H., 1981, "On finding transition states," J. Chem. Phys., 75, p. 2800.

[40] Simons, J., Joergensen, P., Taylor, H., and Ozment, J., 1983, "Walking on potential energy surfaces," The Journal of Physical Chemistry, 87(15), pp. 2745-2753.

[41] Banerjee, A., Adams, N., Simons, J., and Shepard, R., 1985, "Search for stationary points on surfaces," The Journal of Physical Chemistry, 89(1), pp. 52-57.

[42] Nguyen, D. T., and Case, D. A., 1985, "On finding stationary states on large-molecule potential energy surfaces," The Journal of Physical Chemistry, 89(19), pp. 4020-4026.

[43] Nichols, J., Taylor, H., Schmidt, P., and Simons, J., 1990, "Walking on potential energy surfaces," J. Chem. Phys., 92, p. 340.

[44] Tsai, C., and Jordan, K., 1993, "Use of an eigenmode method to locate the stationary points on the potential energy surfaces of selected argon and water clusters," The Journal of Physical Chemistry, 97(43), pp. 11227-11237.

[45] Gotō, H., 1998, "A frontier mode-following method for mapping saddle points of conformational interconversion in flexible molecules starting from the energy minimum," Chemical physics letters, 292(3), pp. 254-258.

[46] Munro, L. J., and Wales, D. J., 1999, "Defect migration in crystalline silicon," Physical Review B, 59(6), p. 3969.

[47] Kumeda, Y., Wales, D. J., and Munro, L. J., 2001, "Transition states and rearrangement mechanisms from hybrid eigenvector-following and density functional theory.: application to C10H10 and defect migration in crystalline silicon," Chemical physics letters, 341(1), pp. 185-194.

[48] Rothman, M. J., and Lohr Jr, L. L., 1980, "Analysis of an energy minimization method for locating transition states on potential energy hypersurfaces," Chemical Physics Letters, 70(2), pp. 405-409.

[49] Williams, I. H., and Maggiora, G. M., 1982, "Use and abuse of the distinguished-coordinate method for transition-state structure searching," Journal of Molecular Structure: THEOCHEM, 89(3), pp. 365-378.

[50] Chekmarev, S. F., 1994, "A simple gradient method for locating saddles," Chemical physics letters, 227(3), pp. 354-360.

[51] McIver Jr, J. W., and Komornicki, A., 1972, "Structure of transition states in organic reactions. General theory and an application to the cyclobutene-butadiene isomerization using a semiempirical molecular orbital method," Journal of the American Chemical Society, 94(8), pp. 2625-2633.

[52] Müller, K., and Brown, L. D., 1979, "Location of saddle points and minimum energy paths by a constrained simplex optimization procedure," Theoretica chimica acta, 53(1), pp. 75-93.

[53] Müller, K., 1980, "Reaction paths on multidimensional energy hypersurfaces," Angewandte Chemie International Edition in English, 19(1), pp. 1-13.

[54] Smith, C. M., 1988, "Application of a dynamic method of minimisation in the study of reaction surfaces," Theoretica chimica acta, 74(2), pp. 85-99.

[55] Smith, C. M., 1990, "How to find a saddle point," International Journal of Quantum Chemistry, 37(6), pp. 773-783.

[56] Sun, J. Q., and Ruedenberg, K., 1994, "Locating transition states by quadratic image gradient descent on potential energy surfaces," J. Chem. Phys., 101, p. 2157.

[57] Quapp, W., 1996, "A gradient-only algorithm for tracing a reaction path uphill to the saddle of a potential energy surface," Chemical physics letters, 253(3), pp. 286-292.

[58] Mousseau, N., and Barkema, G., 1998, "Traveling through potential energy landscapes of disordered materials: The activation-relaxation technique," Physical Review E, 57(2), p. 2419.

[59] Quapp, W., Hirsch, M., Imig, O., and Heidrich, D., 1998, "Searching for saddle points of potential energy surfaces by following a reduced gradient," Journal of computational chemistry, 19(9), pp. 1087-1100.

[60] Hirsch, M., and Quapp, W., 2002, "Improved RGF method to find saddle points," Journal of computational chemistry, 23(9), pp. 887-894.

[61] Anglada, J. M., Besalú, E., Bofill, J. M., and Crehuet, R., 2001, "On the quadratic reaction path evaluated in a reduced potential energy surface model and the problem to locate transition states*," Journal of Computational Chemistry, 22(4), pp. 387-406.

[62] Lin, Y., and Stadtherr, M. A., 2004, "Locating stationary points of sorbate-zeolite potential energy surfaces using interval analysis," J. Chem. Phys., 121, p. 10159.

[63] Pratt, L. R., 1986, "A statistical method for identifying transition states in high dimensional problems," J. Chem. Phys., 85, p. 5045.

[64] Elber, R., and Karplus, M., 1987, "A method for determining reaction paths in large molecules: application to myoglobin," Chemical Physics Letters, 139(5), pp. 375-380.

[65] Beck, T. L., Doll, J., and Freeman, D. L., 1989, "Locating stationary paths in functional integrals: An optimization method utilizing the stationary phase Monte Carlo sampling function."

[66] Czerminski, R., and Elber, R., 1990, "Self - avoiding walk between two fixed points as a tool to calculate reaction paths in large molecular systems," International Journal of Quantum Chemistry, 38(S24), pp. 167-185.

[67] Ulitsky, A., and Elber, R., 1990, "A new technique to calculate steepest descent paths in flexible polyatomic systems," The Journal of chemical physics, 92(2), pp. 1510-1511.

[68] Choi, C., and Elber, R., 1991, "Reaction path study of helix formation in tetrapeptides: Effect of side chains," J. Chem. Phys., 94, p. 751.

[69] Gillilan, R. E., and Wilson, K. R., 1992, "Shadowing, rare events, and rubber bands. A variational Verlet algorithm for molecular dynamics," J. Chem. Phys., 97, p. 1757.

[70] Sevick, E., Bell, A., and Theodorou, D., 1993, "A chain of states method for investigating infrequent event processes occurring in multistate, multidimensional systems," The Journal of chemical physics, 98, p. 3196.

[71] Smart, O. S., 1994, "A new method to calculate reaction paths for conformation transitions of large molecules," Chemical physics letters, 222(5), pp. 503-512.

[72] Ayala, P. Y., and Schlegel, H. B., 1997, "A combined method for determining reaction paths, minima, and transition state geometries," J. Chem. Phys., 107, p. 375.

[73] Jonsson, H., Mills, G., and Jacobsen, K., 1998, "Classical and Quantum Dynamics in Condensed Phase Simulations," World Scientific, Hackensack, NJ, pp. 385-404.

[74] Henkelman, G., and Jónsson, H., 2000, "Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points," The Journal of chemical physics, 113(22), pp. 9978-9985.

[75] Henkelman, G., Uberuaga, B. P., and Jónsson, H., 2000, "A climbing image nudged elastic band method for finding saddle points and minimum energy paths," The Journal of chemical physics, 113(22), pp. 9901-9904.

[76] Maragakis, P., Andreev, S. A., Brumer, Y., Reichman, D. R., and Kaxiras, E., 2002, "Adaptive nudged elastic band approach for transition state calculation," J. Chem. Phys., 117, p. 4651.

[77] Weinan, E., Ren, W., and Vanden-Eijnden, E., 2002, "String method for the study of rare events," Physical Review B, 66(5), p. 052301.
[78] Ren, W., 2003, "Higher order string method for finding minimum energy paths," Communications in Mathematical Sciences, 1(2), pp. 377-384.
[79] Peters, B., Heyden, A., Bell, A. T., and Chakraborty, A., 2004, "A growing string method for determining transition states: Comparison to the nudged elastic band and string methods," J. Chem. Phys., 120, p. 7877.
[80] Trygubenko, S. A., and Wales, D. J., 2004, "A doubly nudged elastic band method for finding transition states," The Journal of chemical physics, 120(5), pp. 2082-2094.
[81] Burger, S. K., and Yang, W., 2006, "Quadratic string method for determining the minimum-energy path based on multiobjective optimization," J. Chem. Phys., 124, p. 054109.
[82] Weinan, E., Ren, W., and Vanden-Eijnden, E., 2007, "Simplified and improved string method for computing the minimum energy paths in barrier-crossing events," J. Chem. Phys., 126, p. 164103.
[83] Zhu, T., Li, J., Samanta, A., Kim, H. G., and Suresh, S., 2007, "Interfacial plasticity governs strain rate sensitivity and ductility in nanostructured metals," Proceedings of the National Academy of Sciences, 104(9), pp. 3031-3036.
[84] Galván, I. F., and Field, M. J., 2008, "Improving the efficiency of the NEB reaction path finding algorithm," Journal of Computational Chemistry, 29(1), pp. 139-143.
[85] Sheppard, D., Xiao, P., Chemelewski, W., Johnson, D. D., and Henkelman, G., 2012, "A generalized solid-state nudged elastic band method," J. Chem. Phys., 136, p. 074103.
[86] Sheppard, D., Terrell, R., and Henkelman, G., 2008, "Optimization methods for finding minimum energy paths," J. Chem. Phys., 128, p. 134106.
[87] Ionova, I. V., and Carter, E. A., 1993, "Ridge method for finding saddle points on potential energy surfaces," The Journal of chemical physics, 98, p. 6377.
[88] Henkelman, G., and Jónsson, H., 1999, "A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives," The Journal of chemical physics, 111(15), pp. 7010-7022.
[89] Heyden, A., Bell, A. T., and Keil, F. J., 2005, "Efficient methods for finding transition states in chemical reactions: Comparison of improved

dimer method and partitioned rational function optimization method," The Journal of chemical physics, 123, pp. 224101-224114.

[90] Dewar, M. J., Healy, E. F., and Stewart, J. J., 1984, "Location of transition states in reaction mechanisms," Journal of the Chemical Society, Faraday Transactions 2: Molecular and Chemical Physics, 80(3), pp. 227-233.

[91] Miron, R. A., and Fichthorn, K. A., 2001, "The Step and Slide method for finding saddle points on multidimensional potential surfaces," J. Chem. Phys., 115, p. 8742.

[92] Passerone, D., Ceccarelli, M., and Parrinello, M., 2003, "A concerted variational strategy for investigating rare events," J. Chem. Phys., 118, p. 2025.

[93] Saad, Y., 2003, Iterative methods for sparse linear systems, Siam.

[94] Sinclair, J., and Fletcher, R., 1974, "A new method of saddle-point location for the calculation of defect migration energies," Journal of Physics C: Solid State Physics, 7(5), p. 864.

[95] Bell, S., Crighton, J. S., and Fletcher, R., 1981, "A new efficient method for locating saddle points," Chemical Physics Letters, 82(1), pp. 122-126.

[96] Schlegel, H. B., 1982, "Optimization of equilibrium geometries and transition structures," Journal of Computational Chemistry, 3(2), pp. 214-218.

[97] Fischer, S., and Karplus, M., 1992, "Conjugate peak refinement: an algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom," Chemical physics letters, 194(3), pp. 252-261.

[98] Chen, L., Ying, S., and Ala-Nissila, T., 2002, "Finding transition paths and rate coefficients through accelerated Langevin dynamics," Phys. Rev. E, 65(4), p. 042101.

[99] Dey†, B. K., and Ayers, P. W., 2006, "A Hamilton–Jacobi type equation for computing minimum potential energy paths," Mol. Phys., 104(4), pp. 541-558.

[100] Carr, J. M., Trygubenko, S. A., and Wales, D. J., 2005, "Finding pathways between distant local minima," J. Chem. Phys., 122(23), pp. 234903-234903-234907.

[101] Govind, N., Petersen, M., Fitzgerald, G., King-Smith, D., and Andzelm, J., 2003, "A generalized synchronous transit method for transition state location," Computational materials science, 28(2), pp. 250-258.

[102] Ruedenberg, K., and Sun, J. Q., 1994, "A simple prediction of approximate transition states on potential energy surfaces," J. Chem. Phys., 101, p. 2168.

[103] Ulitsky, A., and Shalloway, D., 1997, "Finding transition states using contangency curves," The Journal of chemical physics, 106(24), pp. 10099-10104.

[104] Hestenes, M. R., and Stiefel, E., 1952, "Methods of conjugate gradients for solving linear systems," NBS.

[105] Shewchuk, J. R., 1994, "An introduction to the conjugate gradient method without the agonizing pain," Carnegie Mellon University, Pittsburgh, PA.

[106] Golub, G. H., and O'Leary, D. P., 1989, "Some history of the conjugate gradient and Lanczos algorithms: 1948-1976," SIAM review, 31(1), pp. 50-102.

[107] Fletcher, R., and Reeves, C., 1964, "Function minimization by conjugate gradients," The computer journal, 7(2), pp. 149-154.

[108] POLA, E., and Ribiere, G., 1969, "Note sur la convergence de methodes de directions conjugées," Rev Française Informat Recherche Operationelle, 3e Année, 16, pp. 35-43.

[109] Zoutendijk, G., 1970, "Nonlinear programming, computational methods," Integer and nonlinear programming, 143(1), pp. 37-86.

[110] Al-Baali, M., 1985, "Descent property and global convergence of the Fletcher—Reeves method with inexact line search," IMA Journal of Numerical Analysis, 5(1), pp. 121-124.

[111] Powell, M., 1984, "Nonconvex minimization calculations and the conjugate gradient method," Numerical Analysis, pp. 122-141.

[112] Hu, Y., and Storey, C., 1991, "Global convergence result for conjugate gradient methods," Journal of Optimization Theory and Applications, 71(2), pp. 399-405.

[113] Gilbert, J. C., and Nocedal, J., 1992, "Global convergence properties of conjugate gradient methods for optimization," SIAM Journal on Optimization, 2(1), pp. 21-42.

[114] Guanghui, L., Jiye, H., and Hongxia, Y., 1995, "Global convergence of the Fletcher-Reeves algorithm with inexact linesearch," Applied Mathematics-A Journal of Chinese Universities, 10(1), pp. 75-82.

[115] Dai, Y., Han, J., Liu, G., Sun, D., Yin, H., and Yuan, Y. X., 2000, "Convergence properties of nonlinear conjugate gradient methods," SIAM Journal on Optimization, 10(2), pp. 345-358.

[116] Mortenson, M. E., 1985, Geometrie modeling, John Wiley, New York.
[117] Watkins, M. A., and Worsey, A. J., 1988, "Degree reduction of Bézier curves," Comput. Aided. Des., 20(7), pp. 398-405.
[118] Eck, M., 1993, "Degree reduction of Bézier curves," Computer Aided Geometric Design, 10(3), pp. 237-251.
[119] Forrest, A. R., 1972, "Interactive interpolation and approximation by Bézier polynomials," Comput. J, 15(1), pp. 71-79.
[120] Eck, M., 1995, "Least squares degree reduction of Bézier curves," Comput. Aided. Des., 27(11), pp. 845-851.
[121] Bogacki, P., Weinstein, S. E., and Xu, Y., 1995, "Degree reduction of Bézier curves by uniform approximation with endpoint interpolation," Comput. Aided. Des., 27(9), pp. 651-661.
[122] Brunnett, G., Schreiber, T., and Braun, J., 1996, "The geometry of optimal degree reduction of Bézier curves," Computer Aided Geometric Design, 13(8), pp. 773-788.
[123] Kim, H., and Moon, S., 1997, "Degree reduction of Bézier curves by< i> L</i>< sup> 1</sup>-Approximation with endpoint interpolation," Computers & Mathematics with Applications, 33(5), pp. 67-77.
[124] Kim, H., and Ahn, Y., 2000, "Good degree reduction of Bézier curves using Jacobi polynomials," Computers & Mathematics with Applications, 40(10), pp. 1205-1215.
[125] Ahn, Y. J., 2003, "Using Jacobi polynomials for degree reduction of Bézier curves with< i> C</i>< sup> k</sup>-constraints," Computer Aided Geometric Design, 20(7), pp. 423-434.
[126] Chen, G. D., and Wang, G. J., 2002, "Optimal multi-degree reduction of Bézier curves with constraints of endpoints continuity," Computer Aided Geometric Design, 19(6), pp. 365-377.
[127] Sunwoo, H., 2005, "Matrix representation for multi-degree reduction of Bézier curves," Computer Aided Geometric Design, 22(3), pp. 261-273.
[128] Lu, L., and Wang, G., 2008, "Application of Chebyshev II–Bernstein basis transformations to degree reduction of Bézier curves," Journal of Computational and Applied Mathematics, 221(1), pp. 52-65.
[129] Lu, L., and Wang, G., 2006, "Optimal multi-degree reduction of Bézier curves with G2-continuity," Computer Aided Geometric Design, 23(9), pp. 673-683.

[130] Rababah, A., Lee, B. G., and Yoo, J., 2006, "A simple matrix form for degree reduction of Bézier curves using Chebyshev–Bernstein basis transformations," Applied mathematics and computation, 181(1), pp. 310-318.

[131] Woźny, P., and Lewanowicz, S., 2009, "Multi-degree reduction of Bézier curves with constraints, using dual Bernstein basis polynomials," Computer Aided Geometric Design, 26(5), pp. 566-579.

[132] Chen, X. D., Ma, W., and Paul, J. C., 2011, "Multi-degree reduction of Bézier curves using reparameterization," Comput. Aided. Des., 43(2), pp. 161-169.

[133] Truhlar, D. G., Steckler, R., and Gordon, M. S., 1987, "Potential energy surfaces for polyatomic reaction dynamics," Chemical Reviews, 87(1), pp. 217-236.

[134] Varandas, A., 1988, "Intermolecular and intramolecular potentials," Adv. Chem. Phys, 74, pp. 255-338.

[135] Schatz, G. C., 1989, "The analytical representation of electronic potential-energy surfaces," Reviews of Modern Physics, 61(3), p. 669.

[136] Maisuradze, G. G., Thompson, D. L., Wagner, A. F., and Minkoff, M., 2003, "Interpolating moving least-squares methods for fitting potential energy surfaces: Detailed analysis of one-dimensional applications," J. Chem. Phys., 119, p. 10002.

[137] Raff, L., Malshe, M., Hagan, M., Doughan, D., Rockley, M., and Komanduri, R., 2005, "Ab initio potential-energy surfaces for complex, multichannel systems using modified novelty sampling and feedforward neural networks," J. Chem. Phys., 122, p. 084104.

[138] Schatz, G. C., 2000, "Fitting potential energy surfaces," Reaction and Molecular Dynamics, Springer, pp. 15-32.

[139] McLaughlin, D. R., and Thompson, D. L., 1973, "Ab initio dynamics: HeH+ H→ He+ H (C) classical trajectories using a quantum mechanical potential ‐ energy surface," J. Chem. Phys., 59, p. 4393.

[140] Sathyamurthy, N., and Raff, L., 1975, "Quasiclassical trajectory studies using 3D spline interpolation of ab initio surfaces," J. Chem. Phys., 63, p. 464.

[141] Sathyamurthy, N., Rangarajan, R., and Raff, L., 1976, "Reactive scattering calculations on a splinefitted ab initio surface: The He+ H (v= 0, 1, 2)→ HeH+ H reaction," J. Chem. Phys., 64, p. 4606.

[142] Chapman, S., Dupuis, M., and Green, S., 1983, "Theoretical three-dimensional potential-energy surface for the reaction of Be with HF," Chem. Phys., 78(1), pp. 93-105.

[143] Bowman, J. M., Bittman, J. S., and Harding, L. B., 1986, "Ab initio calculations of electronic and vibrational energies of HCO and HOC," J. Chem. Phys., 85, p. 911.

[144] Koizumi, H., Schatz, G. C., and Walch, S. P., 1991, "A coupled channel study of HN unimolecular decay based on a global ab initio potential surface," J. Chem. Phys., 95, p. 4130.

[145] Wall, F. T., and Porter, R. N., 1962, "General Potential ‐ Energy Function for Exchange Reactions," J. Chem. Phys., 36, p. 3256.

[146] Bowman, J. M., and Kuppermann, A., 1975, "A semi-numerical approach to the construction and fitting of triatomic potential energy surfaces," Chem. Phys. Lett., 34(3), pp. 523-527.

[147] Gray, S. K., and Wright, J. S., 1977, "Classical trajectories for the H+ H reaction on a spline ‐ generated potential energy surface," J. Chem. Phys., 66, p. 2867.

[148] Murrell, J. N., and Murrell, J. N., 1984, Molecular potential energy functions, J. Wiley Chichester, UK.

[149] Ho, T. S., and Rabitz, H., 1996, "A general method for constructing multidimensional molecular potential energy surfaces from ab initio calculations," J. Chem. Phys., 104, p. 2584.

[150] Hollebeek, T., Ho, T.-S., and Rabitz, H., 1999, "Constructing multidimensional molecular potential energy surfaces from ab initio data," Annual review of physical chemistry, 50(1), pp. 537-570.

[151] Ho, T.-S., and Rabitz, H., 2003, "Reproducing kernel Hilbert space interpolation methods as a paradigm of high dimensional model representations: Application to multidimensional potential energy surface construction," J. Chem. Phys., 119, p. 6433.

[152] Blank, T. B., Brown, S. D., Calhoun, A. W., and Doren, D. J., 1995, "Neural network models of potential energy surfaces," J. Chem. Phys., 103, p. 4129.

[153] Gassner, H., Probst, M., Lauenstein, A., and Hermansson, K., 1998, "Representation of intermolecular potential functions by neural networks," The Journal of Physical Chemistry A, 102(24), pp. 4596-4605.

[154] Lorenz, S., Groß, A., and Scheffler, M., 2004, "Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks," Chem. Phys. Lett., 395(4), pp. 210-215.

[155] Ischtwan, J., and Collins, M. A., 1994, "Molecular potential energy surfaces by interpolation," J. Chem. Phys., 100, p. 8080.

[156] Jordan, M. J., Thompson, K. C., and Collins, M. A., 1995, "The utility of higher order derivatives in constructing molecular potential energy surfaces by interpolation," J. Chem. Phys., 103, p. 9669.

[157] Jordan, M. J., Thompson, K. C., and Collins, M. A., 1995, "Convergence of molecular potential energy surfaces by interpolation: Application to the OH+ H→ HO+ H reaction," J. Chem. Phys., 102, p. 5647.

[158] Jordan, M. J., and Collins, M. A., 1996, "An interpolated unrestricted Hartree – Fock potential energy surface for the OH+ H→ HO+ H reaction," J. Chem. Phys., 104, p. 4600.

[159] Thompson, K., and Collins, M., 1997, "Molecular potential-energy surfaces by interpolation: Further refinements," J. Chem. Soc., Faraday Trans., 93(5), pp. 871-878.

[160] Bettens, R. P., and Collins, M. A., 1998, "Potential energy surfaces and dynamics for the reactions between C (P) and H (A)," J. Chem. Phys., 108, p. 2424.

[161] Bettens, R. P., and Collins, M. A., 1998, "Interpolated potential energy surface and dynamics for the reactions between N (S) and H (A)," J. Chem. Phys., 109, p. 9728.

[162] Thompson, K. C., Jordan, M. J., and Collins, M. A., 1998, "Polyatomic molecular potential energy surfaces by interpolation in local internal coordinates," J. Chem. Phys., 108, p. 8302.

[163] Thompson, K. C., Jordan, M. J., and Collins, M. A., 1998, "Molecular potential energy surfaces by interpolation in Cartesian coordinates," J. Chem. Phys., 108, p. 564.

[164] Bettens, R. P., and Collins, M. A., 1999, "Learning to interpolate molecular potential energy surfaces with confidence: A Bayesian approach," J. Chem. Phys., 111, p. 816.

[165] Ishida, T., and Schatz, G. C., 1999, "A local interpolation scheme using no derivatives in quantum-chemical calculations," Chem. Phys. Lett., 314(3), pp. 369-375.

[166] Dawes, R., Thompson, D. L., Guo, Y., Wagner, A. F., and Minkoff, M., 2007, "Interpolating moving least-squares methods for fitting potential energy surfaces: Computing high-density potential energy surface data from low-density ab initio data points," J. Chem. Phys., 126, p. 184108.

[167] Guo, Y., Tokmakov, I., Thompson, D. L., Wagner, A. F., and Minkoff, M., 2007, "Interpolating moving least-squares methods for fitting potential energy surfaces: Improving efficiency via local approximants," J. Chem. Phys., 127, p. 214106.

[168] Lancaster, P., and Salkauskas, K., 1986, "Curve and surface fitting. An introduction," London: Academic Press, 1986, 1.

[169] Kbiob, D., 1951, "A statistical approach to some basic mine valuation problems on the Witwatersrand," Journal of Chemical, Metallurgical, and Mining Society of South Africa.

[170] Matheron, G., 1963, "Principles of geostatistics," Economic geology, 58(8), pp. 1246-1266.

[171] Cressie, N., 1993, "Statistics for Spatial Data: Wiley Series in Probability and Statistics."

[172] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, "Design and analysis of computer experiments," Statistical science, pp. 409-423.

[173] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D., 1991, "Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments," Journal of the American Statistical Association, 86(416), pp. 953-963.

[174] Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D., 1992, "Screening, predicting, and computer experiments," Technometrics, 34(1), pp. 15-25.

[175] Morris, M. D., Mitchell, T. J., and Ylvisaker, D., 1993, "Bayesian design and analysis of computer experiments: use of derivatives in surface prediction," Technometrics, 35(3), pp. 243-255.

[176] Jones, D. R., Schonlau, M., and Welch, W. J., 1998, "Efficient global optimization of expensive black-box functions," Journal of Global optimization, 13(4), pp. 455-492.

[177] Rijpkema, J., Etman, L., and Schoofs, A., 2001, "Use of design sensitivity information in response surface and Kriging metamodels," Optimization and Engineering, 2(4), pp. 469-484.

[178] Jakumeit, J., Herdy, M., and Nitsche, M., 2005, "Parameter optimization of the sheet metal forming process using an iterative parallel Kriging algorithm," Structural and Multidisciplinary Optimization, 29(6), pp. 498-507.

[179] Choi, H., McDowell, D. L., Allen, J. K., Rosen, D., and Mistree, F., 2008, "An inductive design exploration method for robust multiscale materials design," Journal of Mechanical Design, 130(3), p. 031402.

[180] Lee, T. H., and Jung, J. J., 2008, "A sampling technique enhancing accuracy and efficiency of metamodel-based RBDO: Constraint boundary sampling," Computers & Structures, 86(13), pp. 1463-1476.

[181] Li, M., Li, G., and Azarm, S., 2008, "A kriging metamodel assisted multi-objective genetic algorithm for design optimization," Journal of Mechanical Design, 130(3), p. 031401.

[182] Zhao, L., Choi, K. K., Lee, I., and Du, L., "Response surface method using sequential sampling for reliability-based design optimization," Proc. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, pp. 1171-1181.

[183] Lee, I., Choi, K., and Zhao, L., 2011, "Sampling-based RBDO using the stochastic sensitivity analysis and Dynamic Kriging method," Structural and Multidisciplinary Optimization, 44(3), pp. 299-317.

[184] Zhao, L., Choi, K., Lee, I., and Gorsich, D., 2013, "Conservative Surrogate Model Using Weighted Kriging Variance for Sampling-Based RBDO," Journal of Mechanical Design, 135(9), p. 091003.

[185] Joseph, V. R., Hung, Y., and Sudjianto, A., 2008, "Blind kriging: A new method for developing metamodels," Journal of mechanical design, 130(3), p. 031102.

[186] Zhao, L., Choi, K., and Lee, I., 2011, "Metamodeling method using dynamic kriging for design optimization," AIAA journal, 49(9), pp. 2034-2046.

[187] Van Beers, W. C., and Kleijnen, J. P., 2003, "Kriging for interpolation in random simulation," Journal of the Operational Research Society, 54(3), pp. 255-262.

[188] Ankenman, B., Nelson, B. L., and Staum, J., 2010, "Stochastic kriging for simulation metamodeling," Operations research, 58(2), pp. 371-382.

[189] Ba, S., and Joseph, V. R., 2012, "Composite Gaussian process models for emulating expensive functions," The Annals of Applied Statistics, 6(4), pp. 1838-1860.

[190] Koehler, J., and Owen, A., 1996, "Computer experiments," Handbook of statistics, 13(13), pp. 261-308.

[191] Martin, J. D., and Simpson, T. W., 2004, "On using Kriging models as probabilistic models in design," SAE Technical Paper.

[192] Chen, V. C., Tsui, K.-L., Barton, R. R., and Meckesheimer, M., 2006, "A review on design, modeling and applications of computer experiments," IIE transactions, 38(4), pp. 273-291.

[193] Wang, G. G., and Shan, S., 2007, "Review of metamodeling techniques in support of engineering design optimization," Journal of Mechanical Design, 129(4), pp. 370-380.

[194] Li, J., and Australia, G., 2008, A review of spatial interpolation methods for environmental scientists, Geoscience Australia Canberra.
[195] Kleijnen, J. P., 2009, "Kriging metamodeling in simulation: A review," European Journal of Operational Research, 192(3), pp. 707-716.
[196] Beale, E. M. L., 1972, Numerical methods for non-linear optimization Academic Press, London.
[197] Weinan, E., Ren, W., and Vanden-Eijnden, E., 2007, "Simplified and improved string method for computing the minimum energy paths in barrier-crossing events," Journal of Chemical Physics, 126(16), pp. 164103(164101) -164103(164108).
[198] Farin, G. E., 1993, Curves and Surfaces for Computer-Aided Geometric Design, Academic Press, Boston.
[199] Polanyi, J., and Wong, W., 1969, "Location of energy barriers. I. Effect on the dynamics of reactions A+ BC," J. Chem. Phys., 51, p. 1439.
[200] Busch, G., Schlapbach, L., Stucki, F., Fischer, P., and Andresen, A., 1979, "Hydrogen storage in FeTi: surface segregation and its catalytic effect on hydrogenation and structural studies by means of neutron diffraction," International Journal of Hydrogen Energy, 4(1), pp. 29-39.
[201] Jang, T., Han, J., and Jai-Young, L., 1986, "Effect of substitution of titanium by zirconium in TiFe on hydrogenation properties," Journal of the Less Common Metals, 119(2), pp. 237-246.
[202] Bratanich, T., Solonin, S., and Skorokhod, V., 1996, "Hydrogen sorption peculiarities of mechanically activated intermetallic TiFe and TiFe-MmNi 5 (LaNi 5) mixtures," International journal of hydrogen energy, 21(11), pp. 1049-1051.
[203] Kresse, G., and Furthmüller, J., 1996, "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set," Computational Materials Science, 6(1), pp. 15-50.
[204] Kresse, G., and Furthmüller, J., 1996, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," Physical Review B, 54(16), p. 11169.
[205] Blöchl, P. E., 1994, "Projector augmented-wave method," Physical Review B, 50(24), p. 17953.
[206] Kresse, G., and Joubert, D., 1999, "From ultrasoft pseudopotentials to the projector augmented-wave method," Physical Review B, 59(3), p. 1758.
[207] Perdew, J. P., and Zunger, A., 1981, "Self-interaction correction to density-functional approximations for many-electron systems," Physical Review B, 23(10), p. 5048.

[208] Lebsanft, E., Richter, D., and Topler, J., 1979, "Investigation of the hydrogen diffusion in FeTiHx by means of quasielastic neutron scattering," Journal of Physics F: Metal Physics, 9(6), p. 1057.

[209] Izanlou, A., and Aydinol, M., 2010, "An ab initio study of dissociative adsorption of H 2 on FeTi surfaces," international journal of hydrogen energy, 35(4), pp. 1681-1692.

[210] Nørskov, J., 1982, "Covalent effects in the effective-medium theory of chemical binding: Hydrogen heats of solution in the 3 d metals," Physical Review B, 26(6), p. 2875.

[211] Juan, A., and Hoffmann, R., 1999, "Hydrogen on the Fe (110) surface and near bulk bcc Fe vacancies: a comparative bonding study," Surface science, 421(1), pp. 1-16.

[212] Jiang, D., and Carter, E. A., 2004, "Diffusion of interstitial hydrogen into and through bcc Fe from first principles," Physical Review B, 70(6), p. 064102.

[213] Gong, X.-g., Zeng, Z., and Zheng, Q.-Q., 1989, "Electronic structure of light impurities in alpha-Fe and V," Journal of Physics: Condensed Matter, 1(41), p. 7577.

[214] Puska, M., and Nieminen, R., 1984, "Theory of hydrogen and helium impurities in metals," Physical Review B, 29(10), p. 5382.

[215] Hayashi, Y., and Shu, W., 2000, "Iron (ruthenium and osmium)-hydrogen systems," Solid State Phenomena, 73-75, pp. 65-114.

[216] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., 2002, "DACE-A Matlab Kriging toolbox, version 2.0."

[217] Cressie, N., and Johannesson, G., 2008, "Fixed rank kriging for very large spatial data sets," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(1), pp. 209-226.

[218] Giannozzi, P., Baroni, S., Bonini, N., Calandra, M., Car, R., Cavazzoni, C., Ceresoli, D., Chiarotti, G. L., Cococcioni, M., and Dabo, I., 2009, "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials," Journal of Physics: Condensed Matter, 21(39), p. 395502.