Development of New Global Optimization Algorithms Using Stochastic Level Set Method

with Application in:

Topology Optimization, Path Planning and Image Processing

by

Seyed Alireza Kasaiezadeh Mahabadi

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

A unique mathematical tool is developed to deal with global optimization of a set of engineering problems. These include image processing, mechanical topology optimization, and optimal path planning in a variational framework, as well as some benchmark problems in parameter optimization.

The optimization tool in these applications is based on the level set theory by which an evolving contour converges toward the optimum solution. Depending upon the application, the objective function is defined, and then the level set theory is used for optimization. Level set theory, as a member of active contour methods, is an extension of the steepest descent method in conventional parameter optimization to the variational framework. It intrinsically suffers from trapping in local solutions, a common drawback of gradient based optimization methods. In this thesis, methods are developed to deal with this drawbacks of the level set approach.

By investigating the current global optimization methods, one can conclude that these methods usually cannot be extended to the variational framework; or if they can, the computational costs become drastically expensive. To cope with this complexity, a global optimization algorithm is first developed in parameter space and compared with the existing methods. This method is called "Spiral Bacterial Foraging Optimization" (SBFO) method because it is inspired by the aggregation process of a particular bacterium called, Dictyostelium Discoideum. Regardless of the real phenomenon behind the SBFO, it leads to new ideas in developing global optimization methods. According to these ideas, an effective global optimization method should have i) a stochastic operator, and/or ii) a multi-agent structure. These two properties are very common in the existing global optimization methods. To improve the computational time and costs, the algorithm may include gradient-based approaches to increase the convergence speed. This property is particularly available in SBFO and it is the basis on which SBFO can be extended to variational framework.

To mitigate the computational costs of the algorithm, use of the gradient based approaches can be helpful. Therefore, SBFO as a multi-agent stochastic gradient based structure can be extended to multi-agent stochastic level set method. In three steps, the variational set up is formulated: i) A single stochastic level set method, called "Active Contours with Stochastic Fronts" (ACSF), ii) Multi-agent stochastic level set method (MSLSM), and iii) Stochastic level set method without gradient such as E-ARC algorithm.

For image processing applications, the first two steps have been implemented and show significant improvement in the results. As expected, a multi agent structure is more accurate in terms of ability to find the global solution but it is much more computationally expensive. According to the results, if one uses an initial level set with enough holes in its topology, a single stochastic level set method can achieve almost the same level of accuracy as a multi-agent structure can obtain. Therefore, for a topology optimization problem for which a high level of calculations (at each iteration a finite element model should be solved) is required, only ACSF with initial guess with multiple holes is implemented. In some applications, such as optimal path planning, objective functions are usually very complicated; finding a closed-form equation for the objective function and its gradient is therefore impossible or sometimes very computationally expensive. In these situations, the level set theory and its extensions cannot be directly employed. As a result, the Evolving Arc algorithm that is inspired by "Electric Arc" in nature, is proposed. The results show that it can be a good solution for either unconstrained or constrained problems. Finally, a rigorous convergence analysis for SBFO and ACSF is presented that is new amongst global optimization methods in both parameter and variational framework.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Prof. Amir Khajepour for his endless support, guidance and encouragement throughout the research. During last five years, his continued support led me to the right way not only in this research but also in my life.

I would also like to extend my sincere appreciation to Prof. Steven L. Waslander for his invaluable advise on development of SBFO algorithm that made the basis of the algorithms developed in this thesis. I wish to acknowledge the other members of my committee, Prof. Steve Lambert and Gordon J. Savage for their useful comments, suggestions and discussions. Also, I wish to express my thanks for the thoughtful review and many helpful suggestions of Prof. Kamran Dehdinan as the external examiner.

I should also appreciate the friendship and support of my friends. I wish to thank Abolfazl Maneshi, Ehsan Chiniforooshan, Nasser Lashgarian Azad, Bashir S. Sadjad, Yaser Shanjani, Mojtaba Haghighi and Babak Ebrahimi.

I will also like to appreciate my beloved wife, Mona, for her support and understanding. Without her continuous support, encouragement and love, my study would have never been done properly.

Finally, my greatest appreciations and thanks are dedicated to my parents whom I am greatly indebted during my whole life.

Table of Contents

List of Tables

List of Figures

Nomenclature

ϕ	Evolving implicit function
Ω	Feasible region defined by ϕ
$\partial\Omega$	Boundary of the feasible region Ω
$\delta(\phi)$	Dirac delta function
$H(\phi)$	Heaviside function
$\nabla\cdot$	Divergence operator
θ	Perturbation map
Ω_θ	Evolved region under perturbation map
∂_S	Shape derivative
∂_T	Topology derivative
κ	Mean curvature of $\partial\Omega$
n	Normal vector to the $\partial\Omega$
V_n	Normal velocity at each point of $\partial\Omega$
$\lvert D\phi\rvert$	Norm of the gradient vector of ϕ
$B_{\rho,x}$	A small ball centered at point x and radius ρ
T_n	Modified topology derivate
w	Balance factor between topology and shape derivative

$C(\phi)$	Mean compliance of structure
$V(\phi)$	Volume of structure
E_{ijkl}	Young’s modulus elasticity tensor
$L(\phi)$	Lagrangian functional
SE	Strain energy
MSE	Mutual strain energy
MA	Mechanical advantage
GA	Geometric advantage
σ_d	Stress field when only a unit dummy load is applied
k_e	Element stiffness matrix
K	Global stiffness matrix
f_b	Body force
g	External traction force
Γ_D	Dirichlet boundary condition
Γ_N	Neumann boundary condition
” $\circ$ ”	Stratonovich operator
$BUC(\mathbb{R}^N)$	Bounded uniformly continuous subset of $\mathbb{R}^N$
$\mathcal{I}(\mathbb{R}^n)$	Lie algebra of all C^∞-vector fields on $\mathbb{R}^n$

Chapter 1

Introduction

In general, global optimization methods are effective tools for engineering applications. One important point, which is usually ignored, is that local solutions are sometimes nonfeasible. In other words, sometimes local solutions cannot even meet the required properties that an engineering solution needs. As a result, global optimization theory is still an open field amongst science and engineering researchers.

Some major issues associated with the existing global optimization methods are as follows: First, these methods are often very slow; second, there is no rigorous mathematical proof for their convergence and; third, for some applications, such as image segmentation, topology optimization, and path planning, the problems need to be formulated in variational framework and usually the existing methods cannot be applied in this framework.

Conventional optimization methods that usually take the advantages of gradient vector are generally faster than zero-order or gradient free approaches. This is because they follow the gradient direction to reach the solution. In contrast, the gradient free algorithms usually employ random search methods along with a multi-agent structure to explore the entire feasible space; this is the time consuming part of the process. The CPU time issue becomes even worse for variational problems; hence, a global optimization method based on gradient flow is highly desirable for such problems. As a result, multi-agent stochastic level set method, which has the advantages of all types of existing algorithms, seems to be a good way to deal with variational problems.

The basic idea behind the stochastic gradient system, and in particular stochastic level set methods, is to add a stochastic operator to the conventional level set method. There is no general proof to show that the resultant solution is the global solution, but since the local solutions are usually not as stable as the global one, once the local solution is perturbed, it often leaves that location and converges towards a more stable one. Again, it is worth mentioning that the process still might be trapped in a local solution.

To introduce a method capable of finding the global solution of a variational problem, the first step is to develop a new global parameter optimization technique. A method has been developed in this thesis called Spiral Bacterial Foraging Optimization (SBFO). Based on the ideas learned from this method, some algorithms in the variational framework are proposed. SBFO has a multi-agent stochastic gradient-based structure, therefore the corresponding variational approach should have the same structure. This is done in two steps. First, a single-agent structure called "Active Contours with Stochastic Fronts" (ACSF) is proposed and then a multi-agent structure is investigated. Each agent in the multiple structure has the same behavior as ACSF, except that the agents are also forced to converge to the same topology. As expected, the mathematical background of ACSF and "Multi-agent Stochastic Level Set Method" (MSLSM) is based on stochastic level set method, which is an extension of the steepest descent approach in a parameter space.

These two algorithms can be practically applied to engineering applications in which the objective functional and the corresponding derivative (gradient vector) are clearly defined in a closed form equation. Obviously, one can find the gradient vector numerically in a variational problem for which a finite difference or finite element model should be solved at each iteration of the algorithm. Topology optimization and image segmentation are two common examples in which closed-form equations exist. In other applications such as optimal path planning, the functionals often have very complex forms. In path planning problem, a functional may contain energy consumption of a rover, radiation received by solar panels and the performance of the rover; hence, a closed-form equation for the gradient vector is not available. In such cases, Evolving Arc (E-ARC) algorithm has been developed. E-ARC is based on stochastic level set method but since the gradient vector is not available in closed form, a flexible numerical approach is designed in order to replace the gradient vector. To clarify the connection of proposed methods with the existing algorithms in both parameterized and variational framework, a chart is drawn in Figure (1.1).

As can be seen, optimization algorithms can be formulated in two different frameworks: Variational and Parameterized space. In each of these two frameworks, there are many algorithms that are capable to find local or global solutions. The most famous techniques to find global solutions in parameterized space are Genetic (and in general evolutionary) Algorithm, Simulated Annealing and Particle Swarm Optimization methods. Local methods in this framework, such as sequential quadratic programming, conjugate gradient method and gradient descent method, are mostly gradient-based approaches. Although fast, gradient based methods suffer from being trapped in local optima. Spiral Bacterial Foraging Optimization (SBFO) method takes the advantages of both groups of algorithms. This method is one of the contributions of this thesis. It is usually faster than other global optimization methods (because it is a gradient based method) but capable to find global solutions. The dashed-line between gradient descent block and SBFO in Figure (1.1) shows that SBFO is based on steepest descent method but belongs to global category.

The extension of the gradient-based methods into variational framework has the same problem. As stated, level set method, that is a an extension of steepest descent method, is very useful to solve variational formulation of engineering applications such as image processing, topology optimization and optimal path planning. To take the advantages of level set method, in this thesis global version of level set method is introduced. Active contours with stochastic fronts (ACSF), multi-agent stochastic level set method (MSLSM) and E-ARC are the main contributions of this thesis that are based on stochastic level set method. The same as SBFO, the dashed-line in Figure (1.1) between local and global algorithms shows that this class of algorithms is based on level set theory but all of its members belong to global techniques. The gray blocks in Figure (1.1) are the developed algorithms in this thesis.

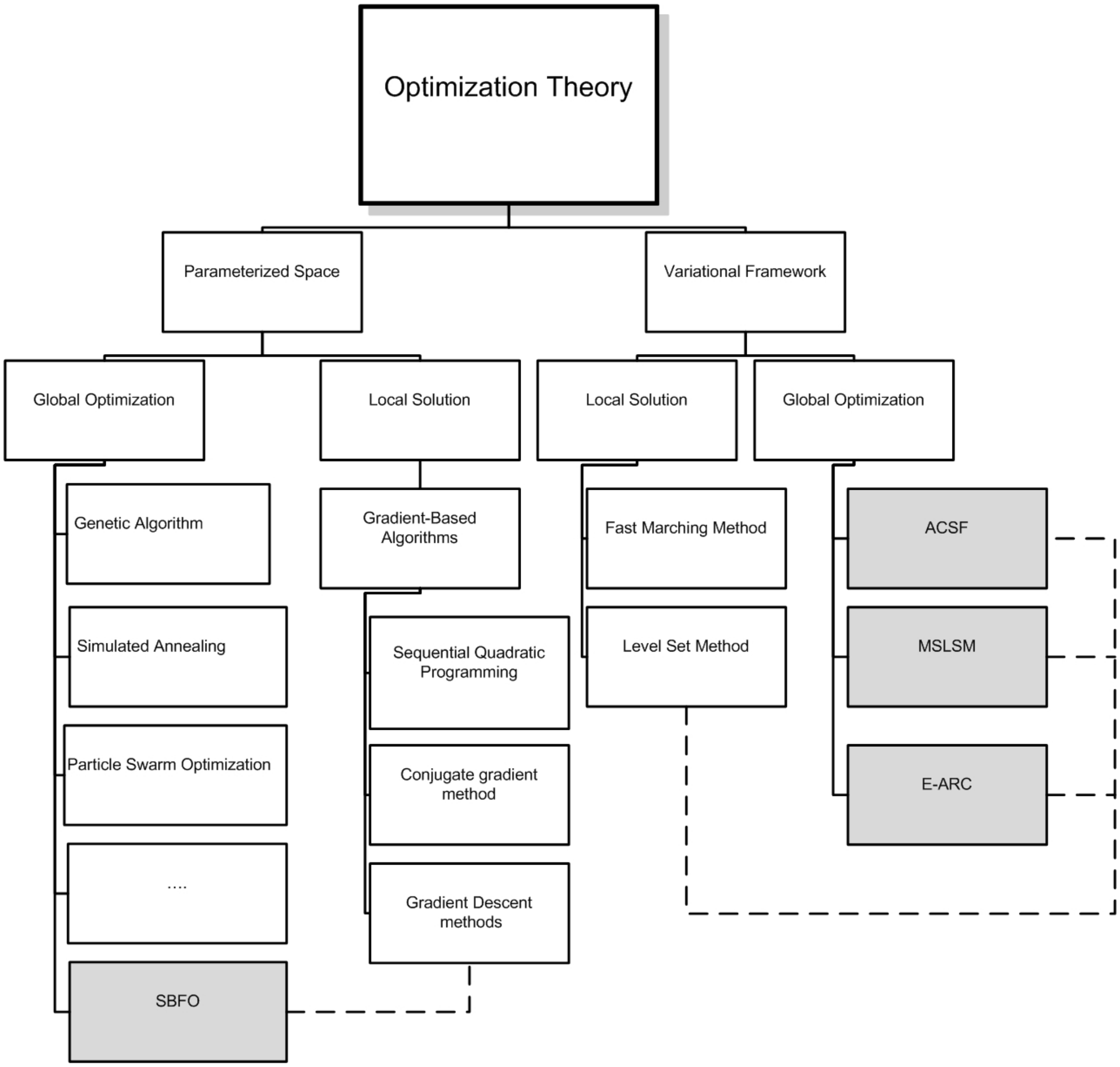


Figure 1.1: A block diagram to show the contributions of this thesis

To have a preliminary background on global optimization methods, level set theory and its applications to image processing, topology optimization, path planning, and stochastic partial differential equations, are reviewed in the following sections.

1.1 Global Parameter Optimization Methods

Inspired by Darwinian evolution and natural selection, evolutionary algorithms (EA) have come to be used extensively in artificial intelligence and optimization problems since the mid-1970s [2]- [4]. Two decades later, particle swarm (PSO) [5] and Ant Colony (ACO) [6] optimization theories were introduced to solve many engineering problems. The main shared property between these algorithms is their multi-individual structure. Being effective at finding the global optima, many extensions have been proposed for these algorithms [7] during recent years.

Two major advantages of PSO and ACO methods are speed of convergence and ease of use, as they have very few parameters to adjust in comparison with EA [8]. However, premature convergence affects the convergence of the PSO [9] by reducing the diversity of particle search locations. Several approaches have been proposed to cope with this problem. A two-stage transformation has been proposed by Parsopulos et al. [10] for stretching the objective functions to discard the local optima problem. In another attempt, a predator-prey strategy is employed to preserve the diversity and prevent premature convergence [11].

In yet another example of a biologically-inspired optimization method, Bremermann et al. [12]- [13] and Anderson [14] established a resemblance for the behavior of bacteria reacting to chemo-attractants in concentration gradients. This idea was advanced by Müller et al. [15], in which several features were added in order to attain a more capable technique with enhanced convergence properties for both local and global optimization problems. Passino et al [16]- [17] proposed the Bacterial Foraging Optimization (BFO) algorithm by analogy to the foraging theory of natural creatures that explains how the "energy intake per unit time spent for foraging" is optimized. The investigated procedures in reference [16]- [17] have a variety of tunable parameters, therefore it is very hard to determine the proper values of these paramters.

A mathematical model has been derived for the chemotaxis phenomenon in [18], and the stability and convergence have been investigated using the Lyapunov stability theorem. This analysis suggests a suitable bound for tunable parameters to have a proper convergence.

Starting with a population of size N_p, D-dimensional parameter vectors, Differential

Evolution (DE) employs mutation, crossover and selection in order to find the global solution of a general nonlinear optimization problem. Depending upon the way that DE creates its donor vectors, various DE schemes are distinguished from each other. Some major schemes are: "DE/rand/1/bin", "DE/best/1", "DE/target-to-best/1", "DE/best/2" and "DE/rand/2". All of these schemes are implemented and publicly available at [19].

Investigation by Biswas et al. [20] resulted in a hybrid method called Chemotaxis Differential Evolution (CDE) in which the BFO algorithm is coupled with DE algorithm [19]. In this method, both issues with slow and premature convergence are overcome.

Working along the same lines, the authors in [21] combine PSO and BFO in order to optimize multi-modal and high dimensional functions. Abraham et al. [22] focus on the improvement of the major operator of the BFO algorithm, the "reproduction phenomenon" of the virtual bacteria. The authors consider the effects of reproduction on bacterial dynamics. Their results show that this phenomenon leads to fast convergence of the bacterial population near the optimal point.

In another extension to BFO, a cooperative approach is proposed by Chen et al. in [23]. It has been claimed that the accuracy, robustness and speed of convergence have been improved significantly by adding the "serial heterogeneous cooperation". The corresponding results show a remarkable improvement in efficiency of the new algorithm, called "CBFO", in comparison with the classic BFO technique. BF-PSO is another improvement for BFO by [24] to combine both BF and PSO algorithms. In this new algorithm PSO is used in order to exchange the social information. Besides, elimination and dispersal are performed by BFO. Shen et al. has investigated BF-PSO presented in [25], in which 23 numerical benchmark problems have been considered and the performance of the new algorithm in a global search is demonstrated.

1.2 Level Set Theory and Its Applications

Implicit description of curves and surfaces is the main concept of the level set method. This capability allows the level set method to be used as a multi-dimensional tool to represent any complex topology for structures. Changes in topology is started from a zero level set or isophote with an initial guess which continuously evolves through a Hamilton-Jacobi PDE formulation based on a certain cost function. Therefore, the complex topological boundaries cannot only be represented by this method, but also the optimal topology can be achieved using this method, [52]- [54].

The level set method is one of the most important algorithms in deformable models and has been widely used in many applications. In comparison with other active contour methods, it has several advantages. These include: 1) there is no need for parameterization of the contour; 2) flexibility in topology definition; 3) numerical stability, and; 4) easiness of extension to higher dimensions [51], [55] and [56].

In this research three main applications of LSM are investigated and improvements of the algorithms are assessed in the same applications. These three applications are: 1) image processing 2) topology optimization 3) optimal path planning. Research literatures associated with these three areas are as follows:

1.2.1 Level Set Theory and Image Processing

Finding a precise, fast, automated and robust approach to segment each image into meaningful sections is crucial in image processing [145]. There are many classical [47] and advanced methods [48] in image segmentation currently available. These include: thresholding, clustering, histogram-based, region-growing, graph partitioning, and optimization model based. However, amongst these, only a few can be applied in practical and non-ideal images. Several review papers on image segmentations [48], [49] and [50] verify that there is no general computational method to cope with all practical images. Therefore, for each class of images, a method should be selected and tuned in order to achieve satisfactory results. Deformable models or active contours are relatively new approaches in image segmentation. As their names imply, they provide an explicit representation of the boundary and the shape of the objects. Compared to the above-mentioned methods, they have several desirable properties, such as inherent connectivity and smoothness, and are capable of adding knowledge domain about the object of interest [51].

The level set method (LSM) [52], [53] and [54] is one of the most important algorithms in deformable models and has been widely employed in image processing and segmentation. Essentially, this method is an extension of steepest descent to the variational framework, and therefore the problem of getting trapped in local optima is inherited to this algorithm. As a result, the final solution is very sensitive to the initial guess [57]. This problem has been relentlessly investigated in recent researches. The level set-based approaches can be grouped into two categories. The methods in the first category try to reformulate the objective functions used in image segmentation in order to improve the performance of the algorithm. On the other hand, the methods in the second division focus on the optimization procedure and the evolution process in the level set method. Kichenassamy et al. [58] and Caselles et al. [59] proposed "Geodesic Active Contours" as an energy-based algorithm for image segmentation, intended to improve the capability of non-convex feature detection. In addition, by this extension, the researchers improved the speed of convergence as well. An interesting review based on "Geodesic Active Contours" can be found in [60], in which the performance of this method on the convexification of the cost functions in image segmentation is evaluated. Finding a general convex function, appropriate for segmentation, is obviously impossible; hence, instead of evolving based on local edge information [61], the use of region-based cost functions [63] (with less local optima) would be far less sensitive to the initial guess. Region-based active contours are originated by Mumford-Shah [63]. In this research, they proposed a new segmentation framework in which the minimization of the functional in Equation(1.1) was employed in order to segment the image, I, by using a set of contours, $\partial\Omega$ [51], [63].

$$E(\partial\Omega, f) = \alpha \int\limits_{\Omega} (f - I)^2 dx + \beta \int\limits_{\Omega\backslash\partial\Omega} |\nabla f| \, dx + H^{n-1}(\partial\Omega) \tag{1.1}$$

where $H^{n-1}(\partial\Omega)$ is the $(N-1)$ dimensional Hausdorff measure and $\alpha, \beta \in \mathbb{R}^+$ are positive real coefficients, f is a smooth approximation of image I and Ω is the region surrounded by $\partial\Omega$

The main drawback of this method is its high sensitivity to in-homogeneities caused by noise and artifacts in practical images. The moving organs in medical images in particular exaggerate these in-homogeneities.

Using this framework, Chan and Vese create a new vision as "Active Contour without Edge" in a series of oft-cited papers [64]- [68]. For them, object detection is performed not necessarily by the gradient at its edges. Rather, the basic idea is based on the minimal partition problem in which the stopping criteria does not depend on the gradient of the image. While the problem is not convex yet has less local optima, the level set method, which is a gradient-based algorithm, usually finds the global (or the best possible solution) topology that defines an accurate segmentation. Next is a brief review of the second approach to improving the performance of the level set method.

There are some other models that sometimes demonstrate better responses than Chan-Vese model. An outstanding example of this works is the research by X. Bresson [69] in which the proposed model has a different local optimal than ACWE, and usually performs better than ACWE in terms of accuracy of segmentation particularly when the images suffers from low contrast features. The basic idea behind it is to add edge-based indicator into the region-based model. Obviously, as this method is still founded on gradient flow the algorithm may stuck in local solutions but this local solutions are usually more accurate than the solutions of ACWE in terms of segmentation of low contrast edges.

1.2.2 Level Set Theory and Topology Optimization

Topology Optimization and Classic Level Set Theory

In recent years, topology optimization of continuum structures has become an intriguing task in engineering. There are a few popular homogenization methods for the topological design of structures [91], [92] and [93]. However, these techniques work well in linear or slightly nonlinear cases, while they fail in highly nonlinear problems [94]. Moreover, some of these approaches are restricted to certain objective functions such as eigenvalues or compliance [93]. These methods are based on material models with micro scale voids, which optimize the porosity of a structure based on a defined cost function. Generally speaking, a homogenization method replaces the problem of topology optimization with a sizing problem which is able to create internal holes [92]- [93].

In the past decade, the level set method has been developed for many applications in engineering problems such as motion by mean curvature [95], fluid mechanics, image processing, etc., in which evolving fronts play a main role [95]- [53]. The ease in describing closed boundaries and less computational efforts are some of the advantages of this method. Hamilton-Jacobi's partial differential equations (PDE) are used to represent the complex material boundaries in the structure [96]- [53]. Additionally these methods can be well-positioned and employed where nonlinearities such as material nonlinear behavior and large deformations are present in the system [94].

Osher and Santosa in [96] focus on a two-phase optimization of a membrane and combine the level set method with a shape sensitivity analysis. In another study [53], the authors use the interface method to capture the shape of the structure as a free boundary on a fix mesh. Allair et al. [94] extended the previous work by implementing a systematic level set method in which the evolving velocity is derived from a shape sensitivity analysis. The authors address more complex objective functions like eigen-frequencies and multiple loads in their work [97]. Instead of considering two-phase optimization, they apply shape optimization and substitute the immersed interface method by the simpler ersatz material [98] approach.

In [94], the authors study the case of a nonlinear elasticity model. To do so, the shape derivative was computed by an ad-joint problem and then used as the normal velocity of front in the optimization process. In other words, front propagation is carried out by solving a Hamilton-Jacobi equation for a level set function [94], [99]. Wang, et al. in [100] propose a level set method for designing monolithic compliant mechanism consisting of several materials. The main contribution of this work is to accurately describe the material regions and their sharp interfaces at the structure boundaries. In addition, a scheme to investigate the connectivity of the structure is presented by [101]- [102].

Level Set Theory and Global Optimization

Finding the global optimum solution is one of the most challenging aspects of topology optimization. Since the level set theory is based on the gradient methods, trapping scenarios in local optimum points is highly expected. A very common symptom of the local solution is the dependency of the final solution on initial guesses. Different research studies are designed in order to address this problem. The existing approaches to achieve the best possible solution of topology optimization problem can be divided into two main groups: first, methods based on modified objective functions and; second, improved level set approaches.

In the first set, cost function is usually modified such that it has less local optima, but the second set of approaches generally combine the classic level set method with other algorithms to improve it.

A class of shape optimization problems is addressed by new optimality conditions developed by J. SokoLowski et. al, [108]. Topological derivative methods are exploited in the interior of the design domain, as well as boundary variations techniques applied on the boundaries. This method then improved by G. Allaire and F. Jouve [110] in which they combined two main approaches of shape and topology optimization, namely level set method and bubble method. As known, classic level set method uses shape derivative to manipulate boundary propagation, but it is unable to nucleate new holes in the structure. On the other hand, bubble method is introduced to create new holes when needed. Based on what is claimed in this paper, this combination is expected to help mitigating the dependency on initial guess. But in practice, the use of this approach still resulted in dependency and the reason is clear; by using a gradient based approach (for both shape and topology derivatives), the algorithm can still be potentially stuck in local minima.

L. He et al. [111], applied the same idea to find an optimal shapes for maximal band gaps in photonic crystal and acoustic drum problems. Results show that the algorithm has less local optima, although still finding a global solution is a remaining challenge.

The other way to decrease the number of local minima, is to use the filtering techniques that are usual in image processing. For example in an interesting work by S. Amstutz and H. Andra [109] in which instead of using Hamilton-Jacobi equation, an evolution equation has been proposed by generalizing the concept of topological gradient. Finally filtering techniques are employed to make the objective function smoother. It is claimed that by using this approach, not only the creation of new holes is doable, but additionally, all kind of topology changes are achievable.

In second group of attempts, M. Rouhi et al [113] proposed a stochastic direct search method called the Element Exchange Method (EEM). The idea of this method is to exchange the less effective elements to void and conversely, the more influential void elements to solid elements. The random shuffle is intended to discover the design space widely, especially when the checkerboard control scheme is employed. Although some simple examples are investigated in this paper, this method shows the potential capability of stochastic operators in global topology optimization problem.

H. Jia et al. [114] proposed a mixed method to mitigate the aforementioned dependency on the initial guess. Evolutionary structure optimization (ESO) method and level set method are combined such that not only the new hole nucleation allowed, but also the computational expenses are more reasonable compared to ESO. While it is claimed that the algorithm is independent of initial guesses in finding the global solution, all investigated examples have enough holes in their initial step that comparison of the method with other approaches is slightly difficult.

Rong and Liang [115] attempted to eliminate some drawbacks of the level set method. Firstly, a set of new operators is presented to deal with the issue of structural boundary movements. This set of robust and efficient numerical algorithms includes level set regularization, gradient projection, nonlinear velocity, mapping, and return mapping. To cope with the limited ability of level set method to creating new holes during the optimization procedure, they introduce GA-inspired algorithms. This operator helps the proposed algorithm to search the design space more accurately.

According to the literature, the approaches in second part, although effective, do not have a rigorous mathematical foundation, guaranteeing convergence; this holds true especially for modified level set methods. In the next part, a review is carried out on the stochastic level set method that has a solid mathematical foundation. To the best knowledge of the authors, there is no paper in topology optimization of mechanical structures based on this method yet.

1.2.3 Level Set Theory and Optimal Path Planning

Two major categories of path planning methods have been extensively investigated for autonomous mobile robotic applications: local and global planning [123], [124], [125], [126]. As the name states, local methods find a collision-free path, or avoid hazardous regions, they however, usually do not consider global environmental information and therefore may result in trapping scenarios, preventing the algorithm from achieving global optimal states. Global methods, on the other hand, are more likely capable to escape such situations, but they require more computational resources. It is worth mentioning that the original categorization, namely global or local planning, only focuses on availability of information and not necessarily global or local optimal solution of the algorithms. In this thesis, finding the global optimum solution is intended. The main link between these two classifications is, the higher information available for planning, the higher chance to find the global optimal solution and therefore usually using global planning method leads to achieve the best available solution.

The graph-theoretic approaches [127], such as different extensions of $A^\star$ search method [128], are proposed to address the optimal path planning problem [129] globally. These groups of algorithms are applicable even in cases wherein the environmental variables are continuously changing, and consequently the objective function is updated correspondingly. Other methods include PDE-based algorithms, particularly the level set and fast marching methods [54]- [53], are also employed for finding the risk-free and optimal costs paths [130], [131], [132]. Using the level set method, each point on the path evolves along the optimal trajectory in order to minimize the objective function. The aforementioned sets of algorithms are collated in [133] qualitatively. Results show that the level set method [54] and its approximated solution, fast marching method [53], are better suited for path planning in continuous spaces. The authors in [133] have employed optimality metrics instead of Euclidean distance criterion.

Modifying the classic fast marching method, a dynamics version is proposed by [134] in order to re-plan the path when the environment slightly changes. This article undertakes a rigorous investigation on how the environmental changes can affect the level set. This analysis shows that the suggested algorithm is able to reduce the computational costs significantly. Serial and parallel algorithms to solve the discretized version of the Hamilton-Jacobi equation are used in [129] to cope with the trajectory optimization problem. These two algorithms are designed to resemble Dijkstra's and Dial's shortest path algorithms, respectively. The author has shown that the latter algorithm is capable to be parallelized. A fast, consistent and computationally reasonable version of the fast marching algorithm, along with the back-tracking method, is investigated for locating optimal paths for robot navigation in small-size configuration space by Kimmel and Sethian [132].

The most important advantage of this method is that the value function has no local minima, so the global optimum solution can be found by gradient descent of the value function. At the same time, the approach suffers from an issue that the value function which solves the Eikonal equation is rarely differentiable everywhere in the feasible domain. To solve the Eikonal equation, usually the viscosity solution can be obtained by finite difference approximation. This approximation is unique, bounded and differentiable. One can conclude that although the approach leads to the global solution of the optimal path planning problem, implementation of the algorithm particularly when some constraints exist is quite challenging. A method with a grid model, a new consumption function, and a search method is proposed by Na and Ping [135] to reduce the complexity of the level set method for path planning applications. Cecil and Marthaler, in [136], extended the two-dimensional path planning algorithm proposed by Canny [137], which is a known method in variational framework to three-dimensional space. They defined an energy integral over the path and let the path evolve along the gradient flow to obtain a local optimal solution. One major problem in this kind of formulation is that the algorithm is highly dependent on definition of energy and analytical gradient flow, which is not usually available in real engineering applications. Furthermore, trapping in local solution is a drawback in this formulation.

Most of the existing algorithms, even those considering global information, might trap in local solutions. Moreover, analytical gradient is not always available in active contour methods. Obviously, numerical calculation of gradient needs considerable computational resources.

1.3 Stochastic Flows and Level Set Method

As stated, to deal with general nonlinear optimization problem, stochastic based operators are known as effective tools [1]. Therefore, to utilize the level set method a review on stochastic partial differential equations and stochastic integration is significantly helpful. In engineering application, stochastic level set method was initially applied to image processing and segmentation [57].

The idea of stochastic flow, as proposed by Walsh [70], is based on stochastic partial differential equations. This concept has been employed in the modeling of physical phenomena and even computer vision. Yip [71] has performed a literature review on stochastic motion by mean curvature that shows the recent trends in this field. One of the most interesting set of works on the viscosity solution of a stochastic partial differential equation has been done by Lions and Souganidis in a series of articles [72]- [73] which are very helpful for convergence analysis of stochastic level set method with spatial free noise source.

A stochastic curve evolution has been implemented by Juan et al. [57] in a level set framework. By using a Stratonovich differential, the well-posedness of the evolution is achieved. In addition, the final solution becomes almost independent of the initial curve. The stochastic active contour proposed in this reference is able to deal with the local optima problem and even with a complicated circumstance in which the derivation of the exact gradient is not possible. From mathematical point of view, the model that is developed in [57] is more general than the model investigated in [72]- [73], because [57] has proposed a spatial dependent noise source, to get a better performance.

A very comprehensive extension to [57] has been done in [74], where several aspects of simulation of stochastic partial differential equations are investigated. In addition to the extension of previous work by Juan [57] , a set of definitions for numerical stability of stochastic partial differential equation (SPDE) has been suggested. M. Caruana and co-authors in [81] has proposed a (rough) path wise approach by which the convergence of SPDE when the noise source is spatial dependent and linear in $D\phi$, gradient of evolving surface, is investigated. The main results of this reference [81] will be discussed in detail in Section 3.2.1, background theory, when the convergence of the proposed algorithm, ACFS, is considered.

Law et al. [145] investigate global minimization of the multi-phase piecewise constant Mumford-Shah model [63]. To do so, a hybrid method containing the gradient-based and stochastic method has been proposed in order to deal with the dependency of the solution on initial guesses. It is claimed that the algorithm is very successful compared with other stochastic algorithms such as simulated annealing. Chen et al. [75] extended the conventional random walk Metropolis- Hastings method to high-dimensional curves; the asymptotic convergence of the Markov chain is also proved. To accomplish this, the authors used an implicit representation of shapes instead of conventional explicit curve parametrization. They claim that their algorithm can be easily applied to 3D frameworks.

Pan and others in [76], proposed a new formulation for active contours, in which the conventional state estimation is employed for evolving the curves. As a result, a hybrid algorithm based on level set method and particle filters has been created. Results show that this new algorithm works well for complex images.

An interesting comparison has been made in [145] on various level set based methods. Based on this article, the pure stochastic level set algorithm has the best performance in most of the cases in terms of finding the most global solution. In comparison to other algorithms like "Gradient Descent", "Simulated Annealing", or a hybrid method called "Multi-resolution Stochastic Level Set Method", the only drawback associated with stochastic level set method is that it has no systematic procedure to control the random terms. Therefore, this algorithm has to search even those solutions that increase the cost function significantly, which slows the procedure. On the other hand, it is worth mentioning that although all of these algorithms employ stochastic operators in order to escape from local optima, finding the global solution is not yet guaranteed. The problem of a global optimum point is one of the most challenging open problems in mathematics.

Based on the existing algorithms in optimization theory, there are two main approaches to attacking the problem: using random operators and multi-individual (agents) based algorithms. The first one helps the procedure to escape from local optima, and the second one assists in the exploration of the whole feasible region for a solution. Obviously, the more individuals (agents) are involved in the hunt, the higher the probability of finding the global optimum point. This point has been proved in all of the multi-individual (agent) based algorithms such as GA and Particle Swarm optimization method (PSO), including multiagent gradient-based methods [1].

Essentially, along with the aforementioned operators, an algorithm for convergence is required. There is always a tendency for researchers to prove the convergence of their algorithm mathematically. In this regard, GA has a drawback that there is no mathematical proof for its convergence. Even for PSO, the mathematical proofs cope with some simplified models. In level set, there is a very strong mathematical proof even in stochastic framework [72]- [73] and [81]. As long as some conditions are satisfied, the convergence of the stochastic level set method can be proved for some special cases; thus, for any further extension, one should make sure that the above-mentioned conditions are not violated.

1.4 Contributions of the Dissertation

This study is intended to develop a set of new algorithms to deal with global optimization problems in variational framework. In doing so, a new global optimization algorithm is developed in parameter space and then its extension to the variational framework is investigated. In brief, the main contributions of this work are as follows:

1. Development of a global optimization algorithm in a parameter space, called Spiral Bacterial Foraging Optimization (SBFO). Basically, this algorithm is a multi-agent stochastic gradient method.

2. Development of two algorithms based on stochastic level set method, active contours with stochastic fronts (ACSF), and Multi-agent stochastic level set method (MSLSM).

3. Application of ACSF and MSLSM in image processing and topology optimization problems.

4. Development of another (third) algorithm based on stochastic level set method with the absence of the gradient function, called Evolving Arc (E-ARC).

5. Application of E-ARC in optimal path planning problem.

1.5 Outline of the Dissertation

Chapter 2 discusses deterministic and stochastic level set methods. It basically contains a brief explanation on the classic level set method and the derivation of required stochastic level set equations that are developed on [57]. Besides, the main results of [72]- [73] and [81] are presented as a background for convergence analysis.

The new algorithms for parameter optimization and variational framework are proposed in Chapter 3. Section 3.1 is devoted to the development of a new global parameter optimization method. Considering the existing global optimization approaches, and considering their advantages and drawbacks, SBFO as a multi-agent stochastic gradient method is proposed. Then, two methods in variational framework are developed and discussed. For the first method, namely ACSF, a mathematical proof of convergence is also offered. It is shown that the general stochastic model proposed in [57] does not necessarily converge towards the deterministic model; luckily the proposed algorithm, ACSF, converges, and it has been investigated in detail. The second approach, MSLSM, is based on the ideas learned from SBFO in parameter space in which a multi-agent stochastic structure is more capable to find the global solution.

Sections 4.1 and 4.2 of Chapter 4 are assigned to direct applications of ACSF and MSLSM to engineering examples. Image processing, or more specifically image segmentation, is one of the most important branches of engineering in which level set theory and its extensions are employed. Section 4.1 starts with the formulation of the problem and then the results of ACSF and MSLSM are discussed. As can be seen, in contrast to what is expected, MSLSM does not show a very significant improvement in results compared to ACSF. The reasons are listed in the corresponding section, but as a result, only ACSF is employed for the topology optimization problem. In Section 4.2, topology optimization problem is formulated and results of ACSF is compared with classic level set method. In Section 4.3, another problem is addressed. In some engineering applications such as optimal path planning, analytical gradient functions do not exist and numerical calculation of the gradient is obviously very expensive. E-ARC algorithm that is inspired by "Electric Arc" is proposed in this section. A brief analogy of electric arc and optimal path planning, followed by the definition of the algorithm and results are the main topics of this section.

Finally, Chapter 5 contains conclusions and possible future works.

Chapter 2

Background to Deterministic and Stochastic Level Set-based Optimization

2.1 Level Set Theory and Optimization:

Implicit description of curves and surfaces is the main concept of the level set method. This capability allows the level set method to be used as a multi-dimensional tool to represent any complex topology. Changes in topology start from a zero level set or isophote as an initial guess, which continuously evolves through a Hamilton-Jacobi PDE formulation based on a certain cost function [52]- [54] and [63]- [64].

Let's assume there is an implicit function, $\phi(x,t)$, in a given design domain, Ω, which satisfies:

$$\begin{cases} \phi(x,t) > 0, & x \in \Omega^{+} & \text{Exterior} \\ \phi(x,t) < 0, & x \in \Omega^{-} & \text{Interior} \\ \phi(x,t) = 0, & x \in \partial\Omega & \text{Boundaries} \end{cases} \tag{2.1}$$

and also two other well-known Heaviside, $H(s)$, and Dirac delta, $\delta(s)$, functions:

$$H(s) = \begin{cases} 1 & s \geq 0 \\ 0 & s < 0 \end{cases} \tag{2.2}$$

$$\delta(s) = \begin{cases} 0 & s \neq 0 \\ \infty & s = 0 \end{cases} \tag{2.3}$$

This powerful tool can be used as a dynamic representation of boundaries in a wide range of engineering applications such as image processing, mechanical topology optimization and optimal path planning. In most of the applications a shape or topology should be found such that a particular objective function is optimized, hence, the dynamic boundaries should evolve such that at the end of the process, at least one of the local optimal solutions is found.

Euler-Lagrange Theorem [53]- [54] is employed to formulate the optimization problem in variational framework. This formulation needs shape derivatives or sensitivity of the objective function with respect to the evolving boundaries.

2.1.1 Shape Derivatives:

Sensitivity of the objective functional with respect to boundary perturbation is determined by shape derivatives. Murat and Simon have defined this measure as follows [94] , [111]:

Assume that a perturbation occurs in $\Omega \in D \subset \mathbb{R}^N$ under a map of $\theta \in W^{1,\infty}(\mathbb{R}^N, \mathbb{R}^N)$ such that $\|\theta\|_{W^{1,\infty}} < 1$:

$$\Omega_\theta = (I + \theta)\Omega \tag{2.4}$$

in which $W^{1,\infty}$ is Sobolov space [94], I is the identity operator and the set Ω_θ is defined as [111]:

$$\Omega_\theta = \{x + \theta(x) \mid x \in \Omega\} \tag{2.5}$$

The shape derivative of an objective functional $J : \mathbb{R}^N \to \mathbb{R}$ at Ω is equivalent to Frechet differential $J(\Omega_\theta)$ at 0. It has been shown that the shape derivative is only dependent on $\theta \cdot n$ on the boundary $\partial\Omega$ because the shape does not change if θ is in tangential direction of the domain Ω. An objective functional in the form of integral over the domain Ω or along its boundary can be formulated as [111]:

$$J(\Omega) = \int_\Omega f(x)dx \tag{2.6}$$

It has been shown that the shape derivative of this functional is [111]:

$$\partial_S J(\Omega, \theta) = \int_\Omega \nabla \cdot (\theta(x)f(x)) = \int_{\partial\Omega} \theta(x) \cdot n(x)f(x)ds \tag{2.7}$$

If the functional is defined as an integral over the boundary such as [111]:

$$J(\Omega) = \int_{\partial\Omega} f(x)dx \tag{2.8}$$

the shape derivative is [111]:

$$\partial_S J(\Omega, \theta) = \int_{\partial\Omega} \theta(x) \cdot n(x)(\frac{\partial f}{\partial n} + \kappa f)ds \tag{2.9}$$

where κ is the mean curvature of $\partial\Omega$ and can be defined as [111]:

$$\kappa = \nabla \cdot n \tag{2.10}$$

As can be seen both shape derivatives (2.7) and (2.9) only depend on boundary.

Now to optimize the functional, the gradient flow should be determined. One can suppose that the shape derivative is in the form of [111]:

$$\partial_s J(\Omega, \theta) = \int_{\partial\Omega} \theta(x) \cdot n(x) V_n(\Omega, x)ds \tag{2.11}$$

For minimization of the objective functional one can choose the gradient flow as [111]:

$$\theta(x) = V_n(\Omega, x)n(x) \tag{2.12}$$

or basically V_n is the normal velocity at each point of the evolving boundary. Employing the function ϕ to represent the boundary of Ω, the Hamilton-Jacobi equation can simulate the motion under normal velocity.

$$\phi_t + V_n(\Omega, x)\left|D\phi\right| = 0 \tag{2.13}$$

It has been shown [52] that the steady state solution of Equation(2.13) is a local optimal solution of $\partial_s J(\Omega, \theta) = 0$. This can show the importance of using the level set theory in optimal variational problem.

2.1.2 Topology Derivatives:

Topology derivatives are designed to measures the sensitivity of functional to the creation of a small hole at a certain point instead of boundary perturbation for shape derivatives. To find this measure, a small ball $B_{\rho,x}$ centered at point x and radius ρ should be created and then the variation of J with respect to the volume of this small ball should be considered. For $x \in \overline{\Omega}$ at which the material should be removed the topological derivative $\partial_T J(\Omega, x)$ is defined as a limit:

$$\partial_T J(\Omega, x) = \lim_{\rho \longrightarrow 0} \frac{J(\Omega_{\rho,x}) - J(\Omega)}{|B_{\rho,x} \cap \Omega|} \tag{2.14}$$

where $\Omega_{\rho,x} = \Omega - \overline{B(\rho, x)}$. To incorporate the topology derivative, it has been derived for objective functional (2.6) and is represented as [111]:

$$\lim_{\rho \longrightarrow 0} \frac{J(\Omega_{\rho,x}) - J(\Omega)}{|B_{\rho,x} \cap \Omega|} = -V_n(\Omega, x) \tag{2.15}$$

As mentioned, this formulation is derived based on the idea of subtracting material from $\overline{\Omega}$; obviously, in some cases it would be a great idea if the algorithm could add material in Ω region if needed. So, the topology derivative should be reformulated:

$$T_n(\Omega, x) = \begin{cases} -V_n(\Omega, x) & \text{when } x \in \overline{\Omega} \\ V_n(\Omega, x) & \text{when } x \in \Omega \end{cases} \tag{2.16}$$

One can notice that it might not be easy to find the shape derivative a general objective functional, but topology derivative can be achieved as long as shape derivative is derived and the key part, namely $V_n(\Omega, x)$, exists.

2.1.3 Incorporating shape and topological derivatives into the level set method

Based on the literature, one way to improve the trapping scenarios in local solution is to incorporate topology derivatives into level set method. The new equation designed for this purpose is as follows:

$$\phi_t + V_n |D\phi| + wT_n = 0 \tag{2.17}$$

where T_n is the modified topology derivative in Equation (2.16) and w is a positive scalar to balance the effects of shape and topology derivatives.

Combining the Equation (2.16) and (2.17), the final equation becomes [111]:

$$\phi_t = -V_n(|D\phi| + w) \tag{2.18}$$

2.1.4 Computational Scheme

The main advantage of using level set theory is to have a systematic approach (by solving a PDE) to solve the algebraic equation $\partial_s J(\Omega, \theta) = 0$, which is not possible in general without using LSM. There are many different numerical method to solve the Hamilton-Jacobi Equation (2.13), but in this thesis only "Upwind Difference Scheme" is discussed and employed to solve the equation. This approach is one of the well-known methods called "Finite Difference Methods" (FDM). Some literatures use "Finite Element Methods" (FEM) to deal with this problem but our experience with both methods show finite difference methods are strong enough to perform this analysis. Moreover they are much easier in terms of implementation.

As stated acquiring the solution of Equation (2.13), upwind difference scheme is employed in this thesis. The main characteristic of this approach is that, the forward and backward differences are adaptively calculated at any point depend upon the direction of the velocity field [54]- [53].

For 2D applications the first order upwind scheme is as follows:

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} - \Delta t(max((V_n)_{ij}, 0)\nabla^+ + min((V_n)_{ij}, 0)\nabla^-) \tag{2.19}$$

where

$$\triangledown^+ = \{max(D_{ij}^{-x}, 0)^2 + min(D_{ij}^{+x}, 0)^2 + max(D_{ij}^{-y}, 0)^2 + min(D_{ij}^{+y}, 0)^2\}^{\frac{1}{2}} \tag{2.20}$$

$$\triangledown^- = \{max(D_{ij}^{+x}, 0)^2 + min(D_{ij}^{-x}, 0)^2 + max(D_{ij}^{+y}, 0)^2 + min(D_{ij}^{-y}, 0)^2\}^{\frac{1}{2}} \tag{2.21}$$

The indexes of ϕ are defined to represent the relative positions of finite difference nodes and shown in Figure (2.1) [54]- [53].

The chosen time step Δt should satisfy the Courant-Friedrichs-Lewy (CFL) conditions [54]- [53] as follows:

$$\Delta t = \frac{min(\Delta x, \Delta y)}{max\,|(V_n)ij|} \tag{2.22}$$

where Δx and Δy are horizontal and vertical directions respectively, and $D_{ij}^{\pm x}$ and $D_{ij}^{\pm y}$ are also forward and backward finite difference operators that are defined in detail in the following equations [54]- [53]:

$$\begin{aligned} D_{ij}^{+x} &= (\phi_{i+1,j} - \phi_{ij})/\Delta x \\ D_{ij}^{-x} &= (\phi_{ij} - \phi_{i-1,j})/\Delta x \\ D_{ij}^{+y} &= (\phi_{i,j+1} - \phi_{ij})/\Delta y \\ D_{ij}^{-y} &= (\phi_{ij} - \phi_{i,j-1})/\Delta y \end{aligned} \tag{2.23}$$

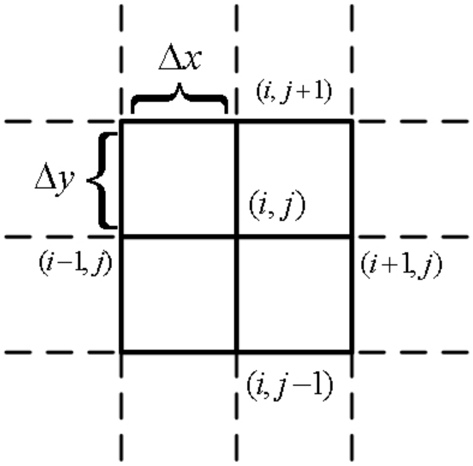


Figure 2.1: Definition of nodes' indexes in FDM

2.1.5 Geometric Quantities as Function of ϕ

Usually the objective function of optimization problems is written as a combination of some often-used quantities like normal vector, curvature, perimeter, area. Here some of these components will be expressed as a function of level set function $\phi(x)$ [120].

1. Normal Vector $\overrightarrow{N}$: This vector points in the direction of steepest increase

$$\overrightarrow{N} = \frac{\nabla \phi}{|\nabla \phi|} \tag{2.24}$$

2. Mean Curvature κ: The divergence of the normal vector $\overrightarrow{N}$ indicates the mean curvature of the interface

$$\kappa = \nabla \cdot \overrightarrow{N} = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \tag{2.25}$$

3. Perimeter of the boundary $|\partial \Omega|$:

$$|\partial \Omega| = \int_{\Omega} \delta(\phi)\, |\nabla \phi|\, d\Omega \tag{2.26}$$

4. Area $|\Omega|$:

$$\int_{\Omega} H(\phi) d\Omega \tag{2.27}$$

2.1.6 Re-Initialization:

Level set method is based on the evolution of an implicit function defined by a surface. If during the evolution process, the surface becomes too steep or too flat, the process might becomes unstable or too slow which either cases are not desirable. One way to resolve this issue is re-initialization. The basic idea behind the re-initialization is, if one employs "Signed Distance Functions" as "level set function", they usually result in more accurate, stable and relatively fast enough evolution. Signed distance function is a subset of implicit functions and defined in Equation (2.28):

$$\phi(x) = \begin{cases} d(x), & \forall x \in \Omega^{+} \\ 0 & \forall x \in \partial \Omega \\ -d(x) & \forall x \in \Omega^{-} \end{cases} \tag{2.28}$$

in which $d(x)$ is the distance function defined as:

$$d(x) = min(|x - x_I|), \text{for } \forall x_I \in \partial \Omega \tag{2.29}$$

By a simple investigation, it can be seen that:

$$|\nabla \phi| = 1, \quad \text{for } \forall x \in \Omega \tag{2.30}$$

This special feature can prevent the evolving surface to becomes too steep or too flat because for all points on the surface the length of the gradient vector has the same size. So, it seems that the level set function should be regularized such that it is close to signed distance function as much as possible. This method is called "re-initialization". Doing so, the re-initialization process should be done after a particular number of iterations. This number should be chosen by trial and error. Basically, the higher the number, the faster computation and less accurate results.

2.2 Stochastic Differential Equations

Stochastic differential equations (SDE) has been widely used in many branches of science and engineering such as biology, economics and finance, chemistry, MEMS and NEMS applications. Advanced probability and stochastic processes are the main prerequisites for complete understanding of SDE but a very brief review on concepts is intended here and for more details, the interested readers are encouraged to refer to some extensive books such as [84], [86] and [87]. Starting with the idea of Brownian motion, stochastic integration will be discussed as key to solving the SDE. All the operators are defined in a standard probability space $(\omega, \mathcal{F}, \mathcal{F}_t, \mathbb{P})$. For definitions of standard probability space and other introductory concepts please refer to [84], [86] and [87].

2.2.1 Brownian Motion:

A standard $m-$dimensional Brownian motion or standard Wiener process, over $[0, T]$ is a random vector $W(t) = [W_1(t), \ldots, W_m(t)]_{t \geq 0}$ that is a continuous function of time $t \in [0, T]$ and satisfies the conditions as follows [85]:

1. $W(0) = 0$ (almost sure or with probability 1)

2. Given by increment $W(t)-W(s)$, for $0 \leq s < t \leq T$, the random variable vector has a normal distribution with mean zero and variance $t-s$; in other words $(W(t)-W(s)) \sim \sqrt{(t-s)}N(0,1)$, where $N(0,1)$ is a symbol for normally distributed random variable with zero mean and unit variance.

3. The increments $W(t) - W(s)$ and $W(v) - W(u)$ are independent, $\forall \ \ 0 \leq s < t < u < v \leq T$.

In addition to these conditions Brownian motion has some other properties such as [88]:

1. $W(t)$ has continuous paths ($\frac{1}{2}$-holder continuous)
2. $W(t)$ is nowhere differentiable

2.2.2 General Theory:

A common representation of stochastic differential equation is shown as [88] :

$$dX_t = G(X_t, t)dt + H(X_t, t)dW_t \tag{2.31}$$

where $X_t = X_t(t)$ is a realization of the stochastic process. $G(X_t, t)$ and $H(X_t, t)$ are called drift and diffusion coefficient, respectively. The drift coefficient is the deterministic part of the equation to determine the local trend of the process, and correspondingly diffusion coefficient is the stochastic part of the coefficient that shows the average fluctuation of X_t. Assuming that the stochastic part obeys the necessary conditions for Wiener process, one may be interested to solve the stochastic Equation (2.31) in order to find at least one realization, X_t. To find the realization, one should integrate Equation (2.31) as [88]:

$$X_t = X_{t_0} + \int_{t_0}^{t} G(X_s, s)ds + \int_{t_0}^{t} H(X_s, s)dW_s \tag{2.32}$$

First integral in Equation (2.32) is an ordinary Riemann integral. For second integral, it should be noted that the Brownian processes are not differentiable. To cope with this situation, Ito, a Japanese mathematician, proposed "Ito Stochastic Integral" in 1940, and 20 years after that, a Russian physicist, R.L. Stratonovich, suggested another stochastic integral in order to solve Equation (2.31). To identify the difference between Ito and Stratonovich integrals, usually Stratonovich integral is shown by a specific operator " $\circ$ " [57].

$$dX_t = G(X_t, t)dt + H(X_t, t) \circ dW_t \tag{2.33}$$

and correspondingly the integral form can be represented as:

$$X_t = X_{t_0} + \int_{t_0}^{t} G(X_s, s)ds + \int_{t_0}^{t} H(X_s, s) \circ dW_s \tag{2.34}$$

In general, the second integral can be reformulated as:

$$\int_{t_0}^{t} H(X_s, s)dW_s = \lim_{h \to 0} \sum_{k=0}^{n-1} H(X_{\tau_k}, \tau_k)(W(t_{k+1}) - W(t_k)) \tag{2.35}$$

in this equation $h = (t_{k+1} - t_k)$ with intermediary points $\tau_k = (1 - \lambda)t_k - \lambda t_{k+1}$, $\forall k \in \{0, 1, \ldots, n-1\}$, $\lambda \in [0, 1]$. Depend upon how the parameter λ is defined, different integrals are introduced. The instance of $\lambda = 0$ results in $\tau_k = t_k$, meaning that the integral is evaluated at the start point of each interval. This method is called Ito integral. Additionally, $\lambda = 1/2$ gives $\tau_k = (t_{k+1} - t_k)/2$, that is corresponding to Stratonovich integral [88].

Ito integral is usually employed in financial mathematics in which only the information about past is taken into account for modeling. In contrast, Stratonovich is very common in physical science modeling. Based on aforementioned formulation, the Ito integral can be written as follows:

$$\int_{0}^{T} H(s)dW_s = \lim_{h \to 0} \sum_{k=0}^{n-1} H(t_k)(W(t_{k+1}) - W(t_k)) \tag{2.36}$$

Similarly, in Stratonovich case, the next two integrals can be employed. The equality of these two integrals is proven in [57].

$$\int_{0}^{T} H(s) \circ dW_s = \lim_{h \to 0} \sum_{k=0}^{n-1} H(\frac{t_k + t_{k+1}}{2})(W(t_{k+1}) - W(t_k)) \tag{2.37}$$

$$\int_{0}^{T} H(s) \circ dW_s = \lim_{h \to 0} \sum_{k=0}^{n-1} (\frac{H(t_k) + H(t_{k+1})}{2})(W(t_{k+1}) - W(t_k)) \tag{2.38}$$

2.2.3 Stochastic Flow Modeling

According to Equation (2.31), in general, the modeling of stochastic processes is based on a deterministic term (drift) and a stochastic term (diffusion). In stochastic flow modeling or curve evolution, the deterministic term can be shown as $G = G(D^2\phi, D\phi, x, t)$ in which ϕ is an implicit representation of the curve or surface and D is gradient operator and D^2 denotes the Hessian matrix. General form of a fully nonlinear stochastic partial differential equation in $\mathbb{R}^N \times (0, \infty)$ is [72]- [73]:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + \sum_{i=1}^{m} H_i(D\phi, x, t)dW_i(x, t) \tag{2.39}$$

where m is the number of independent noise sources $W_i(x, t)$, $i \in \{1, 2, ..., m\}$.

In a special case, where $m = 1$, a stochastic model for curve evolution has been proposed as follows [57]:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + |D\phi|\, dW(t) \tag{2.40}$$

One may notice that in Equation (2.40), Ito operator has been chosen implicitly. Obviously Equation (2.40) in Stratonovich sense can be shown as:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + |D\phi| \circ dW(t) \tag{2.41}$$

As can be seen in Equations (2.40)-(2.41), all points have an extra random force (second term) that is not explicitly a function of position because it is only depends on $|D\phi|$ and Brownian process which is a function of time. A more flexible structure can be achieved by replacing the Brownian motion $W(t)$, by a colored spatial noise.

$$W(x, t) = \sum_{i=1}^{m} \psi_i(x)W_i(t) \tag{2.42}$$

where $\psi_i : \mathbb{R}^N \longrightarrow \mathbb{R}$ are smooth functions with compact support. Thus, the final stochastic evolution model is as follows [57]:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + |D\phi| \sum_{i=1}^{m} \psi_i(x) \circ dW_i(t) \tag{2.43}$$

In order to simplify the model, one can use the same type of function ψ but at different centers. $\psi_i(x) = \psi(x - x_i)$, where ψ can be a convenient regular function.

2.2.4 Convergence Analysis

A rigorous mathematical convergence analysis has been conducted by Lion [72]- [73] in which the authors have suggested the notion of stochastic viscosity solution for fully nonlinear, second order, possibly degenerate stochastic partial differential equations. The following theorem is the main result of [72].

Theorem 1. *Consider the following two equations:*

$$d\phi = G(D^2\phi, D\phi, x, t)dt + \epsilon \left|D\phi\right| \circ dW(t) \text{ with } \phi(\cdot, 0) = \phi_0(\cdot) \tag{2.44}$$

$$d\phi = G(D^2\phi, D\phi, x, t)dt + \left|D\phi\right| \dot{\xi_\alpha}(t) \text{ with } \phi(\cdot, 0) = \phi_0(\cdot) \tag{2.45}$$

where $\epsilon \geq 0$ and $\xi_\alpha : \mathbb{R}^+ \to \mathbb{R}$ is a family of smooth functions. It is "Almost Surely" true that in ω:

1. *Equation (2.44) has a unique solution.*
2. *Assuming $\{\xi_\alpha(t)\}_{\alpha>0}$ and $\{\zeta_\beta(t)\}_{\beta>0}$ are two families of smooth functions in such a way that if α and $\beta \to 0$, then $\xi_\alpha$and ζ_β converge to W uniformly on any compact set in t and almost surely in ω. Let $\{u_\alpha\}_{\alpha>0}$ and $\{v_\beta\}_{\beta>0}$in $BUC(\mathbb{R}^+ \times \mathbb{R}^N)$, namely a Bounded Uniformly Continuous subset of $\mathbb{R}^+ \times \mathbb{R}^N$, be the solution of Equation (2.44). Now, suppose that $\lim_{\alpha,\beta\to 0} \|u_\alpha(\cdot, 0) - v_\beta(\cdot, 0)\|_{C(\mathbb{R}^N)} = 0$, then for all $T > 0$, $\lim_{\alpha,\beta\to 0} \|u_\alpha - v_\beta\|_{C([0,T]\times\mathbb{R}^N)} = 0$. In other words, any smooth approximation of W makes a series of solutions converging to the unique solution of (2.44).*
3. *The solution u^ϵ of (2.45) is convergent to the solution of (2.44) in $C(\mathbb{R}^+ \times \mathbb{R}^N)$ when $\epsilon \to 0$.*

As can be seen, the theorem can only guarantee the convergence of Equation (2.43) when $\psi_i = 1$ or basically when the noise is not spatial-dependent [57]. Although the general case of this theorem is still an open problem in stochastic partial differential equation theory, recently some more general cases have been studied and luckily, these new studies can support our modified version of stochastic level set method presented in Section 3.2.1. These theorems were not available when Juan et al. [57] published their paper. The newer theorems will be discussed in detail in Section 3.2.1.

2.2.5 Numerical Integration

As stated, in contrast to Ito integral, Stratonovich formulation does not have an explicit numerical scheme because it does not only depend on start point in each time interval (see Equation 2.35). For more clarification, if one assumes the simple evolution $d\phi = |D\phi| \circ dW(t)$, the corresponding method for its implicit update scheme would be:

$$\phi_{i+1} = \phi_i + \frac{1}{2}(|D\phi_i| + |D\phi_{i+1}|)\Delta W_i \tag{2.46}$$

To avoid encountering implicit update formulation such as Equation (2.46), let's consider the following equation in Stratonovich sense:

$$d\phi = H(D\phi, x) \circ dW(t) \tag{2.47}$$

A typical example for $H(\eta, x)$, a real valued function in $\mathbb{R}^N \times \mathbb{R}$, is $H(\eta, x) = |\eta|\, \psi(x)$. Here ψ is any regular smooth function. Expanded form of Equation (2.47) in Stratonovich sense [57] is:

$$d\phi = H(D\phi, x)dW(t) + \frac{1}{2}d\left\langle H(D\phi, x), W\right\rangle(t) \tag{2.48}$$

This expansion shows that Stratonovich term is actually Ito term plus an additional term, called drift.

To manipulate the drift term, second term, one can start with integration of Equation(2.47):

$$\phi(x,t) = \phi_0(x) + \int_0^t H(D\phi(x,s), x) \circ dW(s) \tag{2.49}$$

By taking the derivative of both sides with respect to the dependent variable x:

$$\begin{aligned} D\phi(x,t) = D\phi_0 + \int_0^t & [D^2\phi(x,s) D_\eta H(D\phi(x,s), x) \\ & + D_x H(D\phi(x,s), x)] \circ dW(s) \end{aligned} \tag{2.50}$$

in which D_η and D_x are gradient operator with respect to η and x respectively. Using Ito rule [57] leads to:

$$\begin{aligned} H(D\phi(x,t),x) &= H(D\phi_0(x),x) \\ +\int_0^t [D_\eta H\cdot(D^2\phi D_\eta H) &+ D_\eta H\cdot D_x H]\circ dW(s) \end{aligned} \tag{2.51}$$

Now the extra term in Equation (2.48) can be obtained as follows:

$$\begin{aligned} \frac{1}{2}\left\langle H(D\phi,x),W(t)\right\rangle(t) = \frac{1}{2}\int_0^t [D_\eta H\cdot(D^2\phi D_\eta H) \\ + D_\eta H\cdot D_x H]ds \end{aligned} \tag{2.52}$$

As it is assumed that $H = |\eta|\,\psi(x)$, the aforementioned formula in Equation (2.52) changes to:

$$\begin{aligned} &\left\langle H(D\phi,x),W(t)\right\rangle(t) = \\ &\int_0^t [\psi^2(t)\frac{D\phi}{|D\phi|}\cdot D^2\phi(x,s)\frac{D\phi}{|D\phi|} + \psi(x)D\psi(x)\cdot D\phi(x,s)]ds \end{aligned} \tag{2.53}$$

The second order term in Equation (2.53) can be simplified to [57]:

$$\begin{aligned} \frac{D\phi}{|D\phi|}\cdot D^2\phi(x,s)\frac{D\phi}{|D\phi|} = \\ \Delta\phi - |D\phi|\operatorname{div}(\frac{D\phi}{|D\phi|}) = \Delta\phi - |D\phi|\,\kappa \end{aligned} \tag{2.54}$$

where "div" is divergence operator and κ is the mean curvature of level set. Equation (2.53) is obtained based on this assumption that "m", the number of Brownian motions, is 1. For a more general case in which the stochastic evolution PDE is of type:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + |D\phi|\sum_{i=1}^{m}\psi_i(x)\circ dW_i(t) \tag{2.55}$$

the expanded version of Equation (2.55) can be represented as follows [57]:

$$
\begin{aligned}
d\phi = G(D^2\phi, D\phi, x, t)dt + |D\phi| \sum_{i=1}^{m} \psi_i(x) dW_i(t) + \\
\frac{1}{2}\left(\left(\sum_{i=1}^{m} \psi_i^2(x) \right) (\Delta\phi - |D\phi|\,\kappa) + \left(\sum_{i=1}^{m} \psi_i(x) D\psi_i(x) \right) \cdot D\phi \right) dt \qquad (2.56)
\end{aligned}
$$

Chapter 3

Proposed Methods for Global Optimization

3.1 A New Parameter Optimization Algorithm

A biologically-inspired algorithm called "**S**piral **B**acterial **F**oraging **O**ptimization" is presented in this section in order to search for the global optimum of multi-modal objective functions. The proposed algorithm is a multi-agent, gradient-based algorithm, which minimizes both the main objective function (local cost) and the distance between each agent and a temporary central point (global cost). A random jump is included normal to the connecting line of each agent to the central point, which produces a vortex around the temporary central point. This random jump is also suitable to cope with premature convergence that is a feature of swarm-based optimization methods. The most important advantages of this algorithm are as follows: First, this algorithm involves a stochastic type of search with a deterministic convergence. Second, as gradient-based methods are employed, faster convergence is demonstrated over GA, DE, BFO and etc. Third, the algorithm can be implemented in a parallel fashion in order to decentralize large-scale computation. Fourth, the algorithm has a limited number of tunable parameters, and finally mathematical proof of convergence for SBFO, which is rare in existing global optimization algorithms, is shown. A detailed convergence analysis of SBFO is also investigated in this section.

3.1.1 Spiral Bacterial Foraging Optimization

To mitigate the issues discussed in Section 1 and develop a basis for a method in variational framework, in this section Spiral Bacterial Foraging Optimization (SBFO) method is proposed. SBFO algorithm, as a global optimization algorithm that is categorized under particle swarm algorithms (PSO), has been initially introduced in [1]. The most important asset of SBFO is its use of gradient operators in addition to stochastic operators and multi-agent structure. Employing gradient-based operators not only helps to improve the speed of convergence but it is also beneficial for ensuring the convergence of the algorithm. Roughly speaking, each agent in SBFO is driven by a stochastic gradient toward the best agent.

In SBFO, the aggregation process of Dictyostelium cells is imitated in which the cells secrete cAMP protein to attract their neighbors by chemotaxis. The cells move along the gradient of cAMP (local search) to join other cells to build the central mound (global optimum). A spiralling motion toward the central optima provides an opportunity to further search the feasible region.

Although the algorithm is not completely committed to the real biological process, some major numerical problems in previous techniques, such as premature convergence, large number of tunable parameters and the speed of convergence, are improved. The conventional global search methods are too slow for large-scale problems like bioinformatics/protein folding, artificial intelligence and structural topology optimization, and as gradient-based methods are employed in SBFO, this present algorithm converges significantly faster than other global optimization algorithms.

Aggregation of Dictyostelium Cells

The complex life cycle of cellular slime mold, “Dictyostelium discoideum”, is widely considered to be an interesting example in developmental biology [26]- [27]. When there is plentiful food in their environment, Dictyostelium discoideum cells live individually, called “amoeba”, but once the food source starts to decline, the amoebae aggregate together to create a mound of cells. The mound then reshapes to a slug form to search for a new place with a suitable source of food, moisture or even light. After that, the slug changes to a fruiting body, including stalk and spore cells. The movement (Chemotaxis) of the individual cells and also the slug is attributed to a chemical attractant signal cAMP. cAMP is emitted by cells while they are migrating towards the main source of the received signal. The stage in which the cells move due to the chemical attractant is called the Aggregation process. To reach the central point (main source), the cells follow a spiral pattern [27] which can be helpful to guarantee a better performance in exploring the medium for food.

Description of the SBFO Algorithm

As stated, existing BFO methods have several notable drawbacks, including premature convergence, tuning of parameters and speed of convergence. Premature convergence can lead to loss of diversity; as a result, the feasible space would not be explored sufficiently. In other words, when two or more agents encounter each other along the same path, there is no more to be gained by having more than one agent on that path. Figure (3.1) shows premature convergence and how it can affect the performance of the algorithm. It is worthwhile to mention that the ideal convergence for agents is when they encounter each other precisely at the best point – namely, that point where they are able to continue exploring the feasible region separately and can thereby preserve the diversity of the agents.

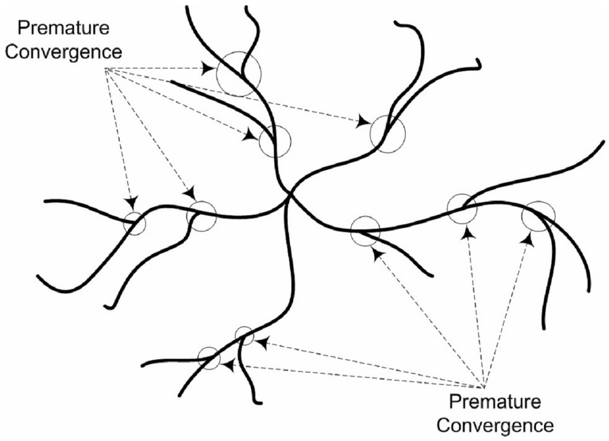


Figure 3.1: premature convergence lead to loss of diversity

Figure (3.2) depicts ideal convergence, in which the agents explore the space more competently in comparison with Figure (3.1), which shows an algorithm with the same number of agents in premature convergence. Parameter tuning is another flaw in evolutionary and swarm particle optimization methods as well as in BFO algorithms, as the proper values of parameters can significantly affect the efficiency of the algorithm even though the user has no clear idea of how to select them. Finally, the speed of convergence, which is one of the most important criteria in evaluating the capability of an algorithm, is not desirable

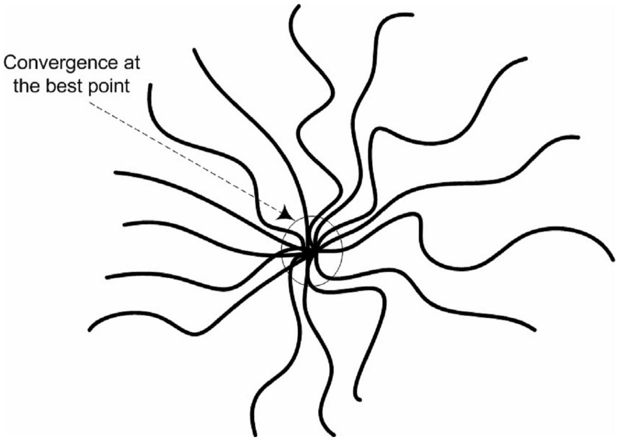


Figure 3.2: ideal convergence to preserve the diversity with the same number of agents

in previous multi-individual methods compared with the classical gradient-based optimization methods. Spiral Bacterial Foraging Optimization (SBFO) is intended to cope with these three major problems.

Basic Concept The main optimization problem can be defined as:

$$\min_{x \in X} h(x) \tag{3.1}$$

in which $X \subseteq \mathbb{R}^n$, $h : \mathbb{R}^n \to \mathbb{R}$ is a general differentiable nonlinear function.

Compared to existing algorithms that have relatively complicated algorithms for individual and collective behaviors, SBFO is relatively simple. In SBFO, each agent moves along the steepest descent direction (obviously, other alternative gradient-based methods such as conjugate gradient, DFP, BFGS, etc., can be employed as well) in which two different objectives are minimized. The first one is the main objective function and the second one is the distance between each agent and the leader. This can be written as [1],

$$f = h + \lambda g \tag{3.2}$$

where h is the main objective function and $g : \mathbb{R}^n \to \mathbb{R}$ is defined as,

$$g = \left\| x^i - x^{ml} \right\|^2 \tag{3.3}$$

where $x^i \in X$ and $x^{ml} \in X$ are corresponding vectors to the i^{th} agent and the main leader respectively. The leader is the best current agent. $\|\|$ is the distance operator in Euclidian space. Here, $\lambda \in \mathbb{R}^+$ is a positive multiplier.

To explain the algorithm, it should be mentioned that, based on this algorithm, each agent tries to minimize the main objective function locally. In addition, each agent has to move towards the leader agent in order to minimize the second term in Equation (3.2), which is the key to escape from local optimum points (global search). During this process, if any agent can find a better solution, the central point (leader) switches to this agent.

This process continues by converging to the best seen solution by any agent. The main tunable parameters in this algorithm are the number of agents and the multiplier λ. Obviously, the higher the number of agents, the higher the probability of finding the global optimum point, but the higher, too, the computational cost. On the other hand, the higher value of the multiplier λ, the higher speed of convergence but less chance to explore the search space.

Like most other multi-individual algorithms [8], SBFO has the advantage of decentralizability. In other words, the corresponding process of each agent can be performed independently on a separate processor and the only information that is required to be exchanged between agents is the position of the leader. Movement along the gradient of the objective function in SBFO is analogous to the movement of Dictyostelium Cells along the gradient of a cAMP protein emitted by other pacemaker Dictyostelium cells [26]- [27].

To eliminate the premature convergence issue and improve the global search capability, a new operator is employed [1] which is inspired by the spiral motion of Dictyostelium cells. Assume that, as the agents are attracted toward the current main leader, a random motion perpendicular to the connecting line between the current agent and the main leader is imposed on the agents. Figure (3.3) shows the concept.

Therefore Equation(3.3) should be modified as:

$$g = \left\| x^i - x^{vl,i} \right\|^2 \tag{3.4}$$

in which $x^{vl,i} \in \mathbb{R}^n$ represents the virtual (pseudo) leader solution in i^{th} iteration, shown in Figure (3.3)

To select a proper normal arm, it is assumed that each agent has a random rotational velocity around the central point in addition to the radial motion. For each agent, this rotational velocity is selected randomly so that the heading of each agent, with respect to the radial direction, will be changing stochastically. The randomness offers some benefits for the algorithm. First of all, as the heading of each agent changes randomly at each

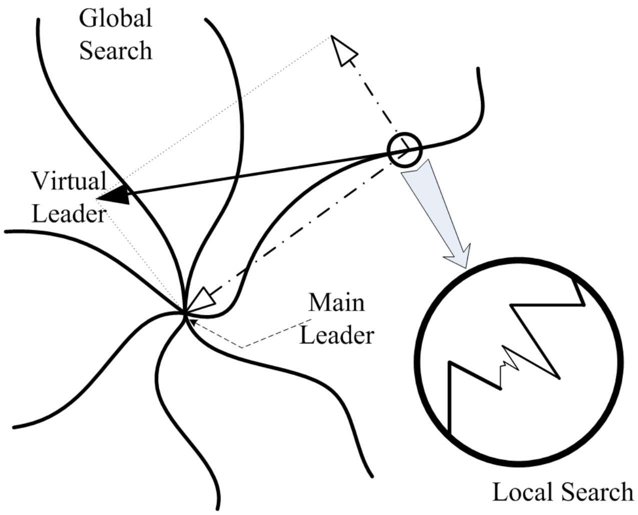


Figure 3.3: A random spiral motion to improve diversity and the global search capability

time-step, none of the agents remain on the same path, so premature convergence does not occur. In addition, this random operator increases the capability of global search because it helps the agents to explore more portions of the feasible space. One of the more interesting characteristics of this operator is that, the farther the agents are from the central point, the larger the tangential motion, which essentially helps them to search a wider region, whereas agents closer to the central agent have a smaller tangential motion and therefore will not deviate from the central agent significantly. This enables the convergence of the algorithm as once the agent arrives in the central point, the tangential motion will also be zero.

SBFO Algorithm Definition As mentioned, the main operator in SBFO can be any of the gradient-based methods, such as the steepest descent method. The algorithm is now described in details. In the following algorithm N_a is the number of agents and the maximum number of iterations in the main loop and line search loop, are called N_{it} and N_γ, respectively.

As stated in step 3 of "Main Part" in Algorithm(1), at each time step, a normal vector for each vector has to be found which is perpendicular to the connecting vector between the current and leading agent. In 3-dimensional problems, one can easily find this normal vector by using a cross-product operator; however, for higher dimensions, that cross-product is

Algorithm 1 ***SBFO Algorithm***

```
Initialization
Choose $N_a$, $N_{it}$ and $N_\gamma$.
Initial guesses for agents $x_1^i$ $(i = 1...N_a)$
Function evaluations for initial guesses $f(x_1^i)$ $(i = 1...N_a)$
Set $\varepsilon_1, \varepsilon_2, \varepsilon_3 > 0$ (tolerances for stopping criteria)
Set $\lambda_1^i = a \in \mathbb{R}^+$ $(i = 1...N_a)$
Set $\|\nabla f(x_1^i)\| = \nabla f_0 > 0$ and $\Delta x_1^i = \Delta x_0 > 0$ $\forall i \in \{1, 2, ..., N_a\}$
    Main Part
    While $(t \leq N_{it})$ and $(\left\|\nabla f(x_{t+1}^i)\right\| \geq \varepsilon_3)$ and $(\|\Delta x_t^i\| \geq \varepsilon_2)$ $\forall i \in \{1, 2, ..., N_a\}$
        Find the main leader $x_t^{ml} = \underset{x_t^i}{\operatorname{argmin}}\, h\left(x_t^i\right)$
      For $(i = 1...N_a)$
        $\Delta f(x_t^i) = f(x_{t+1}^i) - f(x_t^i)$, $\Delta x_t^i = x_{t+1}^i - x_t^i$
        Find $d(x_t^i, x_t^{ml}) = x_t^i - x_t^{ml}$ $(\forall i \neq ml)$ the vector between each agent and leader.
        Find $\rho(x_t^i, x_t^{ml})$ a normal vector to $d(x_t^i)$
        $g(x_t^i) = \left\|d(x_t^i, x_t^{ml})\right\|$distance between current agent and main leader
        $f(x_t^i) = h(x_t^i) + \lambda_t^i g(x_t^i)$ objective function including local and global costs
        $s_t^i = -\nabla f(x_t^i)$ (descent direction)
        $w_t^i = -\eta \|\nabla g(x_t^i)\| \rho(x_t^i, x_t^{ml})$ (random perturbation, $\eta \in$[0,1) )
                While $|\Delta f(x_t^i)| \geq \varepsilon_1$ and $k_t < N_\gamma$ (line search)
                Update $\gamma_t^i$, refer to [43], [28] and [29] and see algorithm (3.1.2)
                $k_t = k_t + 1$
                End While(line search)
        $x_{t+1}^i = x_t^i + \gamma_t^i(s_t^i + w_t^i)$
        Update $\lambda_t^i$ if necessary (See Lemma(3))
            End For $(i)$
    $t = t + 1$
End While (Main Part)
```

not defined, and so the following procedure has been proposed. Assume that there is an arbitrary N-dimensional vector X. Next, two coordinates are selected randomly, such as p and q^{th} entities in Figure (3.4). To create a vector normal to it, one should choose a vector with zero elements except for the p^{th} and q^{th} elements that are replaced by $X(q)$ and $-X(p)$, respectively. Obviously, the dot product of X and Y is zero; hence, they are perpendicular to each other. This idea is shown in Figure (3.4). This vector has to be normalized and then can be used in step 3 of global search in Algorithm(1).

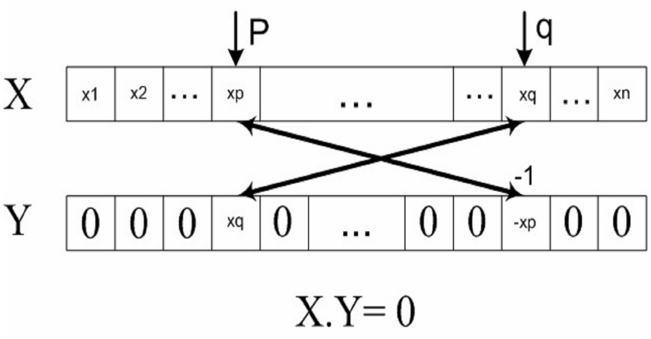


Figure 3.4: A way to find a random vector normal to any arbitrary vector in N-dimensional space

Tunable parameters

An interesting characteristic of SBFO is that it has no tunable parameters other than for the number of agents, two random distributions (random spiral velocity and random selection of normal arm's direction), and the multiplier λ. Evidently, the ability to find the global optimum can be improved with a larger number of agents, although it proportionally (not exponentially) increases the computational costs. Therefore, proper selection of it depends on the complexity of the problem and computational resources. Random distributions have not been observed to have a major influence on the performance of the algorithm. This robustness reduced the main tunable parameter only to the number of agents and the multiplier, λ.

3.1.2 Convergence analysis

Review on Convergence analysis of Stochastic Gradient Systems

To investigate the convergence of the algorithm one has to deal with the switching stochastic gradient system. There are some mathematical proofs for stochastic gradient algorithms but their conditions on random perturbation and some other parts are not exactly compatible with the SBFO structure. Therefore, some modifications are required in order to match them with the SBFO. A survey by Y. Ermoliev [31] reviewed the development of stochastic gradient methods for unconstrained optimization. In [32] Ermoliev and Nurminski have investigated stochastic quasi-gradient (SQG) methods for stochastic minimax problems. These methods can help to deal with optimization problems without having access to the exact value of objective function and constraints. Although there is no convergence analysis in these two surveys, different structures in stochastic gradient systems have been described in detail.

B.T. Poljak [33] has proposed some techniques to examine local convergence of stochastic optimization processes and has been able to prove the convergence of some differentiable optimization algorithms with strong assumptions on perturbation, such as independence of disturbances in their function evaluations and derivatives. The investigated structures are different from the SBFO structure that is concerned with in the current work. These methods are generally regarded as optimization methods in the presence of random noise.

An asynchronous distributed computational model is presented in [34] by J. N. Tsitsiklis et al. to investigate the convergence of a large class of either deterministic or stochastic gradient-like algorithms. It has been shown that under some conditions, these distributed algorithms retain the same convergence properties of centralized versions. This approach could help in investigating multi-agent algorithms such as SBFO, particularly in path planning applications.

Nakonechnyi in [35] surveys the literature on convergence theorems for stochastic gradient processes which employ Lyapunov's second approach. Covering some major theorems and their proofs in addition to stability makes this survey extremely helpful for convergence analysis, however finding a proper Lyapunov function is still an open question.

A sample-path based stochastic gradient-descent algorithm employed to optimize expected value performance has been investigated in [36] in terms of convergence. Two types of convergence are proposed: first, when the "expected value" function is continuously differentiable and second, when the performance functions are convex but not differentiable. The proofs are based on the uniform law of large numbers.

J. C. Fort and G. Pages have proved the global Kushner-Clark theorem for stochastic algorithms [37]. Furthermore, the smoothness assumption they made in the first part of the paper is relaxed by using a new approach based on a path-dependent Lyapunov functional.

A class of descent algorithms for unconstrained optimization with an Armijo-type step size rule is considered by [38], in which the performance function is calculated inaccurately. The most interesting aspect of this work is that it has neither limitation on relative size of perturbation nor is it required to tend to zero. This property helps the algorithm cope with difficult situations one may encounter in real problems. By doing so, the authors have suggested a new Armijo-type rule for computing the step size to ensure having an acceptable approximate solution.

In [39], the convergence analysis of a generalized sub-gradient algorithm when the perturbation exists. According to this article, the previous results on convergence and stability properties of gradient and sub-gradient methods become extended and unified.

Asymptotic stability analysis of the origin for an ordinary differential equation (ODE can imply the stability of the stochastic approximation algorithm) can also be employed [40]. This approach can be advantageous in some ways such as simplicity and convergence without prior assumption of stability.

Asymptotic behavior, distributed and multi-scale algorithms and application of stochastic approximation are all studied in [41], which can be useful for both modeling of stochastic phenomena and also optimization algorithms.

D. P. Bertsekas and J. N. Tsitsiklis in [42] have sharpened the proof of convergence for the classic problem of gradient methods with errors; in so doing, they have considered both deterministic and stochastic errors. Based on the assumptions, the step size in a gradient procedure tends to zero, and also descent direction and error satisfy some standard conditions.

It is useful to be familiar with line search as a traditional method in solving unconstrained optimization problems. The proof of convergence for gradient algorithm (deterministic) based on line search approach has recently attracted a great deal of attention. A work by Z. J. Shi [43] is a good reference for recent development in this regard. A combination of these methods and the stochastic gradient approaches such as [42] and [44] is employed in current thesis for convergence analysis.

The global convergence of two methods called gradient-type method and hybrid projection method with perturbations has been proposed in [44]. For both algorithms, non-monotone line search technique is used. Again, this paper assumes the step size as a decreasing value (in which the step size tends to zero when the algorithm proceeds) in the update process. The general proposition suggested in this work has no such limitations over step size to be compatible with SBFO structure.

Preliminary Definitions and Propositions

The main problem is to minimize a general nonlinear function $h(x)$ where $x \in \mathbb{R}^n$and $\mathbb{R}^n$ represents the n-dimensional Euclidean space and $f(x) : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable function, such that Lipschitz condition holds:

$$\|\nabla f(x) - \nabla f(\overline{x})\| \leq L \|x - \overline{x}\|, \forall x, \overline{x} \in \mathbb{R}^n, \tag{3.5}$$

where L is a constant value.

Definition 1. *A sequence γ_t is called "square summable but not summable", if it obeys the following equations :*

$$\sum_{t=0}^{\infty} \gamma_t = \infty \tag{3.6}$$

$$\sum_{t=0}^{\infty} \gamma_t^2 < \infty \tag{3.7}$$

Let $\{x_t\} \in \mathbb{R}^n$ be a sequence generated by the following recursive equation [42]:

$$x_{t+1} = x_t + \gamma_t(s_t + w_t), \tag{3.8}$$

in which $\gamma_t \in \mathbb{R}$ is a deterministic positive step size, $s_t \in \mathbb{R}^n$ is a descent direction, and $w_t \in \mathbb{R}^n$ is noise term. This corresponds to the solutions generated by SBFO, as shown in Algorithm (1). In addition, the following assumptions are made:

Assumption I) There exist two bounded positive scalars $c_1$and $c_2 \in \mathbb{R}^+$ such that

$$c_1 \|\nabla f(x_t)\|^2 \leq -\nabla f(x_t)^T s_t, \forall t \tag{3.9}$$

$$\|s_t\| \leq c_2 \|\nabla f(x_t)\|, \forall t \tag{3.10}$$

Assumption II) w_t is a deterministic error that has the following characteristic for some $p, q \in \mathbb{R}^+$ and all t,

$$\|w_t\| \leq \gamma_t(q + p \|\nabla f(x_t)\|). \tag{3.11}$$

Assumption III) The sequence γ_t is "square summable but not summable".

The last assumption leads to slow convergence of the process that is necessary to let the algorithm explore search space more precisely.

Proposition 1. *Using the classic descent method presented in Equation(3.8) with Assumptions (I) to (III), either $f(x_t) \to -\infty$ or else $f(x_t)$ converges to finite value. Moreover in the latter case, $\lim\limits_{t\to\infty} \nabla f(x_t) = 0$ and every limit point of $\{x_t\}$is a stationary point of f.*

The outline of proof can be found in [42] by D.P. Bertsekas and J.N. Tsitsiklis or [44] by Mei-xia Li, Chang-yu Wang.

In SBFO, the gradient based algorithms have been employed for each agent and the step sizes are selected based on a line search algorithm (exact or approximate), so here another proposition is required that replaces the assumptions, made in Proposition (1) with conditions, required for this line search method.

Convergence of SBFO

In order to be compatible with the SBFO structure, the Assumptions (I) to (III) should be modified as follows.

Let $\{x_t\}$ be a sequence generated by the recursive Equation(3.8) and assume the main descent direction s_t and perturbation term w_t satisfy the following conditions:

$$\exists c_1 > 0, \|s_t\| \leq c_1 \|\nabla f(x_t)\|, \forall t \tag{3.12}$$

$$\exists c_2 \geq 1, \langle \nabla f(x_t), s_t \rangle \leq -c_2 \|\nabla f(x_t)\|^2, \forall t \tag{3.13}$$

$$\exists p < 1, \|w_t\| \leq p \|\nabla f(x_t)\|, \forall t \tag{3.14}$$

where $c_1 \in (0, \infty)$ and $c_2 \in [1, \infty)$ and let $I = \{t \in \mathbb{Z}^+ | \langle \nabla f(x_t), s_t + w_t \rangle \geq 0\}$ and $J = \mathbb{Z}^+ \backslash I$.

Rather than the constraints in Equations (3.6) and (3.7), one can assume that the sequence γ_t is generated by any line search algorithm [43]. For the sake of simplicity the "Armijo rule" [43] is chosen for this thesis and presented in the Algorithm (3.1.2). It should be noted that this algorithm is a counterpart for the line search algorithm (inner loop for each agent) in Algorithm (1) in which the “Armijo rule” for line search procedure is employed. In this algorithm, the maximum number of iterations in main loop and line search loop, are called N_{it} and N_γ, respectively and $\varepsilon_1 > 0$ is the tolerance for stopping criteria.

Algorithm 2 Armijo Algorithm for Local Search

```
• Define constants $\mu, \nu \in (0, 1), x_0 \in \mathbb{R}^n$ ,

• Set $N_{it}$ and $N_\gamma$, and $\varepsilon_1 > 0$

• Let $t = 0$, and $r_t = s_t + w_t$ where $s_t$ and $w_t$ has been defined previously

• While $\|\nabla f(x_t)\| \geq \varepsilon_1$, and $t \leq N_{it}$
      Suppose $\gamma = a$, $a \in \mathbb{R}^+$,
      while $f(x_t + \gamma r_t) > f(x_t) + \mu\gamma \langle \nabla f(x_t), r_t \rangle$ and $k \leq N_\gamma$
          $\gamma = \nu\gamma$
          $k = k + 1$
      end while
      $\gamma_t = \gamma$
      $x_{t+1} = x_t + \gamma_t r_t$
      $t = t + 1$
  End While
```

The gradient based scheme with perturbation is represented as follows:

In what follows, the main goal is to show that the convergence of the algorithm is guaranteed; to do so, some explanations and lemmas are required. The main difference between Propositions (1) and (2), is attributed to the characteristic of perturbation in SBFO. As a result, the constraints (3.6, 3.7 and 3.11) are limited only to inequality (3.14). In addition, it is assumed that $c_2 \geq 1$ and $p < 1$ which is different from what is explained in Proposition (1) to make the parameters compatible with the requirement of SBFO. Furthermore, employing the line search algorithm, a gradual convergence is obtained in order to have an efficient method in discovering the search space. This is an alternative to the "square summable but not summable" property of Equations (3.6) and (3.7).

Lemma 1. *Under constraint (3.14), it is always true that* $\langle \nabla f(x_t), s_t + w_t \rangle < 0$ *or simply* $I = \varnothing$

Proof. Let's proceed by contradiction:

Suppose $\exists t \in I$ such that $\langle \nabla f(x_t), s_t + w_t \rangle \geq 0$. Therefore

$$- \langle \nabla f(x_t), s_t \rangle \leq \langle \nabla f(x_t), w_t \rangle \tag{3.15}$$

from (3.13) and (3.14), and Cauchy-Schwarz inequality:

$$\begin{aligned} c_2 \left\| \nabla f(x_t) \right\|^2 &\leq - \langle \nabla f(x_t), s_t \rangle \\ &\leq \langle \nabla f(x_t), w_t \rangle \\ &\leq \left\| \nabla f(x_t) \right\| \left\| w_t \right\| \\ &\leq p \left\| \nabla f(x_t) \right\|^2 \end{aligned} \tag{3.16}$$

From (3.16) one can conclude that $c_2 \leq p$ which is a contradiction with our initial assumption that $c2 \geq 1$ and $p < 1$ which implies $p < c_2$. Hence, it always holds that $\langle \nabla f(x_t), s_t + w_t \rangle < 0$ or simply $I = \varnothing$. □

Lemma 2. *Let* $\{x_t\}$ *and* $\{f(x_t)\}$ *be iteration sequences generated by Algorithm 4.4., then the sequence* $\{f(x_t)\}$ *is monotone and non-increasing.*

Proof. Looking at step 3, in Algorithm (3.1.2) it is known that $f(x_t + \gamma r_t) \leq f(x_t) + \mu\gamma \langle \nabla f(x_t), r_t \rangle$, where $r_t = s_t + w_t$. From Lemma (1), $\langle \nabla f(x_t), r_t \rangle < 0$ so $f(x_{t+1}) \leq f(x_t)$ which leads to the conclusion that $\{f(x_t)\}$ is monotone and non-increasing where $x_{t+1} = x_t + \gamma r_t$. □

Proposition 2. *The modified descent method presented by assumptions (3.12-3.14) along with the line search method in the Algorithm (3.1.2) result in either $f(x_t) \to -\infty$ or $f(x_t)$ converges to a finite value. Moreover in the latter case, if $\{\nabla f(x_t)\}$ is uniformly continuous on an open convex set $D \subseteq \mathbb{R}^n$ that contains $\{x_t\}$ then:*

- $\lim\limits_{t\to\infty} \nabla f(x_t) = 0$
- *Every limit point of $\{x_t\}$is a stationary point of f*
- *$\{x_t\}$ is convergent*

Proof. Instead of a direct proof, assume that, contrary to the statement in Proposition (2), $\exists \epsilon_0 > 0$ such that $\|\nabla f(x_t)\| > \epsilon_0, \forall t$. By using (3.13), (3.14) and current assumption:

$$\begin{aligned} -\langle \nabla f(x_t), r_t \rangle &= -\langle \nabla f(x_t), s_t + w_t \rangle \\ &\geq c_2 \|\nabla f(x_t)\|^2 - \|\nabla f(x_t)\| \, p \, \|\nabla f(x_t)\| \\ &= (c_2 - p) \|\nabla f(x_t)\|^2 \end{aligned} \tag{3.17}$$

Also it is known that:

$$\begin{aligned} \|r_t\| &= \|s_t + w_t\| \\ &\leq c_1 \|\nabla f(x_t)\| + p \|\nabla f(x_t)\| \\ &= (c_1 + p) \|\nabla f(x_t)\| \end{aligned} \tag{3.18}$$

From the definition of Algorithm (3.1.2):

$$f(x_{t+1}) \leq f(x_t) + \mu\gamma \langle \nabla f(x_t), r_t \rangle \tag{3.19}$$

Equation(3.19) together with Equation(3.17) imply:

$$\begin{aligned} f(x_t) - f(x_{t+1}) &\geq -\mu\gamma \langle \nabla f(x_t), r_t \rangle \\ &\geq \mu\gamma((c_2 - p) \|\nabla f(x_t)\|^2) \\ &\geq 0 \end{aligned} \tag{3.20}$$

The left hand side of the first row in Equation (3.20) has a limit of zero because as $f(x_t)$ is monotone and non-increasing, therefore $f(x_t)$ should tend to $-\infty$ or to a finite value.

If one assumes that $f(x_t)$ is bounded over the domain the final limit is a finite value then the limit of left hand side (3.20) is zero. By taking the limit of all expressions of inequality (3.20)

$$\lim_{t\to\infty} \gamma_t \left\| \nabla f(x_t) \right\|^2 = 0 \tag{3.21}$$

as it is assumed that $\forall t, \|\nabla f(x_t)\| > \epsilon_0$, it leads to

$$\lim_{t\to\infty} \gamma_t = 0 \tag{3.22}$$

Considering the way that the algorithm defines γ_t , one can say that γ_t is the smallest positive value that satisfies the inequality in definition of algorithm (Armijo rule [43]); so if one defines $\psi_t = \frac{\gamma_t}{\nu}$ the following inequality holds:

$$f(x_t + \psi_t r_t) \geq f(x_t) + \mu\psi_t \langle \nabla f(x_t), r_t \rangle \tag{3.23}$$

then,

$$f(x_t) - f(x_t + \psi_t r_t) \geq -\mu\psi_t \langle \nabla f(x_t), r_t \rangle \tag{3.24}$$

Applying the Median Value Theorem,

$$\begin{aligned} &\langle \nabla f(x_t) - \nabla f(x_t + \theta_t \psi_t r_t), r_t \rangle \\ &\qquad < (1-\mu) \langle \nabla f(x_t), r_t \rangle \end{aligned} \tag{3.25}$$

where $\theta_t \in (0,1)$. By re-arranging the inequality (3.25),

$$\begin{aligned} 1-\mu &< \frac{\langle \nabla f(x_t) - \nabla f(x_t + \theta_t \psi_t r_t), r_t \rangle}{-\langle \nabla f(x_t), r_t \rangle} \\ &\leq \frac{\left\| \nabla f(x_t) - \nabla f(x_t + \theta_t \psi_t r_t) \right\| \left\| r_t \right\|}{-\langle \nabla f(x_t), r_t \rangle} \\ &\leq \frac{\left\| \nabla f(x_t) - \nabla f(x_t + \theta_t \psi_t r_t) \right\| (c_1 + p) \left\| \nabla f(x_t) \right\|}{(c_2 - p) \left\| \nabla f(x_t) \right\|^2} \end{aligned} \tag{3.26}$$

But it is known $\lim_{t\to\infty} \|\psi_t r_t\| = \lim_{t\to\infty} \frac{\gamma_t}{\nu} \|r_t\| = 0$, so the last expressions of the above inequality tends to zero when $t \to \infty$.

Therefore, $1-\mu < 0 \Longrightarrow \mu > 1$ which is a contradiction with our initial assumption for μ and proves that $\lim\limits_{t\to\infty} \nabla f(x_t) = 0$. Now, re-arrange Equation (3.8), and use Equations (3.12), (3.13) and (3.18):

$$\|x_{t+1} - x_t\| = \gamma_t \|r_t\| \leq a(c_1 + p) \|\nabla f(x_t)\| \tag{3.27}$$

By taking limit of both side when $t \to \infty$ the convergence of $\{x_t\}$ is guaranteed.

$$\lim_{t\to\infty} \|x_{t+1} - x_t\| = 0 \tag{3.28}$$

□

Consequently by using Proposition (2), one can deal with convergence of SBFO. Two different conditions have been investigated here. Initially, it is assumed that an agent of SBFO has found the best possible solution; hence, no switching to another agent occurs. Then a more general circumstance will be considered in which the process starts from initial random locations for agents and start exploring the feasible space. Some formal definitions and lemma are needed for the main theorems on convergence of the SBFO algorithm.

Definition 2. ***Best Possible Solution (BPS):*** *Given the set of sequences* $\{x_t^i\}, i \in 1 \ldots N_a$. $x^\star$ *is the Best Possible Solution of the optimization algorithm, if* $x^\star = \lim_{t\to\infty} x_t^i$, $\forall i \in 1 \ldots N_a$, *such that* $f(x^\star) \leq f(x_t^i)$, $\forall i$ *and* t.

Definition 3. ***Permanent Leader (PL)***: *The* i^{th} *agent becomes the Permanent Leader at time* t_0 *if*

$$\forall t \geq t_0,\ \ f(x_t^i) \leq f(x_t^j),\ \forall j \neq i, j \in \{1 \ldots N_a\} \tag{3.29}$$

where the t_0 *is the time at which, SBFO has found this Leader.*

Obviously, the permanent leader (PL) is not the best possible solution (BPS) as it is still moving toward the BPS in a locally convex neighborhood of the BPS. The convexity of this neighborhood will be discussed in Lemma (4) in detail.

It is worth mentioning that, if the SBFO algorithm finds its permanent leader there is no more switching between leaders. It usually happens at the last stage of optimization process in which permanent leader is located at a locally convex neighborhood of best possible solution (BPS). From this point up to Lemma (6), it is assumed that the permanent leader has been found.

Looking at the function,f, represented in equation (3.2) one can see that in SBFO, the objective function contains two major parts: the first is the main function h to be optimized, and the second, is the distance between the current agent and the virtual leader, g. The function g can be re-written as follows (Pythagorean Theorem in N_a dimensional space) [see Figure(3.3)]:

$$g(x_t^i) = \left\| x_t^i - x_t^{vl,i} \right\|^2 = \left\| x_t^i - x_t^{pl} \right\|^2 + \left\| x_t^{vl,i} - x_t^{pl} \right\|^2 \tag{3.30}$$

x_t^{pl} is the permanent leader at time t and it is known that $x_t^{vl,i} - x_t^{pl}$ is a perpendicular arm to the vector connecting x_t^i to $x_t^{pl}$and its length can be determined as:

$$\left\| x_t^{vl,i} - x_t^{pl} \right\| = \eta \left\| x_t^i - x_t^{pl} \right\| \tag{3.31}$$

In which $\eta \in [0, 1)$ is a random number drawn from a uniform distribution over $[0, 1)$.

Let's define the function q as:

$$q(x_t^i, x_t^{pl}) = \left\| x_t^i - x_t^{pl} \right\| \tag{3.32}$$

and also ρ the unit vector perpendicular to the vector connecting x_t^i to x_t^{pl}

$$\rho(x_t^i, x_t^{pl}) = \frac{x_t^{vl,i} - x_t^{pl}}{\left\| x_t^{vl,i} - x_t^{pl} \right\|} \tag{3.33}$$

To be compatible with Proposition (2), It is possible to rewrite s_t^i and w_t^i in terms of ∇h, ∇q and ρ.

$$s_t^i = -\nabla f(x_t^i) = -\nabla h(x_t^i) - \lambda \nabla q(x_t^i) \tag{3.34}$$

$$w_t^i = -\lambda \eta \left\| \nabla q(x_t^i) \right\| \rho(x_t^i, x_t^{pl}) \tag{3.35}$$

Lemma 3. *Given any x_t^i for which $||\nabla q(x_t^i, x_t^{pl})|| \neq 0$, there exists a $\lambda > 0$, such that $\lambda \left\| \nabla q(x_t^i, x_t^{pl}) \right\| < \|\nabla f(x_t^i)\|$*

Proof. From reverse triangle inequality $\forall\ V, U \in \mathbb{R}^n$:

$$|\ \|V\| - \|U\|\ | \leq \|V - U\| \tag{3.36}$$

By replacing U by $-U$:

$$|\ \|V\| - \|U\|\ | \leq \|V + U\| \tag{3.37}$$

Substituting $V = \nabla h$ and $U = \lambda \nabla q$ and by definition of ∇f

$$\begin{aligned} |\; \|\nabla h\| - \|\lambda \nabla q\| \;| &\leq \|\nabla h + \lambda \nabla q\| \\ &= \|\nabla f\| \end{aligned} \tag{3.38}$$

Therefore, if one finds λ in such a way that the following inequality (3.39) is satisfied, the inequality in Lemma (3) will also be satisfied.

$$\lambda \|\nabla q\| < |\; \|\nabla h\| - \lambda \|\nabla q\| \;| \tag{3.39}$$

Solving this inequality for λ leads to the following ranges for λ:

$$0 < \lambda < \frac{\|\nabla h\|}{\|\nabla q\|} - 1 \tag{3.40}$$

$$\lambda > \frac{\|\nabla h\|}{\|\nabla q\|} + 1 \tag{3.41}$$

Pictorial representation of Equation(3.34) when λ is selected from the ranges in (3.40) and (3.41) are shown in Figures (3.5) and (3.6) respectively. As can be seen, the critical situation happens when $\vec{\nabla} h$ and $\vec{\nabla} q$ are in opposite directions and by choosing λ based on inequalities (3.40) or (3.41) the required condition in Lemma (3), $\lambda \left\| \nabla q(x_t^i, x_t^{pl}) \right\| < \|\nabla f(x_t)\|$, will be met.

□

The only problem with inequality (3.40), is that when $\|\nabla h\| \to 0$, λ approaches -1 which is not acceptable as λ has to be a positive value. So, a better choice for λ is the range in inequality (3.41). Theorem (2) demonstrates that the structure defined in Equations (3.34) and (3.35) when the λ is chosen within the range introduced by (3.41) guarantees the convergence of SBFO.

Theorem 2. *Assume that, the Permanent Leader has been found by SBFO algorithm, and the multiplier, λ, in Equation (3.2) is chosen such that Inequality (3.41) satisfied, then the sequence $\{x_t^i\}$, generated by SBFO algorithm converges to a local optimum point, such that $\lim_{t\to\infty} \nabla f(x_t^i) = 0$ for all $i = 1 \ldots N_a$ agents.*

Proof. It can be shown that s_t and w_t in Equations (3.34) and (3.35) satisfy the conditions in Equations (3.12-3.14), hence Proposition (2) applies. To show that, one can select $c_1 = c_2 = 1$.

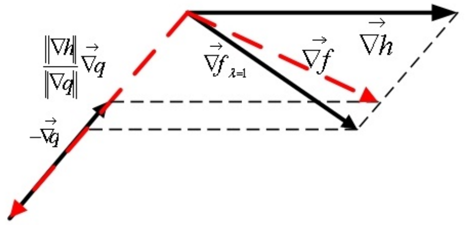

$\frac{\|\nabla h\|}{\|\nabla q\|}\vec{\nabla} q$
$\vec{\nabla} f_{\lambda=1}$
$\vec{\nabla} f$
$\vec{\nabla} h$
$-\vec{\nabla} q$

Figure 3.5: $0 < \lambda < \frac{\|\nabla h\|}{\|\nabla q\|} - 1$

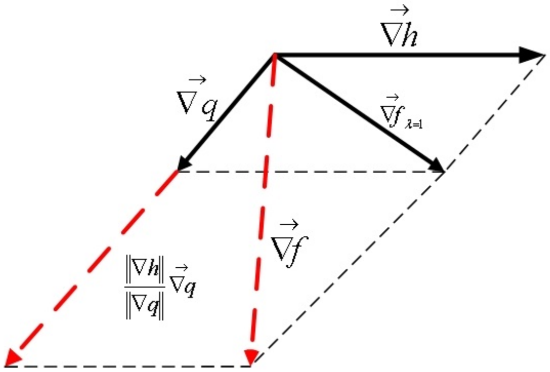

$\vec{\nabla} h$
$\vec{\nabla} q$
$\vec{\nabla} f_{\lambda=1}$
$\vec{\nabla} f$
$\frac{\|\nabla h\|}{\|\nabla q\|}\vec{\nabla} q$

Figure 3.6: $\lambda > \frac{\|\nabla h\|}{\|\nabla q\|} + 1$

Inequality (3.12) holds, because:

$$c_1 = 1 \rightarrow \left\|s_t^i\right\| \leq c_1 \left\|\nabla f(x_t^i)\right\|, \forall t \tag{3.42}$$

and also for Inequality(3.13):

$$c_2 = 1 \rightarrow \left\langle \nabla f(x_t^i), s_t^i \right\rangle \leq -c_2 \left\|\nabla f(x_t^i)\right\|^2, \forall t \tag{3.43}$$

Finally, as $\eta < 1$, if one selects $p = \eta$, then the Inequality (3.14) always holds, which can be seen from Equations (3.34) and (3.35) and also this point that always $\|\rho\| = 1$, one can conclude that:

$$
\begin{aligned}
p = \eta \rightarrow \left\|w_t^i\right\| &= \left\|-\lambda\eta \left\|\nabla q(x_t^i)\right\| \rho(x_t^i, x_t^{pl})\right\| \\
&= \lambda\eta \left\|\nabla q(x_t^i, x_t^{pl})\right\| \\
&= p\lambda \left\|\nabla q(x_t^i, x_t^{pl})\right\| \\
&\leq p \left\|\nabla f(x_t^i)\right\|, \forall t
\end{aligned}
\tag{3.44}
$$

The last inequality in Equation (3.44) is valid as, λ is chosen within the range introduced by (3.41) in Lemma(3).

Finally, if one defines a recursive update algorithm for the set of sequences $\{x_t^i\}$, as in Equation (3.8), according to Proposition (2), the convergence is guaranteed and $\{\nabla f(x_t^i)\} \rightarrow 0$, $\forall i = 1 \ldots N_a$. □

Based on the Theorem (2), it has been noted that each agent converges to a local optimum point. Now one should pay attention that having the condition of $\{\nabla f(x_t^i)\} \rightarrow 0$ does not necessarily mean that all agents have converged to a unique local optimum point of the main function that is expected to be optimized. This happens because at a local optimum point of function h, $\lim\limits_{t\rightarrow\infty} \nabla h(x_t^i)$ and $\lim\limits_{t\rightarrow\infty} \nabla q(x_t^i)$ $\forall i = 1 \ldots N_a$, should be zero simultaneously but $\lim\limits_{t\rightarrow\infty} \nabla f(x_t^i) = 0$ may be caused by the condition: $\lim\limits_{t\rightarrow\infty} \nabla h(x_t^i) = -\lambda \lim\limits_{t\rightarrow\infty} \nabla q(x_t^i)$ for some $i = 1 \ldots N_a$. In such a circumstance, the limit point of the optimization process is not a local optimum of the main function h. Besides, the agents will not necessary converge to a unique point. This situation is shown in Figure (3.7).One way to deal with such a problem is to employ an augmented multiplier in which the multiplier is updated as follows:

$$
\lambda_t = \lambda_0 \chi^{\kappa(t)} \tag{3.45}
$$

In which $\chi > 1$, λ_0 are constant values and $\kappa(t)$ can be chosen simply as $\kappa(t) = \beta_0 t$ where β_0 is a positive constant. It is worth mentioning that rapidly increasing λ_t will not allow the agents to explore the search space efficiently and on the other hand small value of λ_t leads to slow convergence of the optimization process.

The following lemmas show how to cope with the aforementioned problem which leads to an un-converged solution. Lemma(4), shows that there always exists a λ_t by which the function f (including a non-convex function h and strictly convex function q) becomes locally convex. Lemma (5) explains how the convexity of f can lead to this conclusion that the local optimum point of f is also the local optimum point of h which is the main function to be optimized.

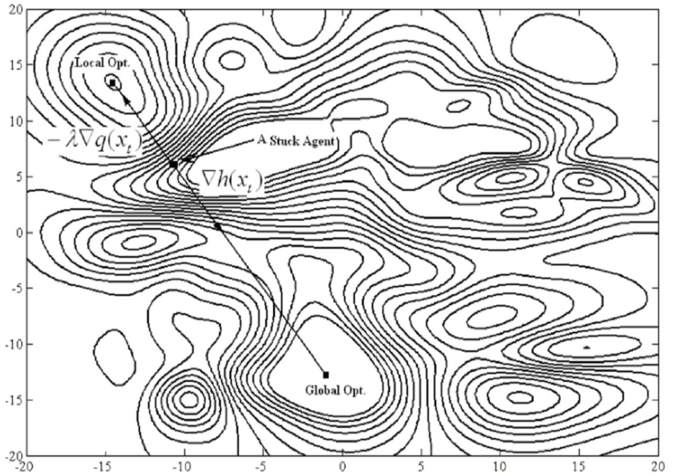


Figure 3.7: An agent is stuck because $\nabla h(x_t) = -\lambda \nabla q(x_t)$

For Lemma(4), one must prove that for a Lipschitz continuous function f defined in (3.2) in which q is strictly convex on $\mathbb{R}^n$, $\exists \lambda > 0$, such that $\forall \epsilon > 0$, f is convex at least in a ball with radius ϵ, about the BPS. Therefore, the gradient based algorithm employed for SBFO leads to a local optimum in which $\nabla h(x_t)$ and $\nabla q(x_t)$ are simultaneously zero and all agents converge to a unique point in such a condition the function f has only one solution where $\nabla q(x_t) = 0$ and $q(x_t) = 0$ which means all agents have converged to the central point. [see Lemma (5)]

Lemma 4. *Assume $f(x)$ defined in equation(3.2) is Lipschitz continuous, then $\forall \epsilon > 0, \exists \lambda$ such that $f(x)$ is locally convex in any neighborhood $D_\epsilon \subseteq D \subseteq \mathbb{R}^n$with radius ϵ, where D is the domain of f.*

Proof. By definition, a function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is convex if $D \subseteq \mathbb{R}^n$, the domain of f, is a convex set and if for all $x, y \in D$, and $\theta \in [0, 1]$, also [45]:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) \tag{3.46}$$

Strict convexity of $f(x)$ requires,

$$f(\theta x + (1-\theta)y) < \theta f(x) + (1-\theta) f(y) \tag{3.47}$$

Moreover, it has been assumed that f is Lipschitz continuous, therefore:

$$\begin{aligned} &\forall x, y \in \mathbb{R}^n, \exists L \in \mathbb{R} \text{ such that:} \\ &\quad \|f(x) - f(y)\| \leq L \|x - y\| \end{aligned} \tag{3.48}$$

To show the convexity of function f in a ball with radius ϵ, one should show that $\exists \lambda > 0$ by which the following inequality holds.

$$\begin{aligned} f(\theta x &+ (1-\theta)y) \\ &= h(\theta x + (1-\theta)y)) + \lambda q(\theta x + (1-\theta)y)) \\ &\leq (\theta h(x) + (1-\theta)h(y)) + \lambda(\theta q(x) + (1-\theta)q(y)) \end{aligned} \tag{3.49}$$

By rearranging (3.50):

$$\begin{aligned} h(\theta x &+ (1-\theta)y)) - (\theta h(x) + (1-\theta)h(y)) \\ &\leq \lambda(\theta q(x) + (1-\theta)q(y)) - q(\theta x + (1-\theta)y)) \\ &= \lambda\sigma \end{aligned} \tag{3.50}$$

As q is a strictly convex function (on $\mathbb{R}^n$), the right hand side of (3.50) is always positive or $\lambda\sigma > 0$ which means that instead of proving (3.50) one should show that for each $\sigma > 0$ and $\epsilon > 0$ there is $\lambda > 0$ such that:

$$h(\theta x + (1-\theta)y)) - (\theta h(x) + (1-\theta)h(y)) \leq \lambda\sigma \tag{3.51}$$

Knowing that h is Lipschitz continuous.

$$\begin{aligned} \|h(\theta x + (1-\theta)y) - (\theta h(x) + (1-\theta)h(y))\| &= \\ \|(h(\theta x + (1-\theta)y) - h(y)) - \theta(h(x) - h(y))\| &\leq \\ 2L_h\theta \|x - y\| &\leq 2L_h\theta\epsilon \end{aligned} \tag{3.52}$$

Where L_h is the constant in "Lipschitz continuity" condition for function h. So to find a proper $\lambda > 0$:

$$2L_h\theta\epsilon \leq \lambda\sigma \Longrightarrow \lambda \geq \frac{2L_h\theta\epsilon}{\sigma} \overset{Max\theta=1}{\longrightarrow} \lambda \geq \frac{2L_h\epsilon}{\sigma} \tag{3.53}$$

□

As λ_t in (3.45) is a monotonic increasing function, it is always possible to find a t_0 such that $\lambda_t \geq \frac{2L_h\epsilon}{\sigma}, \forall t > t_0$. Unfortunately the inequality (3.53) has no practical use to select a suitable λ_t during the optimization process and only helps to prove the existence of a λ_t.

Lemma 5. *There exists a sequence $\{\lambda_t^i\} \in \mathbb{R}^+$ which defines the function $f(x_t^i)$ in Theorem (2), such that if $\{\nabla f(x_t^i)\}$ converges to zero, then $\lim_{t\to\infty} \nabla h(x_t^i) = 0$ and $\lim_{t\to\infty} \nabla q(x_t^i, x_t^{pl}) = 0$ simultaneously and also all agents converge to a unique point $x^\star \in D \subseteq \mathbb{R}^n$.*

Proof. It is known that a convex function has a single point satisfying necessary conditions for optimality [45]. Select an ϵ such that the local optimum of h for the Permanent Leader lies within a ball of radius ϵ centered at the location of the Permanent Leader at iteration t.

Using the Lemma (4), there is a $\{\lambda_t^i\}$ by which the function f is locally convex in the above-mentioned ball. Obviously, the solution $\nabla q(x_t^i, x_t^{pl}) = 0$ and $\nabla h(x^\star) = 0, \forall i$ is a solution for $\nabla f(x) = 0$, where $x^\star$ is the corresponding local solution of h to the permanent leader. But one knows that f is locally convex, therefore this solution is the only solution of $\nabla f(x) = 0$ in that neighborhood. As mentioned, this unique solution will be achieved when $\nabla q(x^\star, x_t^{pl}) = 0$ and as q is a distance function and convex, this situation happens when the tracking agents and the main leader are coincident and lie on the local optimum point of h, $x^\star$, or $\left\| q(x_t^i, x_t^{pl}) \right\| = 0$. □

To accomplish the proof of convergence one should show that the SBFO algorithm will converge to the BPS, regardless of the assumption made in Theorem (2), where it is assumed that the PL has been found. To do so, first one should show that under some conditions there is only a finite number of switching between leaders before finding the PL. The next lemma is designed for this purpose.

Lemma 6. *Assume that there is minimum required improvement at each switching called $\zeta > 0$, then only a finite number of switching occurs between leaders.*

Proof. As stated, at each switching at least an improvement of $\zeta > 0$ is obtained, otherwise no switching occurs. Let's define $K = \{t_1, t_2, \ldots, t_j, \ldots, t_k\} \subset \mathbb{Z}^+$ where t_j is the time at which j^{th} switching occurs between leaders and also $\Delta f_s(t_j)$ is defined as:

$$\Delta f_s(t_j) = f(x_{t_j}^{ml}) - f(x_{t_{j+1}}^{ml}) \tag{3.54}$$

by assumption:

$$\Delta f_s(t_j) > \zeta \quad \forall t_j \tag{3.55}$$

Now if one sums over both side of equation (3.54):

$$\sum_{j=1}^{k-1} \Delta f_s(t_j) = \sum_{j=1}^{k-1} f(x_{t_j}^{ml}) - f(x_{t_{j+1}}^{ml}) \tag{3.56}$$

where k is total number of switching. from (3.55):

$$\sum_{j=1}^{k-1} \Delta f_s(t_j) > k\zeta \tag{3.57}$$

so from Equation(3.56),

$$\begin{aligned} &\sum_{j=1}^{k-1} f(x^{ml}_{t_j}) - f(x^{ml}_{t_{j+1}}) = f(x^{ml}_{t_1}) - f(x^{ml}_{t_k}) \\ &= \sum_{j=1}^{k-1} \Delta f_s(t_j) > k\zeta \end{aligned} \tag{3.58}$$

It is obvious that the first row of Equations (3.58) are bounded as it is assumed that f is bounded. Therefore, k, the total number of switches, can not be infinite, otherwise one arrives at a contradiction in Equation (3.58). □

Theorem 3. *Let $\{x^i_t\}$, $i = 1 \ldots N_a$, be a set of sequences generated by SBFO. All members of this set converge to a local optimum point, $x^\star \in D \subseteq \mathbb{R}^n$ at which $\nabla f(x^\star) = 0$ as long as there is a minimum required improvement, $\zeta > 0$, for switching between leaders.*

Proof. According to Lemma(6), in SBFO, only a finite number of switching occurs, if there is a minimum required improvement, $\zeta > 0$, for switching between leaders. Now let's limit our attention to the k^{th} stage of the optimization, afterthe last switch, in which $k < \infty$ is the total number of switches between leaders. According to Theorem(2) SBFO is globally convergent starting from any initial guesses, $x^i_{1,k} \in D \subseteq \mathbb{R}^n$ for k^{th} stage, or mathematically:

$$\forall x^i_{1,k} \in D \subseteq \mathbb{R}^n \Rightarrow \{x^i_{t,k}\} \rightarrow x^\star \tag{3.59}$$

in which $\{x^i_{t,k}\}$ is a set of sequences starting at the k^{th} stage, converging to $x^\star$.

Note that the initial guesses for the k^{th} stage are points that are produced by the previous stage, $(k-1)$, of the SBFO algorithm.

□

The main difference between this result and Theorem(2), is that switching between the central points is considered in the convergence process. In other words, as the process may find a better solution during the convergence toward the temporary central point, the leader may vary during the optimization. Having a finite number of switches is crucial for this theorem, because only the last stage needs to be considered for convergence. The only point is that, one should make sure that at each stage $x^i_{1,k} \in D \subseteq \mathbb{R}^n$.

3.1.3 Algorithm Evaluation

The performance of the SBFO has been evaluated for different multi-modal functions and has been found to be consistently capable of finding the global optimum point.

Benchmark Functions

Some standard benchmark functions such as: Shekel's Foxholes, Rosenbrock, Rastrigin, Griewank, Ackley functions, as well as two multi-modal random bumpy objective functions in two-dimensional space, have been used to evaluate the SBFO algorithm. As an example for the standard benchmarks, the Shekel's Foxholes is defined as:

$$h(x) = \frac{1}{500} + \sum_{i=1}^{25} \frac{1}{j + \sum\limits_{i=1}^{2} (x_i - a_{ij})^6} \tag{3.60}$$

where a_{ij} a are a set of constant parameters defined in reference [20]- [21]. Figure (3.8) shows Shekel's Foxholes function. For definitions of other standard benchmark functions the reader is referred to [20]- [21]. The other objective function used here is a function that consists of a set of one hundred Gaussian bell functions defined as follows:

$$h(x) = Ae^{-[a(x_1 - x_1^0)^2 + 2b(x_1 - x_1^0)(x_2 - x_2^0) + b(x_2 - x_2^0)^2]} \tag{3.61}$$

where $a = \frac{\cos^2(\theta)}{2\sigma_{x1}^2} + \frac{\sin^2(\theta)}{2\sigma_{x2}^2}, b = \frac{-\sin^2(2\theta)}{4\sigma_{x1}^2} + \frac{\sin^2(2\theta)}{4\sigma_{x2}^2}, c = \frac{\sin^2(\theta)}{2\sigma_{x1}^2} + \frac{\cos^2(\theta)}{2\sigma_{x2}^2}$ in which $A, \theta, \sigma_{x1}, \sigma_{x2}$ are some constant values selected randomly.

By adding 100 functions of these bell shape functions, a bumpy multi-modal function is available that is used to evaluate the SBFO. Figure (3.9) depicts a multimodal function produced by this method. The discovered paths of the function is shown in Figure (3.10). As there are many local optimum points, it is cluttered and not clear to follow the paths, therefore, another function of this type but with simpler shape is added that has only six Gaussian bell functions. This function is shown in Figure (3.11).

Results

Table I presents SBFO's performance compared with respect to other methods found in [20]- [21], for Shekel's Foxholes function. The performance in this table is defined as the average number of function evaluation required to find the global optimum point when the process is repeated 30 times. A brief explanation on how these algorithms work is presented in Introduction, Section (1.1). The point [-31.950, -31.950] is the global optimum of this problem.

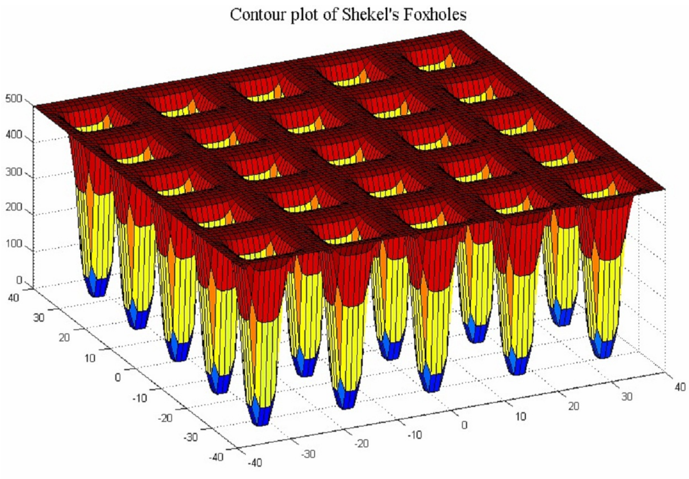


Figure 3.8: Shekel's Foxholes function

Table I: Comparison of Mean Number of Function Evaluations

DE/rand/1/bin	BFOA	BFOA-GA	CDE	SBFO
16472.85	27502.40	18291.86	12237.60	7384.20

It should be noted that the mean number of function evaluations was calculated for 30 runs by using 50 individuals or agents. The next example is related to the function shown in Figure (3.9). Again, the SBFO is successful in solving the problem. Figure (3.10) shows the path followed by some of the selected agents to find the global optimum. The point [-0.787, -13.173] is the global optimum of this example. Figure(3.12) is shown to see how the SBFO algorithm helps to escape from local optimum points to converge to the global solution even when the random term is off. The global optimum point is located at [0.0, 4.0] in this example.

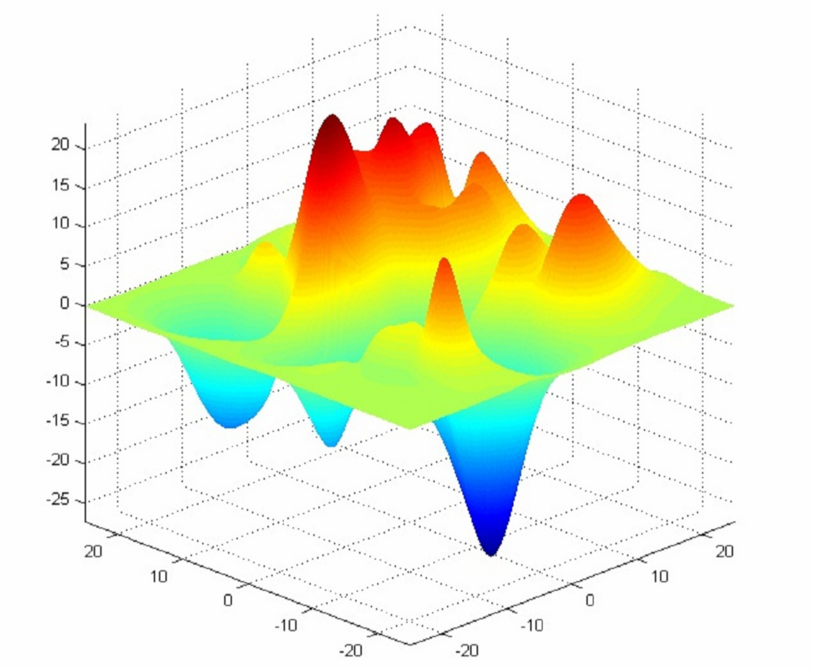


Figure 3.9: A bumpy surface produced by adding one hundred Gaussian bell functions

In another attempt to evaluate the capability of SBFO some benchmark functions are employed in order to compare SBFO with the classic global optimization algorithms like GA and SA. In this part only the number of successful convergence to the global solution out of 30 runs is compared.

Table II reports the results. The initial solutions for all of the algorithms are selected randomly within the same region of interest. In this experiment there is no limitation on total number of function evaluations unless the procedure stops at a local (or global) optimum point. As can be seen when there is no constraint on number of function evaluation GA has very good performance. Although SA behaves poorly in finding the optimum, in such cases that it does find the optimum, the number of function evaluations is extremely low compared to the other methods. Finally SBFO is comparable to GA in terms of reliability with acceptable computational costs.

Table II: Comparison between SA, GA and SBFO algorithms

Percentages of Convergences to global optimum

	Shekel's	Rosenbrock	Rastrigin	Griewank	Ackley	Bumpy O.
	Foxholes					100 Bells
SBFO	95	83	60	80	73	93
SA	10	13	3	3	17	100
GA	10	100	100	100	100	17

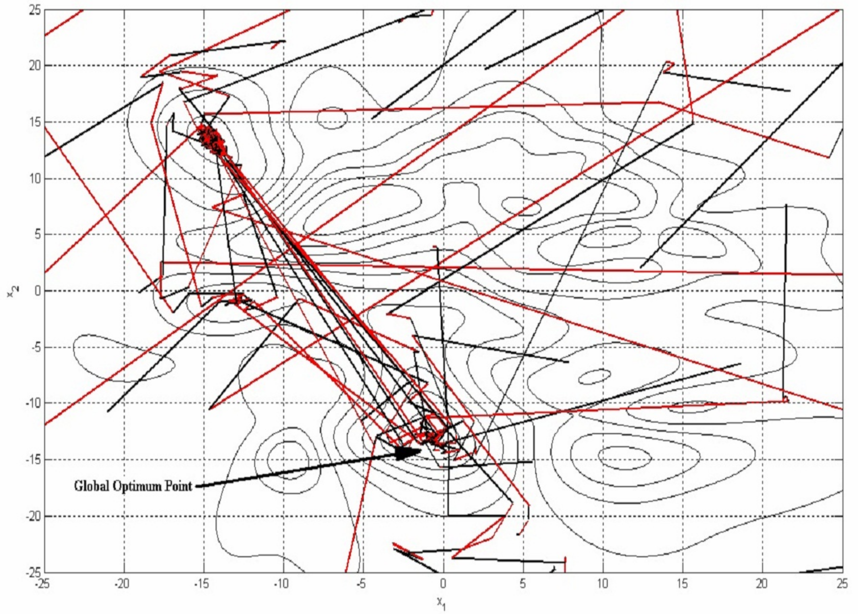


Figure 3.10: Explored region in multi modal function with 100 bell shape functions

3.1.4 Conclusion for Parameter Optimization Algorithm: SBFO

In this section, a bio-inspired method called "Spiral Bacterial Foraging Optimization Method" is introduced. Ease of use, simplicity of concept, and speed of convergence are some major advantages of this algorithm. The inclusion of random operators reduces premature convergence, which would otherwise limit the capability of the algorithm. The results show the proficiency of the SBFO in terms of speed of convergence compared with the existing algorithms under the same circumstances.

SBFO as a gradient based global optimization algorithm is expected to have a strong

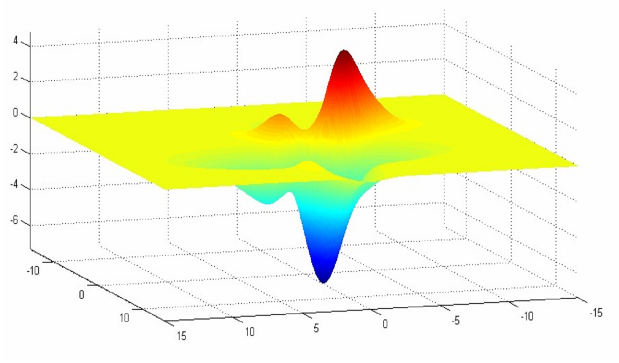


Figure 3.11: A bumpy surface produced by adding six Gaussian bell functions

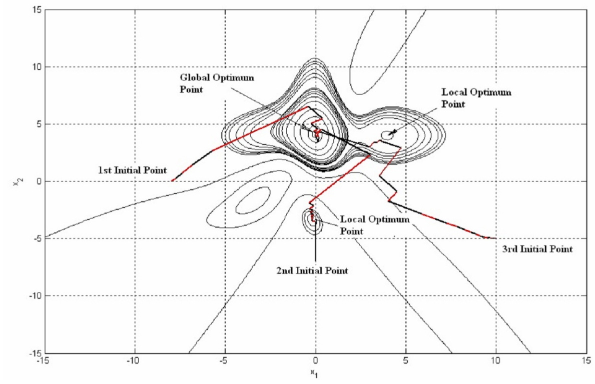


Figure 3.12: SBFO helps to escape from local optimum points

mathematical proof of convergence. In Section (3.1.2) convergence properties were established by expanding on [42] by Bertsekas et al. Having a general proof of convergence is rare among the global optimization methods, but SBFO benefits from possessing this significant asset.

Starting from an initial set of agents, the individuals move spirally toward the best available agent, and this center point might switch to another agent once a better solution is found. The proof presented here only focuses on the last stage of convergence as only a finite number of switches occur (Lemma(6). Therefore, if one uses SBFO as an algorithm to explore a multi-modal region in, for example, path planning problems, the theorems (mainly Theorem (3)) should be modified in order to guarantee the stability of a stochastic switching process.

It has been noted that one of the most important advantages of SBFO is its limited number of tunable parameters (namely λ and number of agents), but it is worth mentioning that a wise selection of parameter λ can significantly affect the performance of the algorithm. Lemmas (3), (4) and (5) are intended to make a constructive foundation in this regard. There may be other ways to define λ properly that are not covered in this thesis and postponed to the ongoing part of this research.

3.2 Proposed Algorithms for Variational Framework

In previous section, it has been shown that a an effective global optimization method should have i) a stochastic operator, or and/or ii) a multi-agent structure. These two properties are very common in the existing global optimization methods. To improve the computational time and costs, the algorithm may include gradient-based approaches to increase the convergence speed. This property is exclusively available in SBFO and it is the basis on which SBFO can be extended to variational framework. Level set method is chosen for optimization in variational framework, and as known, it is an extension of steepest descent method in parameter optimization. The same way that steepest descent method is generalized by SBFO method such that it can find the global solution, level set method can be extended in order to find global solution of variational problems. In three steps, the variational set up is formulated: i) A single stochastic level set method, called "Active Contours with Stochastic Fronts" (ACSF), ii) Multi-agent stochastic level set method (MSLSM), and iii) Stochastic level set method without gradient such as E-ARC algorithm. The first two algorithms can be formulated in general and are presented in this chapter. The latest step, when there is no closed form equation to described the gradient, is specifically designed for path planning problem, so it will be discussed in next chapter that only concerns with engineering applications.

3.2.1 Active Contour with Stochastic Fronts (ACSF) Algorithm

One of the most important applications of stochastic partial differential equations is in image processing and segmentation [57]. By employing the stochastic level set method, apparently, fronts should be able to escape from local optima, but the simulation shows that with small number of noise sources it only works to capture the outer boundary of the features in image. To improve the capability of the algorithm in bypassing the local edges, each point on the evolving curve should have an independent random jump. By local edge, it is meant the situations where the evolving fronts are trapped at some locations and cannot evolve anymore. Also real edges are the edges that are detected by the algorithm and actually exist in the real image. This issue can be encountered in topology optimization problem as well as image segmentation.

One way to bypass the local edges is to use the delta function as spatial weighting functions in Equation (2.56). This way each point over the front has an independent random jump. As the finite difference algorithm has been employed for numerical simulations, every node on evolving contour a delta function is assigned. Substituting delta function and its derivative into Equation (2.56), it can be re-written as Equation (3.62) as long as ε is small enough.

$$\begin{aligned} d\phi &= G(D^2\phi, D\phi, x, t)dt + |D\phi| \sum_{l=1}^{m} \delta_{\varepsilon,l}(x) dW_l(t) \\ &+ \frac{1}{2}\left(\left(\sum_{l=1}^{m} \delta_{\varepsilon,l}^2(x)\right)(\Delta\phi - |D\phi|\,\kappa)\right)dt \\ &+ \frac{1}{2}\left(\left(\sum_{l=1}^{m} \delta_{\varepsilon,l}(x) D\delta_{\varepsilon,l}(x)\right)\cdot D\phi\right)dt \end{aligned} \tag{3.62}$$

here $G(D^2\phi, D\phi, x, t)$ is the deterministic model. In this thesis Chan-Vese model, Equation (4.8), is chosen and obviously any other model can be selected as well. Also m is the number of node on evolving contour that by definition of ACSF, is equal to the number of delta functions. $\delta_{\varepsilon,l}(x)$ is the Dirac delta function centered at l^{th} node of finite difference grid, $l = 1...m$. An approximation for Dirac delta function with width of ε is represented in Equation(3.63).

$$\delta_{\varepsilon,l}(x) = \frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2 + (x - x_l)^2} \tag{3.63}$$

Assuming $\varepsilon \ll L_e$ where L_e is the minimum grid size, $\delta_{\varepsilon,l}(x)$ is almost zero in all nodes but l^{th} node. Keeping this point in mind, Equation (3.62), or basically the original Equation (2.43), can be rewritten in the following form:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + D\phi(x,t) \cdot \Upsilon(x,\varepsilon) \circ dW(t) \tag{3.64}$$

where $\Upsilon = [\upsilon_{il}]$ is a matrix of $m \times m$ in which the entities are defined as:

$$\upsilon_{il} = \delta_{\varepsilon,l}(x_i), \qquad \forall i, l = 1...m \tag{3.65}$$

One issue with this algorithm is that the edges continue wobbling for a relatively long time before the process reaches the steady state; therefore an edge solidification expression should be added to the equation. Here an exponential term is employed for this purpose. The modified equation is as follows:

$$d\phi = G(D^2\phi, D\phi, x, t)dt + e^{-\frac{t}{\tau}} D\phi(x,t) \cdot \Upsilon(x, \varepsilon) \circ dW(t) \tag{3.66}$$

In which $\frac{1}{\tau}$ is the solidification rate and should be determined by trial and error to achieve better performance.

Convergence Analysis

In the theorem proposed by Lion and Souganidis [72]- [73], the noise source has no spatial dependency. The general formulation in Equation (2.56) is fully dependent on spatial terms represented by ψ_i functions. Basically, Theorem 1 cannot support any formulation with spatial-dependent noise. In Equation (3.66), a simpler form is obtained which luckily can be supported by a recent theorem proposed by M. Caruana et.al, in [81]. In spite of what is expected, it has been shown that in presence of spatial term, the stochastic partial differential equation would not necessarily converge to the solution of the deterministic part. Although the form employed by [81] is not the general form, it can obviously confirm that the Theorem 1 is not valid for a general case where the spatial terms exist. To discuss the new theorem and application of it in ACSF, some background information is required.

Nilpotent Lie Algebras Given C^∞-vector fields $H_i = \sum_{j=1}^n H_j^i(x)\frac{\partial}{\partial x^j}$, $1 \le i \le q$ on $\mathbb{R}^n$. Lets denote $L(H_1, \dots, H_q)$, the Lie sub-algebra of $\mathcal{I}(\mathbb{R}^n)$, expanded by $\{H_1, \dots, H_q\}$. Here $\mathcal{I}(\mathbb{R}^n)$ is the Lie algebra of all C^∞-vector fields on $\mathbb{R}^n$ in which the bracket product operator is defined as [90]:

$$[X, Y] = X \otimes Y - Y \otimes X, \qquad X, Y \in \mathcal{I}(\mathbb{R}^n) \tag{3.67}$$

This operator can be expanded as:

$$[X, Y]^i = \sum_{k=1}^n (X^k \frac{dY^i}{dx^k} - Y^k \frac{dX^i}{dx^k}) \text{ where } i \in \{1, ..., q\}, q < n \tag{3.68}$$

Equation (3.68) shows the i^{th} entity of the vector field $[X, Y]$.

A Lie algebra L is called Nilponent of step p if p^{th} term of the series in Eqaution(3.69) vanishes [90].

$$[L, L] \supset [L, [L, L]] \supset [L, [L, [L, L]]] \supset \cdots \tag{3.69}$$

in which

$$\begin{aligned} &\forall A, B \subset L : \\ &[A, B] = \left\{ \sum_{i=1}^{k} [a_i, b_i];\ a_i \in A,\ b_i \in B,\ i = 1, \ldots, k,\ k = 1, 2, \ldots \right\} \end{aligned} \tag{3.70}$$

Supporting Theorem and Remark In a paper by M. Caruana, P.K. Friz and H. Oberhauser [81] preceded by a set of papers by the last two authors [82]- [83], nonlinear parabolic evolution of the form $\partial_t \phi = G(D^2\phi, D\phi, x, t)$ subject to noise of form $H(D\phi, x) \circ dB$ for linear H with respect to $D\phi$ has been studied. As the authors stated this set of researches are motivated by the work done by P.L. Lions and P.E. Souganidis [72]- [73] that is mentioned in the Theorem 1. To deal with the recent case, they proposed the use of rough path analysis suggested by T.J. Lyons [89]. The following theorem is the main result of their work which is very useful for convergence analysis of ACSF.

Theorem 4. *Let $S = (S_1, \ldots, S_m)$ be a collection of C^∞-bounded vector fields on $\mathbb{R}^n$ and W a m-dimensional standard Brownian motions. Then, for every $\alpha = (\alpha_1, \ldots, \alpha_N) \in \{1, \ldots, m\}^N$, $N \geq 2$, there exist (piecewise) smooth approximations (z^k) to W, with each z^k only dependent on $\left\{W(t) : t \in D^k\right\}$, where D^k is a sequence of dissection of $[0, T]$ with mesh tending to zero, such that almost surely*

$$z^k \to W \text{ uniformly on } [0, T]$$

but u^k, solution to

$$\begin{aligned} &d\phi^k = G(D^2\phi^k, D\phi^k, x, t)dt - D\phi^k(x, t) \cdot S(x)dz^k \\ &\phi^k(0, \cdot) = \phi_0 \in BUC(\mathbb{R}^n) \end{aligned} \tag{3.71}$$

converges almost surely locally uniformly to the solution of the "wrong" differential equation:

$$\begin{aligned} d\phi &= [G(D^2\phi^k, D\phi^k, x, t) - D\phi(x, t) \cdot S_\alpha(x)]dt \\ &- D\phi(x, t) \cdot S(x)dW \end{aligned} \tag{3.72}$$

where S_α is the bracket-vector field given by:
$S_\alpha = [S_{\alpha_1}, [S_{\alpha_2}, \cdots [S_{\alpha_{N-1}}, S_{\alpha_N}]]]$.

Remark 5. *It can easily be shown that the preceding theorem is also valid when the Stratonovich differential* $\circ dW$ *is replaced by* dz *for some* $z \in C^1([0,T], \mathbb{R}^m)$*.*

An outline of the proof for Theorem 4 and the Remark 5 is available in [81]. Using this theorem and remark, one is able to verify the convergence of ACSF. The next section is devoted to the convergence analysis of ACSF.

Convergence of ACSF A simple comparison between Equation (3.64) defining the ACSF evolution equation and Equation (3.71) in Theorem 4, shows that the equations have the same structure except that Equation (3.64) is written in Stratonovich sense.

Considering Theorem 4 and Remark 5, one can conclude that assuming $S(x) = -\Upsilon(x, \varepsilon)$, according to Theorem 4, the SPDE (3.66) will converge to a "wrong" differential equation such as:

$$\begin{aligned} d\phi &= (G(D^2\phi, D\phi, x, t) + D\phi(x,t)\Upsilon_\alpha(x,\varepsilon))dt \\ &+ D\phi(x,t)\Upsilon(x,\varepsilon) \circ dW(t) \end{aligned} \tag{3.73}$$

where suppose $\Upsilon = [\Upsilon_1, \cdots, \Upsilon_m]$ then $\Upsilon_\alpha = [\Upsilon_{\alpha_1}, [\Upsilon_{\alpha_2}, \cdots [\Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}]]]$, $\alpha = (\alpha_1, \ldots, \alpha_m) \in \{1, \ldots, m\}^m$, which is basically a bracket-vector field made of Dirac delta functions and Lie bracket operator in Nilpotent fashion.

Now it is required to show that in ACSF, Υ_α is always zero. In other words, the "wrong" differential equation becomes exactly the same as the initial equation, or the stochastic level set equation will converge to the solution of deterministic part which is what is needed.

Theorem 6. *Given the stochastic differential equation (3.64), ACSF evolution equation converges to deterministic solution of Hamilton-Jacobi equation* $d\phi = G(D^2\phi, D\phi, x, t)dt$*.*

Proof. To proceed with the proof, lets focus on Υ_α. One may claim that the Lie algebra, L, defined over $\{\Upsilon_{\alpha_1}, \Upsilon_{\alpha_2}, \cdots, \Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}\}$ is Nilponent of step 1 or simply:

$$[\Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}] = 0, \qquad \forall \alpha_i \in \{1, \ldots, m\}^m \tag{3.74}$$

To show that this claim is true, lets recall Equations (3.65) and (3.68) for Dirac delta functions by which the expanded version of Lie bracket is defined.

$$\begin{aligned} & [\Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}]^i \\ &= \sum_{l=1}^{m} (\upsilon_{l,\alpha_{m-1}} \frac{d\upsilon_{i,\alpha_m}}{dx_l} - \upsilon_{l,\alpha_m} \frac{d\upsilon_{i,\alpha_{m-1}}}{dx_l}) \\ &= \sum_{l=1}^{m} (\delta_{\varepsilon,l}(x_{\alpha_{m-1}}) \frac{d\delta_{\varepsilon,i}(x_{\alpha_m})}{dx_l} - \delta_{\varepsilon,l}(x_{\alpha_m}) \frac{d\delta_{\varepsilon,i}(x_{\alpha_{m-1}})}{dx_l}) \end{aligned} \qquad (3.75)$$

By investigation of Equation(3.75), one can see that for each particular x, one of $\delta(x_{\alpha_{m-1}})$ or $\delta(x_{\alpha_m})$ is zero. This means that both terms inside the bracket in the last equation are zeros, so $[\Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}]^i$, an arbitrary element of $[\Upsilon_{\alpha_{m-1}}, \Upsilon_{\alpha_m}]$ is always zero, therefore the whole bracket in Equation (3.74) is zero. It leads to this conclusion that the Lie algebra, L, is Nilponent of step 1. Based on this conclusion the extra term in deterministic part of "wrong" Equation (3.72) (second term in the bracket) compared to Equation (3.64) is always zero for ACSF evolution. It can confirm that all of the performed simulations based on ACSF differential equations are at least a local solution of the original optimization problem. □

Annealing Scheme to Improve the Chance of Finding the Global Solution and Speed of Convergence

One way to improve the convergence speed of a numerical minimization such as ACSF, is to accept only those iterations that decrease the objective function. Doing so, objective function decrease monotonically but there is a chance of being stuck in a local optima when the process cannot find a better solution. Conversely, ACSF by its own allows the process to increase (or decrease) the objective function in some iterations randomly. It is good for escaping from local optima but it may result in longer convergence time. One intermediate option, is to accept iterations with higher objective function by a slight chance that is exponentially related to how far the value of current objective function is from previous one. Obviously if the objective function is slightly higher than previous iteration it is more likely accepted rather than the case in which a big jump occurs and objective function increased dramatically. The combination of ACSF and annealing scheme is summarized in Table 3.1.

It should be noted that, even by employing the annealing scheme, there is still a chance of being stuck in a local optima when the process can not find a better solution. Generally speaking, using delta function as weighting function in ACSF helps the active contour to have an independent source of noise at each point of the front such that each point can find a better location at each iteration if possible.

Table 3.1: A pseudo code for annealing scheme

Step 1	Define $Temp_0$, stopping criteria and $i = 1...N_{Itr}$
Step 2	Start with initial guess ϕ_1 and evaluate the objective function $J(\phi_1)$
Step 3	update $\phi_i \leftarrow \phi_{i+1}$ based on ACSF update scheme, Equation (3.66).
Step 4	Calculate $J(\phi_{i+1})$
Step 5	$\begin{cases} \text{If J}(\phi_{i+1})<\text{J}(\phi_i)\Longrightarrow & \text{Accept } \phi_{i+1} \\ \text{Else} \Longrightarrow & \text{Accept } \phi_{i+1}\text{with} \\ & probability: \\ & P_{acc}\text{=exp(-}\frac{J(\phi_{i+1})-J(\phi_i)}{Temp(i)}) \end{cases}$
Step 6	Update $Temp(i)$ and loop back to (S2) to fulfill the stopping criteria

In this scheme, $Temp(i)$ is the annealing temperature which is usually a monotically decreasing function of time (iteration). Here this function is defined as:

$$Temp(i) = \frac{Temp_0}{\sqrt{i}} \tag{3.76}$$

where i is the iteration number and $i = 1...N_{Itr}$ in which N_{Itr} is the maximum number of iterations, and Temp0 is the initial guess for temperature [57].

Advantages of Active Contour with Stochastic Fronts

As discussed earlier, two approaches have been proposed in order to improve the performance of active contours in image segmentation: active contour without Edges [64]- [68]

and stochastic level set method [57], [72]- [73]. The first approach omits the dependency of evolution of front on local image gradient, so it is very successful to cope with the images corrupted with noise. But in some cases the real edges within the image are barriers for evolution of active contour. A good example is the image shown in Figure (4.5(a)), the strips on zebras' body prevent the contour to evolve completely and extract the whole bodies. One the other hand, according to the Theorem (4) from [81], the general model developed by [57] will converge to a "wrong" equation that is not necessarily the deterministic model. This is not desirable as the deterministic models are designed such that the optimization process converges to an accurate segmentation. Basically the stochastic algorithms should find the best solution of the deterministic models.

Active contour with Stochastic Fronts (ACSF) as a special case of model in [57], is an attempt to achieve all of the previously-mentioned advantages and mitigate the flaws.

It uses both formulation of region-based and stochastic level set method with a very simple but very effective change in order to extract all possible details within image as well as outer boundaries. At the same time the convergence of the resultant equation toward the deterministic solution is also guaranteed.

3.2.2 Multi Agent Stochastic Level Set Method

In global optimization theory, there exist two main approaches to cope with local optima problems: stochastic operators and multi-agent systems. An interesting feature of the work in [57] is the combination of the stochastic operator with the level set method. This innovation creates a global optimization algorithm with the advantages of gradient based methods. In this section, the second approach (Multi Agent) is intended along with taking the advantages of the first one. In doing so, instead of only one partial differential equation, a set of level set equations are employed. The main idea here is to make sure that the agents are capable of escaping from a local point. Therefore, instead of one H-J equation in active contour with stochastic front (ACSF), several SPDEs have been employed starting from different initial guesses. This can help the evolving front to escape from local edges when the stochastic term in Equation(3.66) is not strong enough for this purpose.

Certainly, all of the agents have to converge to the best current topology, namely leader, so the functional in Equation (2.6) must be revised in order to make this opportunity. The revised equation is:

$$J_{fj}(\partial\Omega_l, \partial\Omega_j) = J_1(\partial\Omega_j) + J_2(\partial\Omega_j) + \lambda_{lj} J_{cj}(\partial\Omega_l, \partial\Omega_j) \tag{3.77}$$

where:

$$J_{cj}(\partial\Omega_l, \partial\Omega_j) = \int\limits_{\Omega} \left(H(\phi_l)\left(1 - H(\phi_j)\right) + H(\phi_j)\left(1 - H(\phi_l)\right)\right) d\Omega \tag{3.78}$$

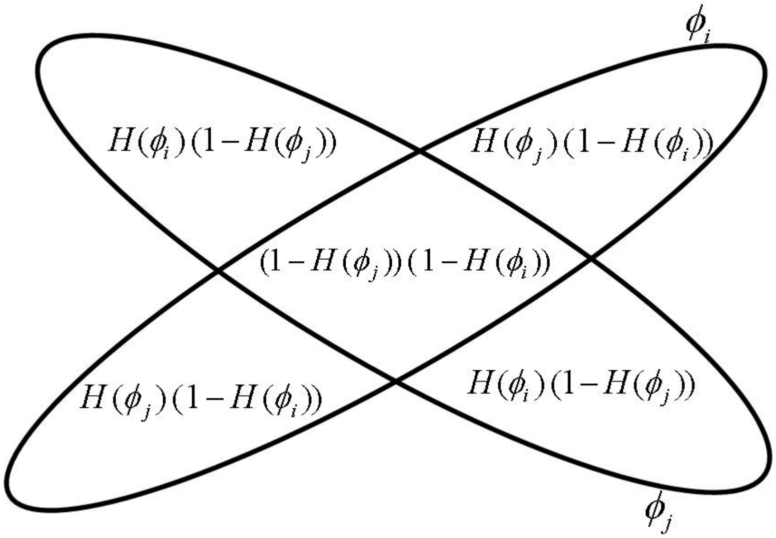


Figure 3.13: Region surrounded by agents

The converging force $J_{cj}(\partial\Omega_l, \partial\Omega_j)$ is simply a metric showing the region that is surrounded between the leader, l^{th} agent and the agent $j^{th}, l \neq j$ but does not include the intersection area. To achieve the same topology, the area of this region has to converge to zero. Figure (3.13) shows the surrounded area between the two agents i and j in general. It is worth mentioning that although Equation (3.77) is written only for two agents, namely leader and an arbitrary individual, it can be employed for a system with any number of agents because the "converging force" is only applied between leader and each individual. Obviously the leader evolves based on Equation(2.13) only and the converging force is equal to zero for this agent, $J_{cl}(\partial\Omega_l, \partial\Omega_l) = 0$.

Advantages of Multi-Agent Stochastic Level Set Method

As stated, the main problem in LSM, or indeed in the stochastic level set method, is that when the fronts converge to an edge, there is no guarantee that this edge is a real edge or even if the edge is real, the segmentation may be stuck over that edge and can not evolve to find a better solution. To increase the probability of finding a real edge, the zero level sets should start from different parts of the feasible region; consequently, a multi-agent system can overcome this problem.

In level set method, it is very common that one employs an initial level set such that it has enough holes in it. It might seem that a stochastic level set method with multiple holes in initial guess, Figure (4.3), is an multi-agent stochastic system by itself and there is no need for using a complicated system such as the one proposed in in this thesis. To clarify the difference between a ACSF algorithm initiated with a multiple holes zero level set and the multi-agent structure proposed here, one should notice that every single node on the front in first case evolves independently, so optimum solution for this case is similar to a case that one runs a gradient based optimization procedure with different initial guesses, hence each agent finds its own local optimum points. But in the latter case as the agent are forced to finally converge to the same shape, the probability of finding the real edges in image is more than the first case. This is similar to convergence of individuals in multi-agent optimization algorithms such as GA, PSO and/or SBFO [1], in which each agent starts from different initial guess but they are all supposed to converge to the best possible solution. Obviously, this is more probable to find the global solution rather than the case in which each agent only find its local solution depend upon the initial condition that it starts with. This point has been demonstrated by an example in Section (4.1.2) and (4.2.2).

In addition, using a multi-agent system helps to explore the whole feasible region, which clearly increases the chance of finding the global solution. Although the computational cost of this algorithm is higher than the previous ones, it can be performed in a decentralized fashion, as each agent evolves almost independently based on its own dynamics. Compared to other multi-individual optimization algorithms like GA or PSO, the convergence should be faster, as this algorithm is an extension of a gradient-based called SBFO algorithm [1] to the variational framework.

In contradistinction to GA, PSO, etc., the number of tunning parameters is extremely low, namely the number of agents and weighting parameter λ_{lj} in Equation(3.77).

Chapter 4

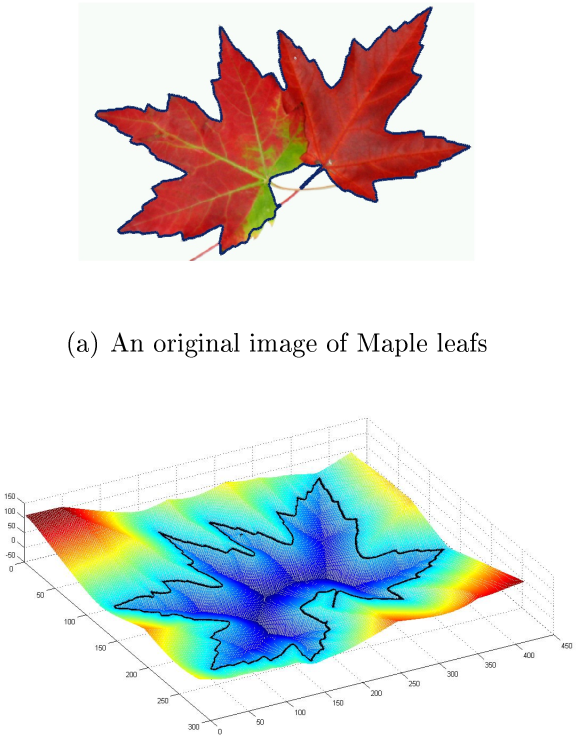
Applications of ACSF and MSLSM in Engineering Problems

4.1 Application of ACSF and MSLSM Image Segmentation

This section is devoted to the application of ACSF and MSLSM to image segmentation. Section 4.1.1 provides a brief background on edge-based, region-based formulation of image segmentation in a variational framework. Then by employing the theories developed in Chapter 3 in Section 4.1.2, results are investigated, and finally in Section 4.1.3, a comprehensive conclusion is given.

4.1.1 Image Segmentation and Level Set Based Methods

In the following, a brief overview of edge-based level set method [52]- [54], re-formulation of LSM for active contours without edge [66]- [68] and stochastic level set method [57], [145] will be presented. One way to address the segmentation problem is to find curve(s) or surface(s) that surround the region of interest in an image. Essentially, this is the primary idea behind active contours methods. However, in these algorithms, the curve(s) or surface(s) can be defined directly using control points, which can result in some issues in the control points updating, merging and splitting closed curve(s) or surface(s). Despite this, there are remedies for these issues, and level set method can easily cope with them. The concept behind the level set method is to embed the above-mentioned curves in a surface, such that the evolution of the curve surrounding the objects in an image can be implicitly represented by the evolution of the surface. Figure(4.1(a)) and (4.1(b)) show how a surface embedding a curve can be morphed in order to achieve a desired topology.



(a) An original image of Maple leafs

(b) Evolved surface to segment the image

Figure 4.1: The basic idea behind the Level Set method in image segmentation

Edge-Based Level Set Method Formulation

As discussed in Section 2.1.1, the level set function is founded on the basis of the Hamilton-Jacobi equation [143]:

$$\phi_t + V_n \left|\nabla\phi\right| = 0 \tag{4.1}$$

The so-called V_n is known as a normal velocity of the evolving boundaries that control the speed evolution of the fronts, and $|\nabla\phi|$ is the amplitude of the gradient vector. Solving this partial differential equation starting from an initial guess transfers the segmentation problem to an initial value problem in PDE framework. A popular formulation [143] for V_n in equation (4.1) is as follows:

$$\nabla I^G = (I_x^G, I_y^G) \tag{4.2}$$

$$V_n = \exp(-\alpha \left|\nabla I^G\right|) \tag{4.3}$$

where I^G is the filtered image by a Gaussian filter G, α is the convergence rate V_n and I_x^G and I_y^G are the partial derivatives of Image, I, with respect to x and y respectively. As can be seen, V_n depends on a local gradient, which can lead to failure in inaccurate segmentation.

Region-based Formulation and Active Contour without Edge (ACWE)

In the region-based LSM formulation, an energy functional contains the fitting energy and the energy associated with the length and the area of the initial contour $\partial\Omega$, $\partial\Omega \subset \Omega$, $\phi : \Omega \longrightarrow \mathbb{R}$. This functional energy can be represented as follows [64]- [68]:

$$E(\partial\Omega) = F_f + \mu L_{\partial\Omega}^p + \upsilon A_{\Omega^-} \tag{4.4}$$

where $\partial\Omega$ is the boundary of Ω, $L_{\partial\Omega}$ is the length of $\partial\Omega$, $p = \frac{N}{N-1}, N$ is the dimension of $\mathbb{R}^N$, and A_{Ω^-} is the area inside the $\partial\Omega$.

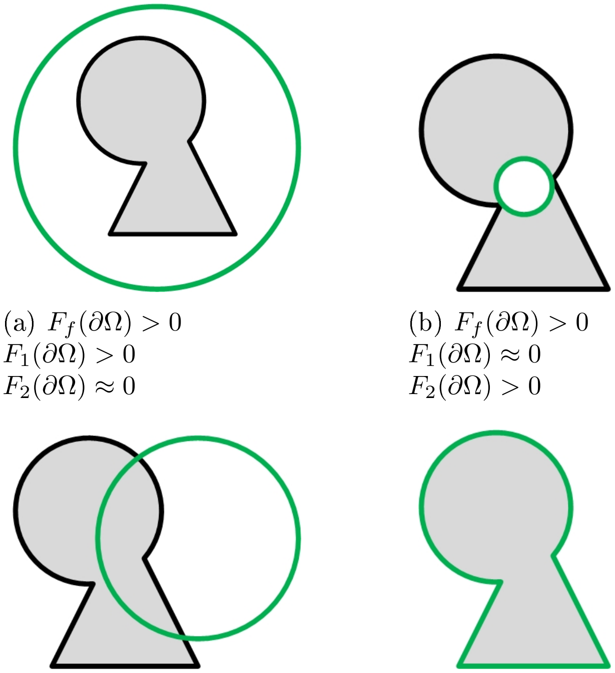
The LSM as an optimization procedure will minimize the functional in equation (4.4) in order to segment the image. The fitting energy is defined as [64]- [68]:

$$F_f = F_1(\partial\Omega) + F_2(\partial\Omega) = \int_{\Omega^-} |I - c_1|^2 \, d\Omega + \int_{\Omega^+} |I - c_2|^2 \, d\Omega \tag{4.5}$$

in which the c_1 and c_2 are the average of image I inside and outside of the $\partial\Omega$, respectively.

$$c_1(\phi) = \frac{\int_\Omega I\,(1 - H_\varepsilon(\phi))dxdy}{\int_\Omega (1 - H_\varepsilon(\phi))dxdy} \tag{4.6}$$

$$c_2(\phi) = \frac{\int_\Omega I\,H_\varepsilon(\phi)dxdy}{\int_\Omega H_\varepsilon(\phi)dxdy} \tag{4.7}$$



(a) $F_f(\partial\Omega) > 0$
$F_1(\partial\Omega) > 0$
$F_2(\partial\Omega) \approx 0$

(b) $F_f(\partial\Omega) > 0$
$F_1(\partial\Omega) \approx 0$
$F_2(\partial\Omega) > 0$

(c) $F_f(\partial\Omega) > 0$
$F_1(\partial\Omega) > 0$
$F_2(\partial\Omega) > 0$

(d) $F_f(\partial\Omega) \approx 0$
$F_1(\partial\Omega) \approx 0$
$F_2(\partial\Omega) \approx 0$

Figure 4.2: Comparison of the fitting function for different cases in image segmentation

To see how this formulation works, let us look at the following four options in Figure(4.2).

As can be seen, the fitting function is always positive except in a case where the evolving curve completely fits the objects in the image and functional F_f is in its global minimum. Therefore, it can be claimed that minimization of the cost function leads to the segmentation of the image. It is known that the steady state solution of Hamilton-Jacobi PDE, Equation(1.1), is the optimum solution of segmentation problem [52]- [54]. The Hamilton-Jacobi PDE can be re-written in Equation(4.8) [64]- [68]:

$$\frac{\partial\phi}{\partial t} = \delta_\varepsilon(\phi)(\mu p \left(\int_\Omega \delta_\varepsilon(\phi)\,|\nabla\phi|\,d\Omega\right)^{p-1} \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) + \left(-\upsilon - \lambda_1(I-c_1)^2 + \lambda_2(I-c_2)^2\right)) \tag{4.8}$$

where the following conditions are held:

$$\begin{array}{ll} p\left(\int_\Omega \delta_\varepsilon(\phi)\left|\nabla\phi\right|\right)^{p-1}\frac{\delta_\varepsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial n}=0 & \text{on } \partial\Omega \\ \phi(t,x,y)=\phi_0(x,y) & \text{in } \Omega \end{array} \tag{4.9}$$

in which $\upsilon, \mu, \lambda_1, \lambda_2 \in \mathbb{R}^+$ are fixed parameters, n, $\frac{\partial\phi}{\partial n}$denotes the exterior vector normal to the boundary and the normal derivative of ϕ respectively, denotes boundary. δ_ε, H_ε are approximation for Dirac delta and Heaviside function as follows:

$$\delta_\varepsilon(x)=\frac{1}{\pi}\frac{\varepsilon}{\varepsilon^2+x^2} \tag{4.10}$$

$$H_\varepsilon(x)=\frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{x}{\varepsilon}\right)\right) \tag{4.11}$$

when ε is a tiny positive parameter and when $\varepsilon \to 0$, these functions approach $H(\phi(x))$, $\delta(\phi(x))$.

4.1.2 Results of Image Segmentation

To evaluate the procedures, a set of benchmark images has been selected, including a nature scene, Figure (4.5(a)), and some medical images [See Tables 4.1-4.4]. The set of medical images is provided by [80]. This collection is a very good source of benchmark images for evaluation of segmentation algorithms. For this thesis, 10 images of breast cancerous tissues are selected.

Region-based level set, stochastic level set, active contour with stochastic fronts and multi-agent stochastic level set methods are applied to the first image and the Figure (4.5(a)) depict the performance of each algorithm. According to the results, Figure (4.5(b)-4.6(c)), one can come to this conclusion that the multi-agent stochastic level set method has the best performance. The performance is defined based on Equation(4.5).

The level set method is capable to find the details within the image but it is stuck over some edges which leads to incomplete segmentation, Figure(4.5(d)). The stochastic level set method usually finds a few details within the image but almost always is able to find the outer boundaries. The results presented in Figure (4.5(b)-4.6(c)) are achieved when 5 noise sources have been considered for Stochastic level set method. It is worth mentioning that, as the stochastic operators are used, each of the algorithms does not necessarily converge exactly to the same segmentation for different simulation but the latter algorithm (MSLSM) has far less deviation as the segmentation is very close to the, global solution, complete segmentation.

Depend upon selection of solidification rate the proposed algorithm could be able to find the outer and inner boundaries but it is a compromisation problem, hence, the smaller the solidification rate, the longer convergence time and the higher probability to find more details exist in the image. On the other hand, even a single stochastic LSM may be able to segment the benchmark image completely, but this does not always occur, for example in Figure (4.6(a)), the rear legs of both zebras are not segmented in single stochastic method.

Moreover, even by using a multi-agent system, there are some cases in which the procedure can not segment accurately. Therefore, one can only discuss the probability of finding the best segmentation (or global solution) which can be improved by increasing the number of agents, the same as the number of individuals in Genetic Algorithm or agents in Particle Swarm optimization methods. Obviously, there is a trade-off between computational cost and accuracy of solution. As expected in Figure (4.6(c)) the steady state solutions of both agents converge to a single value because of the term $F_{cj}(\partial\Omega_l, \partial\Omega_j)$ in Equation (3.77). As mentioned, based on Figure (4.6(c)) the best performance achieved by MSLSM. The performance of stochastic level set method is even worse than region-based level set method in terms of ability to find the details in image.

In practical image processing, and in particular medical image processing and segmentation, an algorithm which is capable to find details within the image is required. For example, in images in Tables (4.1-4.4) one should be able to find the cancerous cells within the breast tissues. Therefore, the algorithms proposed in this thesis are evaluated based on this set of images and compared with region-based level set to show the performance. The relative improvement with respect to region-based LSM is given in forth column of each of the Tables (4.1-4.4). To have a fair comparison, initial guesses for all simulations are the same for all of the algorithms, except that for the multi-agent structure another initial guess is employed. Both of these initial conditions are shown in Figure (4.4).

The percentage of relative improvement of performance in forth columns of Table (4.1-4.4) is calculated based on the ground truth image provided by [80]. Each ground truth image has precisely shown the cancerous cells in a portion of the image called as, Ω_G. So, the calculation should be done based that area only.

The performance, P_f, of each algorithm is defined as a sum square error between values of dots (0 or 1) in Ω_G of the ground truth image, I_G and the corresponding area in the segmented image, I_S, by each algorithm.

$$P_f = \sum_{i=1}^{N_p} (I_G(i) - I_S(i))^2 \tag{4.12}$$

where N_p is the total number if pixels in ground truth area.

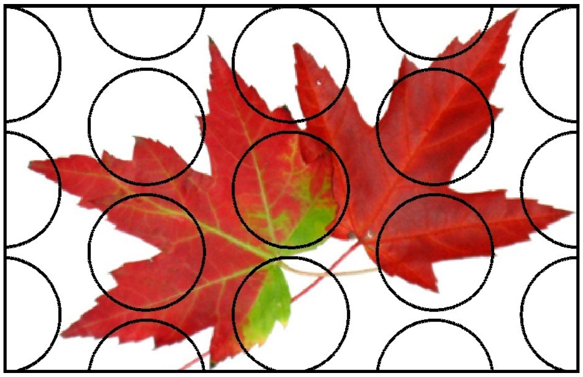


Figure 4.3: Zero Level Set with multiple holes

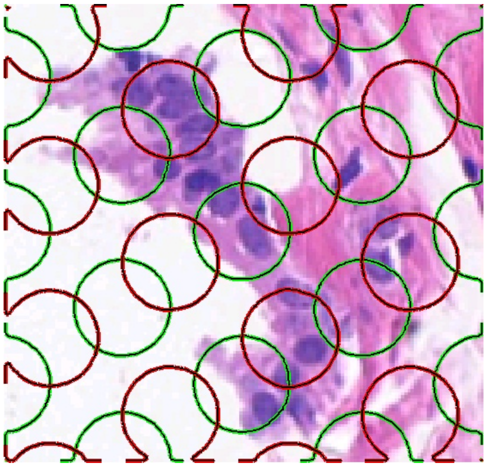
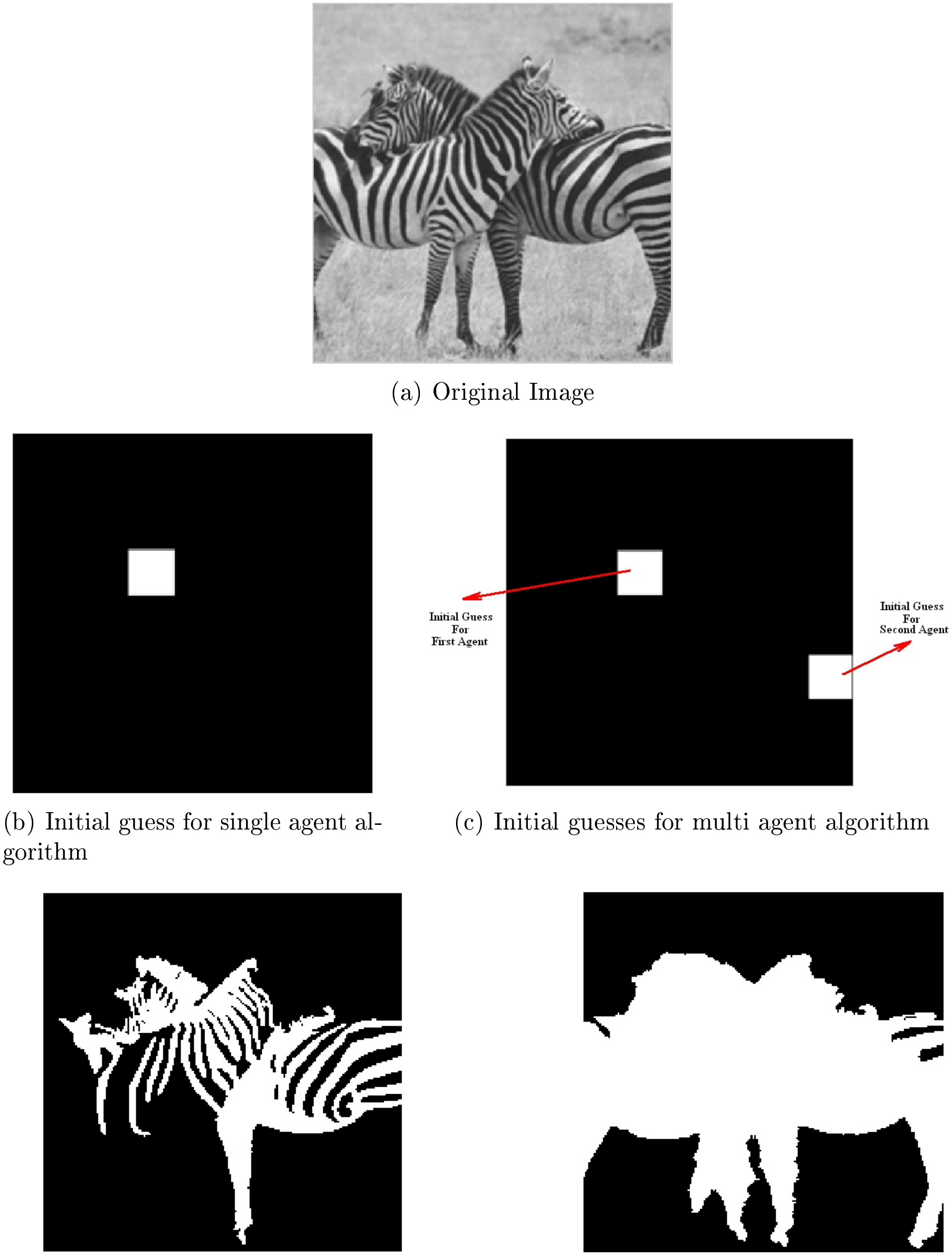


Figure 4.4: Two zero level sets with multiple holes for medical imaging samples



(a) Original Image

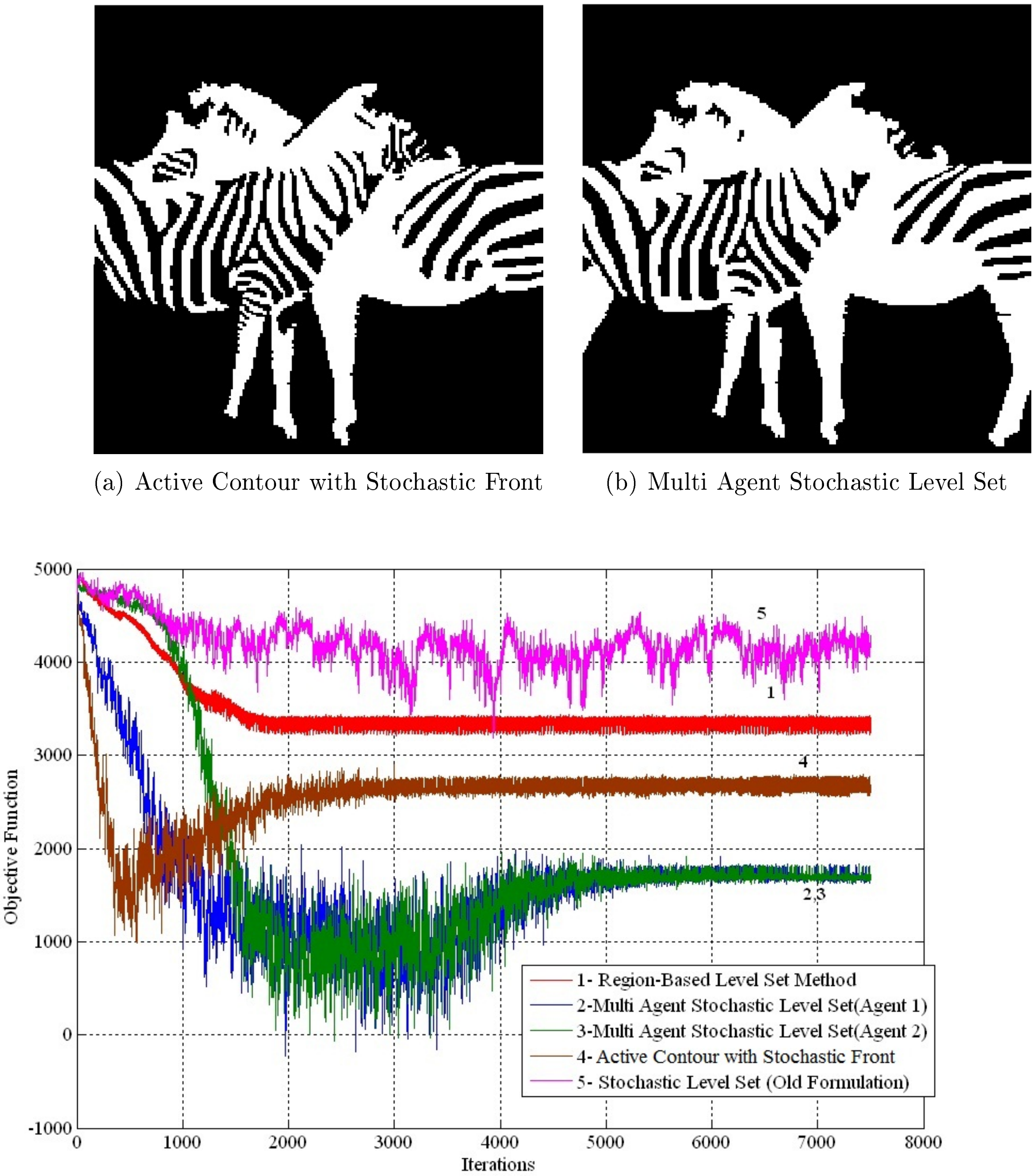
(b) Initial guess for single agent algorithm

(c) Initial guesses for multi agent algorithm

(d) Region-based Level Set method

(e) Stochastic Level Set method

Figure 4.5: Evaluation of existing algorithm on a nature scene



(a) Active Contour with Stochastic Front

(b) Multi Agent Stochastic Level Set

(c) Comparison of Objective Function for Different Algorithms

Figure 4.6: Evaluation of proposed algorithms

Table 4.1: Comparison of ACSF and Region-Based LSM

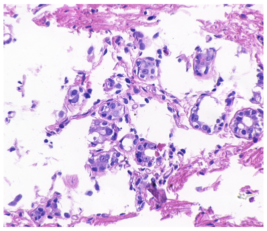
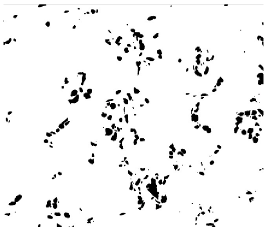
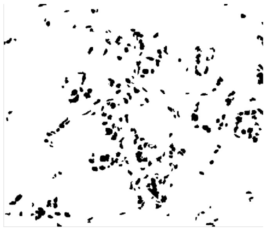
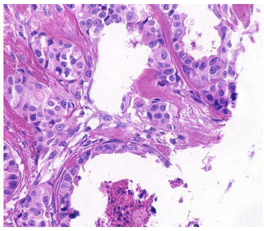
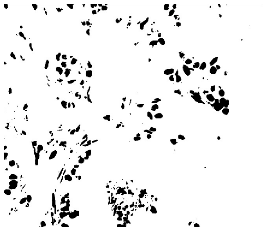
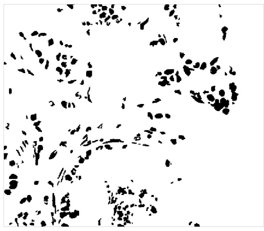
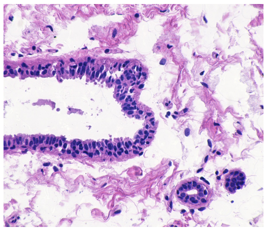
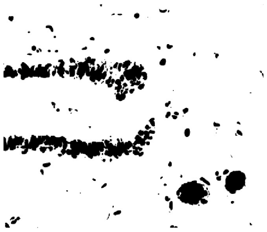
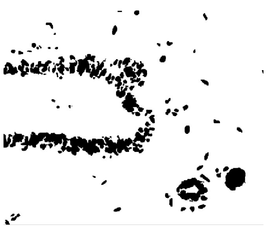
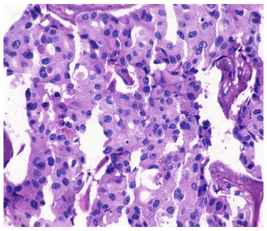
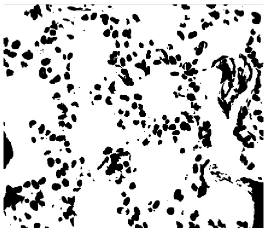
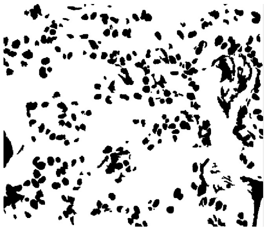
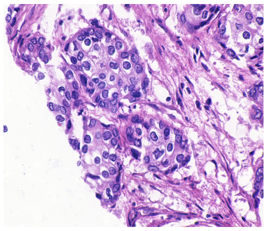
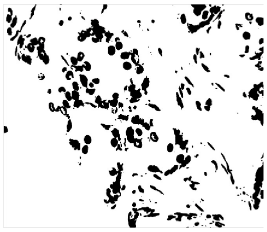
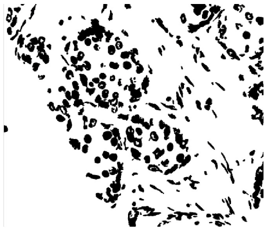
Original Image	Classic LSM	ACSF	Imprv. Prcntg
			45.2
			38.2
			37.8
			32.1
			50.5

Table 4.2: Comparison of ACSF and Region-Based LSM

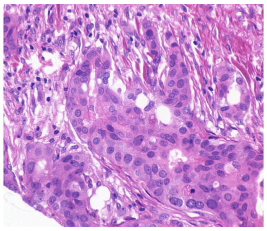
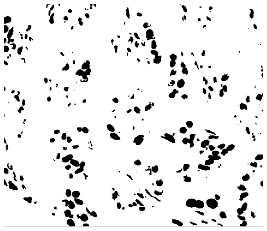
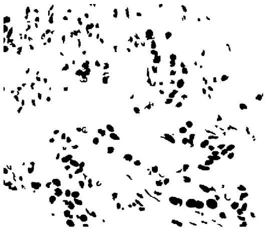
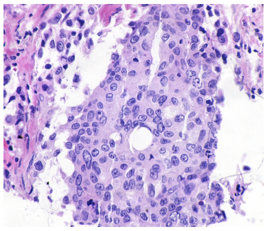
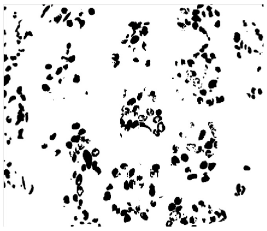
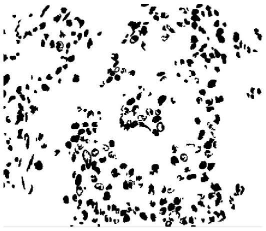
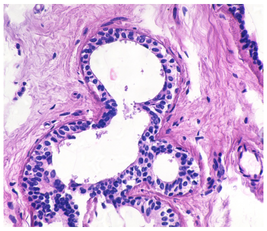
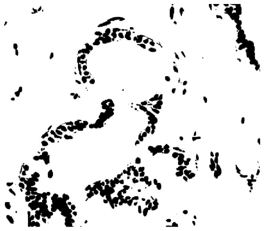
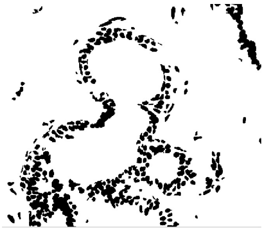
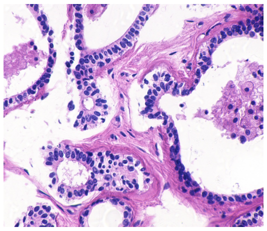
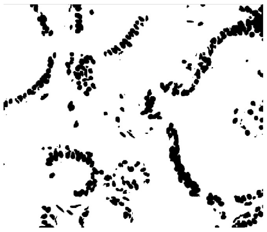
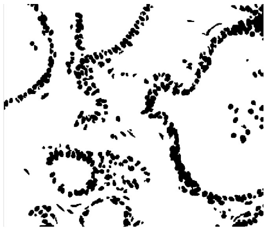
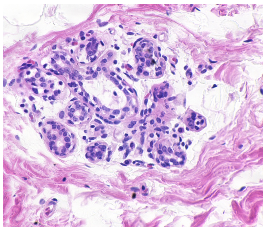
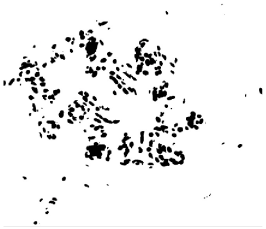
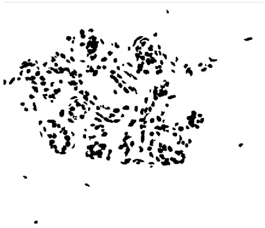
Original Image	Classic LSM	ACSF	Imprv. Prcntg
			31.3
			42.1
			30.6
			27.3
			38.0

Table 4.3: Comparison of MSLSM and Region-Based LSM

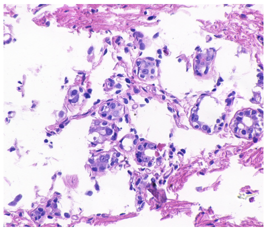
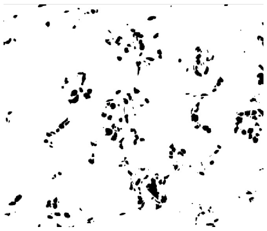
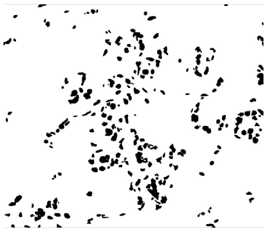
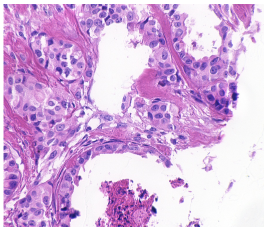
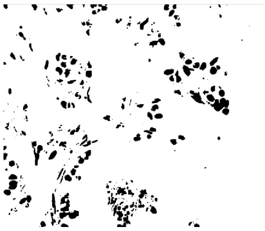
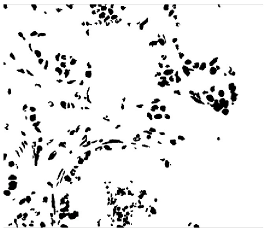
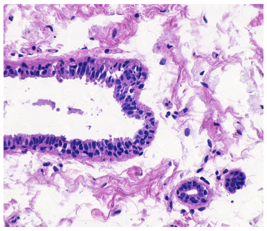
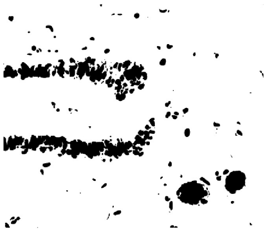
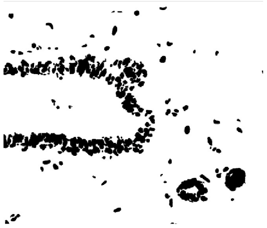
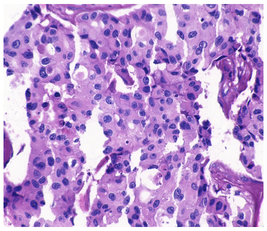
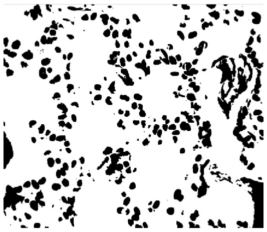
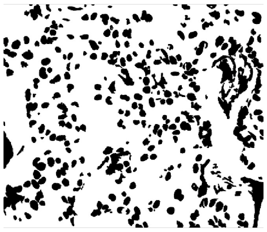
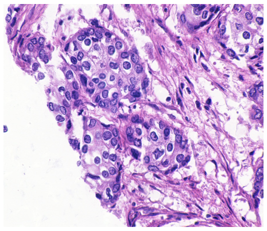
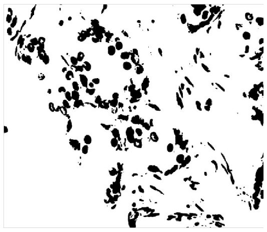
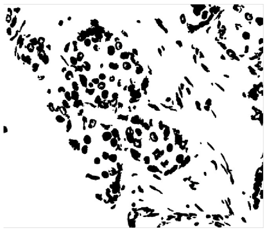
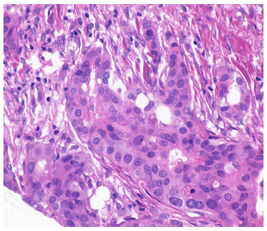
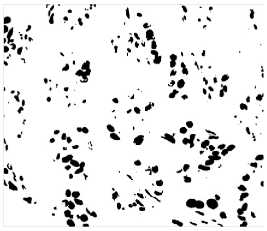
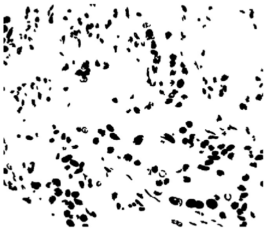
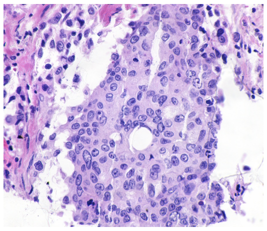
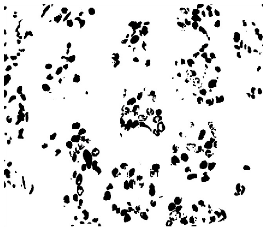
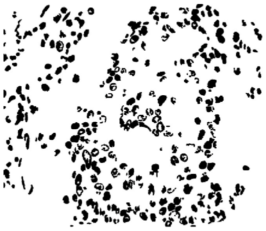
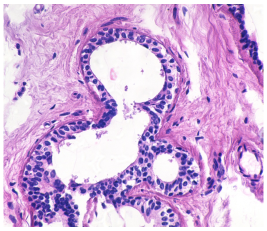
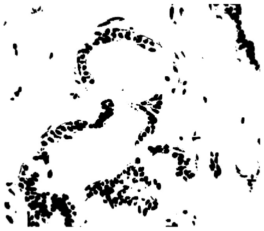
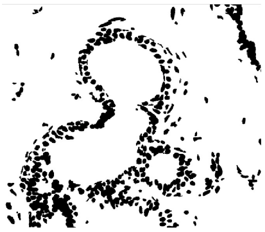
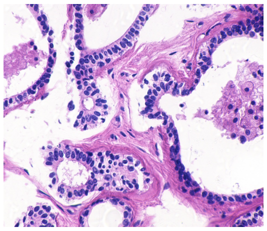
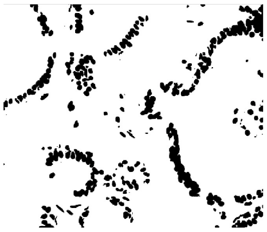
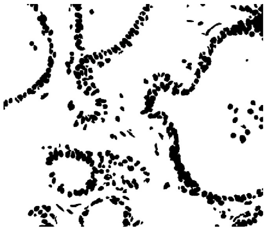
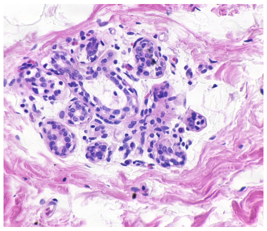
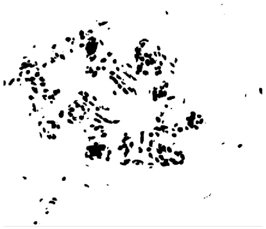
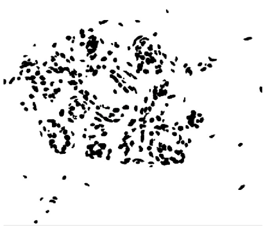
Original Image	Classic LSM	MSLSM	Imprv. Prcntg
			41.3
			35.3
			33.8
			39.7
			52.2

Table 4.4: Comparison of MSLSM and Region-Based LSM

Original Image	Classic LSM	MSLSM	Imprv. Prcntg
			33.6
			45.1
			35.8
			32.7
			40.8

4.1.3 Conclusion of Image Segmentation

Extending the work done by Juan [57], active contour with stochastic front (ACSF),and a multi-agent random structure, called Multi-agent stochastic level set (MSLSM), have been investigated in this thesis. The ACSF, a natural extension of the stochastic level set method, emerges as a well-founded algorithm to represent the stochastic motion of fronts coping with local optima problems. Despite its capability in this regard, there are still some instances where ACSF is not able to find the global solution. A multi-agent structure is thus proposed in this thesis in order to improve the algorithm. In this method, and similar to the stochastic active contours method, each agent evolves based on its own stochastic dynamics developed in Section(3.2).

The random structure proposed in this thesis is significantly effective in helping the algorithm escape from local minima and becomes even more competent when utilized by multi-agent structure. Starting from different initial guesses assists the latter algorithm in exploring the feasible region, helping to find every possible feature in the image. Ultimately, all of the agents must converge on the same topology. To do so, one should add a metric to the original functional given in Equation(4.5). The results, for example in Figure(4.6(c)), show that, as expected, the multi-agent stochastic level set method is more capable of finding the global solution.

As can be seen in Tables 4.1-4.4, both ACSF and MSLSM have significant improvement compared to Region-Based LSM. MSLSM employed two agents. The relative improvement of MSLSM is not very considerable with respect to ACSF, on contrary to what is expected. This is because the ACSF, can find most of the cancerous cells within the image, so the relative improvement of MSLSM is not remarkable compared to ACSF. In other words, ACSF with a zero level set presented in Figure (4.3), is strong enough to segment the selected images. But in some cases that ACSF can not segment the image accurately, MSLSM can help in order to overcome this problem.

It is worth mentioning that, what is claimed in our thesis as global optimum algorithm is somewhat different from what is meant by Bresson et al. [69] in their work. Obviously the capability of edge indication of our algorithm is limited to the deterministic model as our baseline (Here ACWE) and it can not be claimed that the algorithm is more capable (compared to ACWE or any other base model we choose) to find the edges with low contrast. One common issue that happens in segmentation by active contours is that depend upon the initial guess some part of the image will not be discover by the algorithm especially when there are lots of details in image like our application in medical imaging of breast cancerous tissues 4.1-4.4. The reason of this is, once the algorithm find some part of the real edges of the image and all of the active boundaries are busy with real edges, it will not evolve to detect other edges because if it wants to evolve to detect other features it means that it should leave a local solution where the deterministic level set method is not able to do so. As it only evolves based on gradient flow and the gradient flow for this current topology is toward the local solution. The first example of this section is designed to show this concept. As can be seen the stripes in zebras bodies are real edges but the ACWE algorithm is stuck and can not evolve any more to detect all of the stripes. Now if one runs the algorithm for another initial guess that is located in the region that is not discovered by the first trial one might be able to detect other stripes but again it will be stuck in another solution. The idea behind multi agent structure is to have an algorithm that discovers all of the image. Obviously each agent can pull out the other agents from their own local solutions and at the end (hopefully) the whole image is discovered. Stochastic operators can also help to jump from local solutions to discover the whole image, so combination of these two tool, helps to detect every single object in image with higher probability. Obviously if one chooses the baseline model as the one that is proposed by Bresson et al. [69] along with multi-agent ACSF proposed in current section, the final approach becomes a very sophisticated algorithm that not only can detect the low contrast edges but also it can detect most of the features in the image as much as possible.

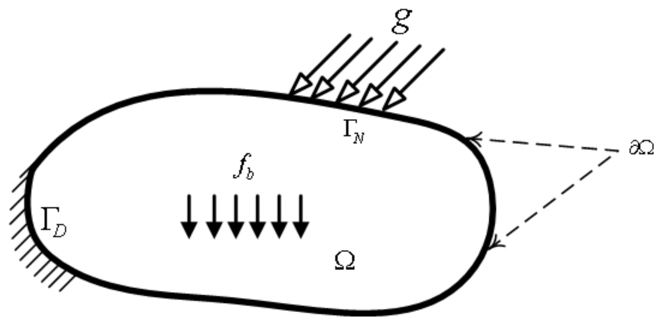


Figure 4.7: A general load case including external and body forces

4.2 Application of ACSF in Topology Optimization

As stated in the literature review Section 1.2.2, the classic level set method, even when it is utilized by topology derivative, suffers from being stuck in local solutions and consequently, it is highly dependent on initial guesses. Topology optimization as an application of level set theory is investigated in this chapter when the ACSF algorithm is applied. The outline of this section is as follows: In Section 4.2.1, the fundamental theories behind the topology optimization problem, including the general formulation of complaint mechanism design, are explained. Then the performance of the ACSF algorithm in topology optimization is investigated and discussed.

4.2.1 Mechanical Structures and Level Set Theory

Using the theories explained in Section 1.2.2, another application of our proposed algorithm is investigated in this section. In this section, the often-used cost functions in topology optimization are introduced and shape and topology derivatives are calculated for this particular problem, then ACSF algoirthm is applied followed by the results and discussion.

Topology Optimization to Minimize Mean Compliance

A general load case for a mechanical structure is shown in Fig. 4.7, in which g is the distributed force and f_b is body force. Γ_N and Γ_D are Neumann and Dirichlet boundary conditions respectively.

Assuming mean compliance optimization may be helpful in understanding how level set equation helps in topology optimization problems, as follows:

$$\begin{aligned}&\underset{\phi}{\text{Min}}\; C(\phi) = \textstyle\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u)\varepsilon_{kl}(v)d\Omega\\ &\text{ Subject to:}\\ &V(\phi) = \textstyle\int_{V} H(\phi)dV = V^{\star}\\ &\textstyle\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u)\varepsilon_{kl}(v)d\Omega = \int_{\partial\Omega} g.v\, ds\end{aligned} \tag{4.13}$$

where E_{ijkl} is Young's modulus elasticity tensor and is defined as:

$$E(\phi) = E_0 H(\phi) + E_{min}(1 - H(\phi)) \tag{4.14}$$

in which E_0 and E_{min} are Young's modulus of elasticity in structure and a dummy value for voids respectively. This problem set up can be re-written in discrete finite element form as follows [119]:

$$\begin{aligned}&\underset{\phi}{\text{Min}}\; C(\phi) = U^T K(\phi) U = \textstyle\sum_{ne=1}^{N} u_e^T k_e u_e = \sum_{ne=1}^{N} x_e u_e^T k_1 u_e\\ &\text{ Subject to: } V(\phi) = V^{\star}\\ &KU = F\\ &x_e = 0 \text{ or } x_e = 1\; \forall e = 1, ..., N\end{aligned} \tag{4.15}$$

where $x = (x_1, ..., x_N)$ is the vector of element densities. $x_e = 0$ for void element and $x_e = 1$ for structural elements. $C(\phi)$ is the mean compliance of the structure. F and U are global force and displacement vectors respectively. K in elasticity equilibrium equation $KU = F$ is global stiffness matrix. k_e and k_1 are general and solid element matrix respectively. N is total number of elements. V and $V^{\star}$ are current and desired volume of structure.

In this set up, the mean compliance of structure is intended to be minimized. For other problems like geometric and mechanical advantage or any other purpose, the main optimization problem (4.13) should be revised.

The level set function is founded on the basis of the Hamilton-Jacobi equation:

V_n and T_n are scalar fields over design domain and w is a weight for T_n. The so-called V_n is known as a normal velocity, or shape derivative, of the evolving boundaries that control the speed evolution of the fronts, and $|D\phi|$ is the amplitude of the gradient vector. T_n is a forcing term and depends on topological derivative of the cost [119]. It basically determines the nucleation of new holes within the structure [99], [119] and [111] . If w is chosen equal to 0, Equation (2.17) turns to the classic level set method for shape (and not topology) optimization, and is unable to create new holes. In order to satisfy the volume constraint, Lagrangian should be defined as follows:

$$L(\phi) = C(\phi) + \lambda_v(V(\phi) - V^\star) + \frac{1}{2\Lambda_v}[V(\phi) - V^\star]^2 \tag{4.16}$$

where λ_v and Λ_v are required to be updated at each iteration based to the following sequence [99], [119] :

$$\lambda_v^{k+1} = \lambda_v^k + \frac{1}{\Lambda_v^k}(V(\phi) - V^\star), \qquad \Lambda_v^{k+1} = \alpha\Lambda_v^k \tag{4.17}$$

in which $\alpha \in (0, 1)$ is a constant value. Using the aforementioned procedure implements augmented Lagrangian multiplier.

Shape Derivatives: To minimize the Lagrangian L one should choose normal velocity field V_n as a descent direction of objective function [119]. Therefore one should take the derivative of Lagrangian L to find the descent direction. For the sake of simplicity, assume that the boundary conditions are traction free so the shape derivative of mean compliance can be found as:

$$\frac{\partial C}{\partial \Omega} = -u_e^T k_e u_e \tag{4.18}$$

similarly the shape derivative for volume $V(\phi)$ can be calculated as:

$$\frac{\partial V}{\partial \Omega} = 1 \tag{4.19}$$

by mixing last two derivatives, descent direction or normal velocity field can be determined as:

$$V_n = -\frac{\partial L}{\partial \Omega} = u_e^T k_e u_e - \lambda_v^k - \frac{1}{\Lambda_v^k}(V(\phi) - V^\star) \tag{4.20}$$

Topology Derivatives: The topological sensitivity of Lagrangian, Equation (4.16), can be achieved simply by using Equation (2.16) and it can finally be incorporated into Equation (2.18) to build the modified level set equation.

Compliant Mechanisms Design

A complaint mechanism is a monolith structure that transmits motion or energy from input to output port. Topology optimization can be employed as an automatic design procedure. In doing so, the objective function should be defined such that the optimum value of it leads to desired response of the mechanism. The most famous objective functions in this area are:

1. A weighted sum of mutual strain energy (MSE) and strain energy (SE), [121], [118]
2. Ratio of MSE to SE [121]
3. Geometric advantage (GA), mechanical advantage (MA) and work efficiency [120]

An extensive implementation of the last method has been performed by S. Chen et al. [120], [101]- [102]. In this series of works, geometric advantage has been formulated for level set method. It should be mentioned that different objective functions lead to different solutions [120]. In this thesis, a general formulation by Ananthasuresh is considered for the first approach and formulated for level set method. The detailed calculations of shape derivative and velocity fields are available in [120] for interested readers. A general objective function for compliant mechanism design is as follows [121]:

$$\text{Minimize } J = -\alpha\Upsilon(MSE) + (1-\alpha)\Theta(SE) \tag{4.21}$$

where Υ and Θ are monotonically increasing functions of SE and MSE. SE, is the strain energy and MSE is, mutual strain energy, and is defined as:

$$MSE = \int_\Omega \sigma_d \epsilon d\Omega = \int_\Omega E_{ijkl}(\phi)\varepsilon_{ij}(u)\varepsilon_{kj}(v)d\Omega = \delta_{out} \tag{4.22}$$

σ_d is stress field when only a unit dummy load is applied in the direction of the output displacement at output port. ϵ is strain tensor when only the actual force F_{in} is applied at input port. $\varepsilon_{ij}(u)$ and $\varepsilon_{kl}(v)$ are local strain when the input force F_{in} and unit dummy output force are applied to the structure separately. α is a positive multiplier. In discretized finite element fashion, one can represent it as:

$$MSE = \{u\}^T[K]\{v\} \tag{4.23}$$

in which $\{v\}$ is the displacement vector when only the unit dummy load is applied and $\{u\}$ is the displacement vector when actual load F_{in} is applied. Basically, $\{v\}$ and $\{u\}$ can be achieved by solving the following static equations:

$$\begin{gathered} [K]\{v\} = \{f\}_{\text{unit dummy load}} \\ [K]\{u\} = \{f\}_{\text{actual}} \end{gathered} \tag{4.24}$$

As can be seen in Equation (4.21), when J is minimized, MSE which is the deflection at output port [120] will be maximized (it has negative multiplier). At the same time, the second term in Equation (4.21) helps to have a stiff enough structure such that it can tolerate the applied forces.

Now the definition of cost function based on some measures of flexibility (MSE) and stiffness (SE) should be formulated for level set method. Let's assume J in Equation (4.21) as a function of ϕ.

$$\begin{aligned} J(\phi) &= -\alpha\Upsilon(\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u)\varepsilon_{kl}(u)d\Omega) \\ &+ (1-\alpha)\Theta(\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u)\varepsilon_{kl}(v)d\Omega) \end{aligned} \tag{4.25}$$

where MSE and SE are replaced by their definitions.

Shape Derivatives and Velocity Field To find the velocity field, one should take the shape derivative of J and write it as [120] :

$$D_{\Omega}J = -\alpha D_{_{SE}}\Upsilon D_{\Omega}(SE) + (1-\alpha)D_{_{MSE}}\Theta D_{\Omega}(MSE) \tag{4.26}$$

in which $D_{_{SE}}\Upsilon$ and $D_{_{MSE}}\Theta$ are derivative of Υ and Θ with respect to SE and MSE respectively. As Υ and Θ are two monotonically increasing functions with respect to SE and MSE respectively, both derivatives, $D_{_{SE}}\Upsilon$ and $D_{_{MSE}}\Theta$, are always positive. Equation (4.26) is derived by using chain rule. The next step is to calculate the shape derivative of strain energy and mutual strain energy to accomplish the process.

From mechanics of continua, it is known that:

$$\begin{aligned} SE_1 &= \textstyle\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(u_1)d\Omega = u_{1i} \\ MSE_1 &= \textstyle\int_{\Omega} E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(v)d\Omega = u_{1o} \end{aligned} \tag{4.27}$$

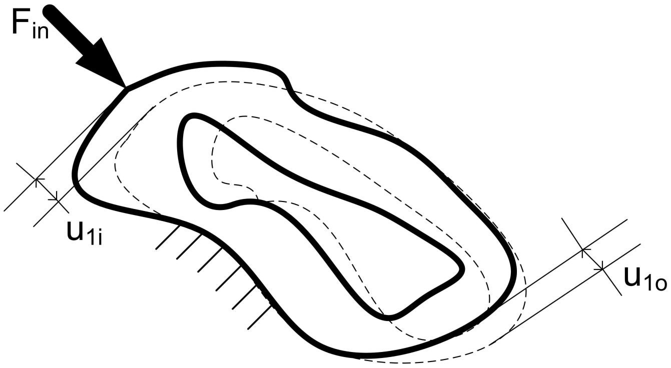


Figure 4.8: Deflections at input and output ports due to the external force

the index 1 in SE_1, $MSE_1$and u_1 are corresponding to the case in which input force is unit. Fig. 4.8 depicts the definition of u_{1i} and u_{1o} that are deflection at input and output ports respectively when the force F_{in} is applied at input port. Fig. 4.8 shows the definition of these parameters.

Shape Derivative u_{1i}**:** The derivation of the shape derivatives for u_{1i} is explained in [120] and final results are as follows:

$$\begin{aligned} D_\Omega u_{1i} = 2\{&\textstyle\int_{\Gamma_{N1}}[\frac{\partial(g.u_1)}{\partial n} + \kappa g.u_1]V_n ds + \int_\Gamma f_b.u_1 V_n ds\} \\ &- \textstyle\int_\Gamma E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(u_1)V_n ds \end{aligned} \tag{4.28}$$

If the body force f_b and length of Neumann boundary condition are negligible, only the last term is considered.

Shape Derivative u_{1o} :

$$D_\Omega u_{1o} = \frac{\partial l_1(v)}{\partial\Omega} + \frac{\partial l_2(u_1)}{\partial\Omega} - \frac{\partial a(u_1,v)}{\partial\Omega} \tag{4.29}$$

in which

$$\frac{\partial l_1(v)}{\partial\Omega} = \int_{\Gamma_{N1}}[\frac{\partial(g_1.v)}{\partial n} + \kappa g_1.v]V_n ds + \int_\Gamma f_{b1}\cdot vV_n ds \tag{4.30}$$

$$\frac{\partial l_2(u_1)}{\partial\Omega} = \int_{\Gamma_{N2}}[\frac{\partial(g_2.u_1)}{\partial n} + \kappa g_2.u_1]V_n ds + \int_\Gamma f_{b2}\cdot u_1 V_n ds \tag{4.31}$$

$$\frac{\partial a(u_1, v)}{\partial \Omega} = \int_{\Gamma} E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(v)V_n ds \tag{4.32}$$

similar to the previous shape derivative for traction free boundary condition, only the last term will be used in practical cases.

Setting The Normal Velocity Field V_n: To optimize the objective function naturally, one can use steepest descent optimization method by substituting $V_n = -P$ in Hamilton-Jacobi Equation (2.13), where P can be defined in the following equation [120].

$$D_{\Omega}J = \oint_{\partial\Omega} PV_n d\Omega \tag{4.33}$$

Comparing this equation with shape derivative of cost function J, Equation (4.26), one can conclude that:

$$\begin{aligned} P &= -\alpha(D_{_{SE}}\Upsilon)E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(u_1) \\ &+ (1-\alpha)(D_{_{MSE}}\Theta)E_{ijkl}(\phi)\varepsilon_{ij}(u_1)\varepsilon_{kl}(v) \end{aligned} \tag{4.34}$$

Topology Derivatives: Similar to the compliance minimization problem as long as the normal velocity V_n is available, one can use Equation (2.18) to incorporate topology derivative into the level set method. Now, Equation (4.35) calculates P defined as: $P = -V_n$, so again employing Equation (2.18), the Complaint mechanism design problem can be formulated and solved for different boundary conditions. Section 4.2.2 is associated to this problem.

4.2.2 Results of Topology Optimization

In this section, the developed algorithms has been implemented and compared with classic level set theory over a set of benchmark problems. In the first part, local solutions and their dependency on initial guess in level set method (including both shape and topology derivatives) are investigated. This basically shows the importance of the contributions of this thesis that tries to eliminate the dependency of algorithm on initial guesses. In the second part, the results of ACSF will be discussed and finally MSLSM is applied over the same set of benchmark problems.

Local Solutions of Level Set Method

Based on the literature, topology derivatives are suggested in order to reduce the number of local optima. Here the formulation, Equation (2.17), based on both shape and topology derivatives is employed. As expected, local solutions are still a significant issue of the algorithm. Table 4.5 shows the results of the "Bridge Problem" for different initial guesses. Topology and shape derivatives are combinations with the same weight, $w = 1$. The simulation is performed over a grid of 150×75 in which both lower corners are fixed and a unit load is applied downward. It has been simulated for 100 iterations. Obviously if the convergence criterion is met does the process stop before 100 iterations.

As can be seen topology, derivative is effectively working, for example in the first simulation that structure has no hole in initial guess, the algorithm can nucleate some holes, caused by topology derivative term. Obviously the existing hole in structure still can combine together to make bigger holes if needed.

Results of ACSF Modified by Annealing Scheme

In this section ACSF algorithm is implemented and its results for mean compliance minimization problem (Bridge Problem) and complaint mechanisms design are presented and investigated.

Mean Compliance Minimization Problem (Bridge problem) Using ACSF For classic problem of mean compliance of bridge problem, a grid of 100×100 is created based on which both structural and PDE analysis is performed. To reduce the unnecessary complexity of model, a symmetric model is considered. The ACSF algorithm utilized by annealing scheme is implemented and simulation is performed 20 times to ensure whether the final solution is unique. In this set of simulations, 90% of cases (18 times) converge to the same topology and only 2 times the final topologies are different. The final shape of the 18 solutions with the same topology do not have exactly the same shape, however the objective functions are almost the same. In other words, although the shape (and not topology) is not the same, the algorithm converges to values that are slightly different. Therefore, as the algorithm only works based on the objective function, and it is really difficult for it to distinguish between different shapes. Table 4.6 shows the 5 most different final shapes (out of 18) that have almost the same topologies and cost functions. Fig. 4.2.2 depicts the convergence patterns of each shape shown in Table 4.6. As can be seen, final values in all 5 simulations are almost the same but they have totally different path to reach

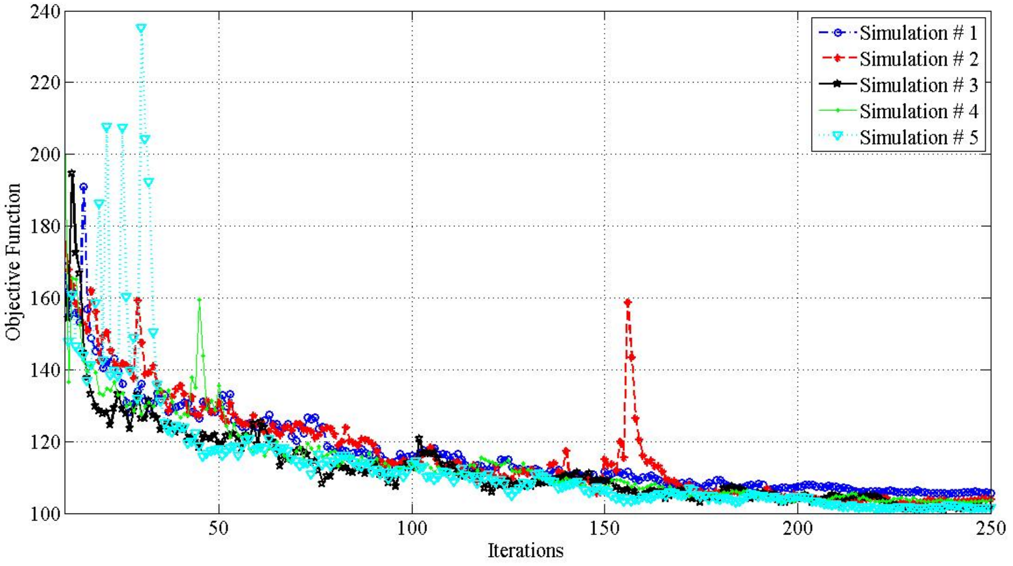


Figure 4.9: Convergence of the ACSF algorithm with annealing scheme for the Bridge problem shown in Table 4.6

this final value. One can conclude that the simulation # 3 is still a local solution of the topology in the simulation #1 because of the thin element at the middle-down point of the structure. It technically says that depending upon the solidification rate τ in Equation (3.66), ACSF might be unable to escape from local fronts, resulting in unnecessary parts in the final solution. The interesting point about ACSF is that there is absolutely no dependency on initial guess, since the initial guess for all simulations are selected randomly, and as seen, most of them are converging to the same topologies.

Complaint Mechanisms Design Using ACSF Using the Equation (4.26)-(4.35) presented in Section 4.2.1, compliant mechanisms design problem is investigated for some case studies. For each case, classic level set method is applied starting from four different initial guesses and then results are compared to the results of ACSF. Generally, classic level set method is notably dependent on initial guesses while ACSF almost always converges to the same topologies. It should be noted that, similar to the Bridge problem, sometimes the shapes are different in terms of length and width of arms but usually the process approaches the same topologies and almost the same objective functions. Employing ACSF, the pattern of convergence for all of the case studies is similar to Fig. 4.2.2 showing the convergence of ACSF for Bridge problem. To have a fair comparison, in this section, ACSF is run only 1 time for each initial condition and the most frequent topology is reported in the third column of the Tables 4.7-4.9. Obviously if ACSF is allowed to be run for more times, it is capable to find the same topologies and also shapes for all initial guesses.

Case Study 1: The boundary conditions along with the input and output of a force inventor mechanism are depicted in Fig.4.10(a).

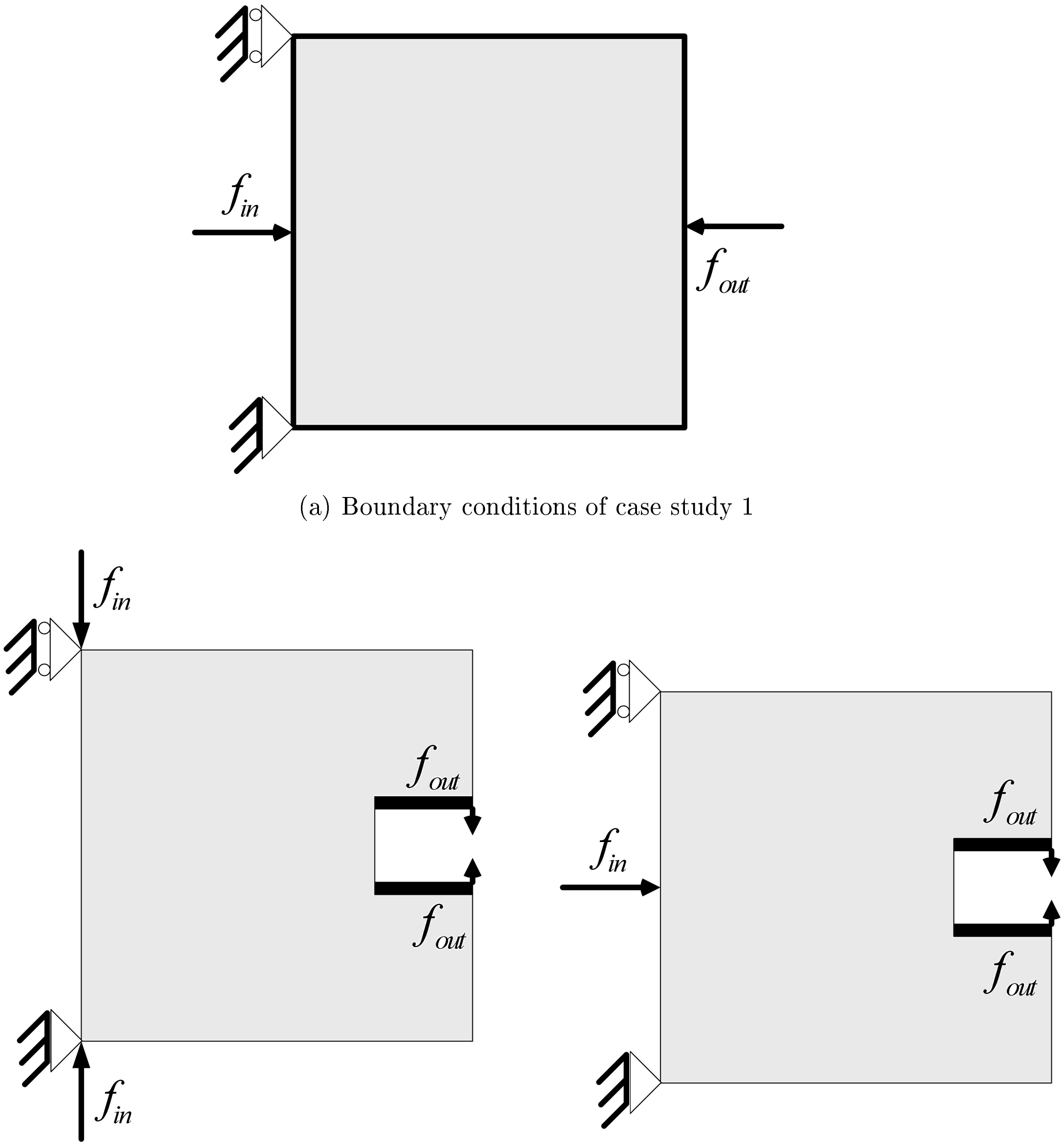
The results of classic level set method with different initial guesses are shown in Table 4.7. The first column shows the initial guesses and the second one shows the final results of classic level set method and finally last column shows the results of ACSF algorithm. To be fair, ACSF algorithm is run only one time and reported in the third column. Obviously if the ACSF algorithm runs more than one time, it is very probable that it would find exactly the same solution for last column. Patterns of convergence for all of these 10 simulations are depicted in Fig. 4.11. As can be seen, the final values of objective functions are almost the same, which is predictable based on the results in Table 4.7 because the final solutions are almost the same except for simulation 4 for classic level set method.

Case Study 2: In this study, the mechanism shown in Fig. 4.10(b) is considered. Again, to represent the mechanism boundary conditions are depicted.

Dependency of classics level set method and ACSF on initial guesses for this mechanism is investigated in Table 4.8. Convergence graph for this case study is shown in Fig. 4.12. One can clearly see that the objective functions for ACSF algorithms are less than classic level set method. Another point that is represented in this figure is, simulations based on classic level set method are stuck and do not proceed after some iterations. At the same time, the simulations of ACSF algorithm converge to the almost the same objective function, which is compatible with almost the same topologies in last column of Table 4.8.

Case Study 3: For the last case study, boundary conditions and correspondingly the role of mechanism has changed and shown in Fig. 4.10(c).

The performance of the ACSF algorithm is compared with classic level set method in Table 4.9. Similar to the previous case studies, both classic level set method and ACSF are compared for different initial guesses. Results are shown in Table 4.9 and Fig. 4.13. The preliminary conclusion is the same as case study 2.



(a) Boundary conditions of case study 1

(b) Boundary conditions of case study 2

(c) Boundary conditions of case study 3

Figure 4.10: Definition of Boundary Conditions

4.2.3 Conclusion of Topology Optimization

Ignoring some differences in the final solutions of ACSF algorithm, it seems that this algorithm is a promising approach to mitigate the dependency of the level set method on initial guess. The results that are obtained and presented in Tables 4.7-4.9 confirm this claim. Investigating the aforementioned results, it can be seen that sometimes when the ACSF is applied, the values of the objective functions for each case study are almost the same, although the corresponding shapes and topologies have some discrepancies. In other words, although the objective function which is the only criteria to distinguish between different solutions is almost the same, the corresponding shapes and topologies are not the same. To cope with this difficulty one should add some other terms to the objective function such that the optimization procedure can make difference between different design. These extra terms could be mathematical models for manufacturability, limitations on stress and durability of the structure, in addition to robustness of the design against geometrical and loading uncertainties during fabrication and life cycle of the product respectively.

It is worth mentioning that, as known, usually stochastic optimization methods, especially those one that are derivative-free, are very time consuming compared to gradient based approaches. ACSF, as a stochastic gradient-based method takes the advantages of both, hence, it is capable of finding global solution (stochastic methods) along with fast convergence (gradient-based approaches). Looking at the results, Fig. 4.11-4.13, one will notice that there is not a huge difference between the convergence time of ACSF and classic level set method, although ACSF always takes more time to converge (maximum 2-3 times of the simulation time for classic level set method). Obviously, in this formulation still shape and topology derivative are required in the deterministic model of Equation 3.66, namely $G(D^2\phi, D\phi, x, t)$.

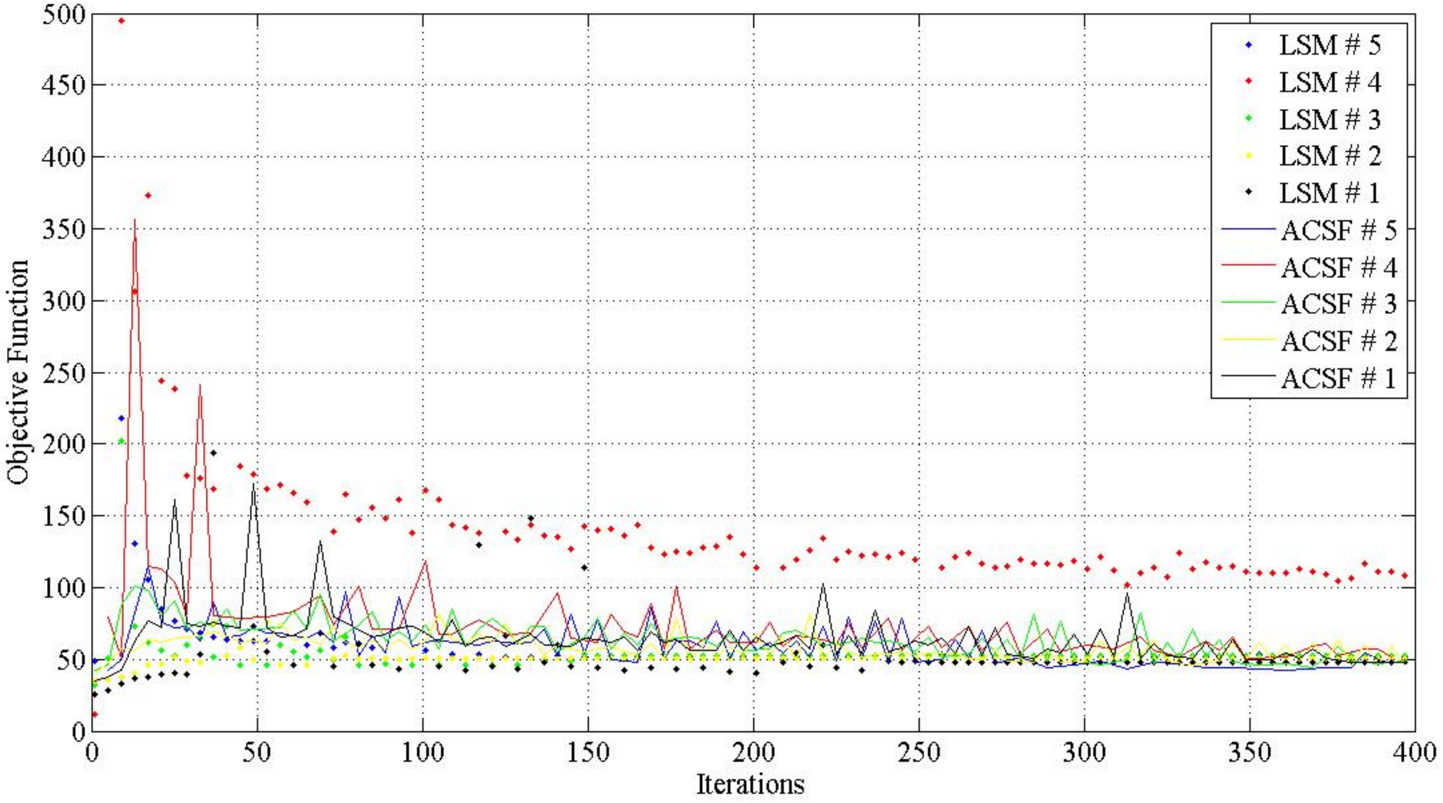


Figure 4.11: Convergence pattern of ACSF for case study 1

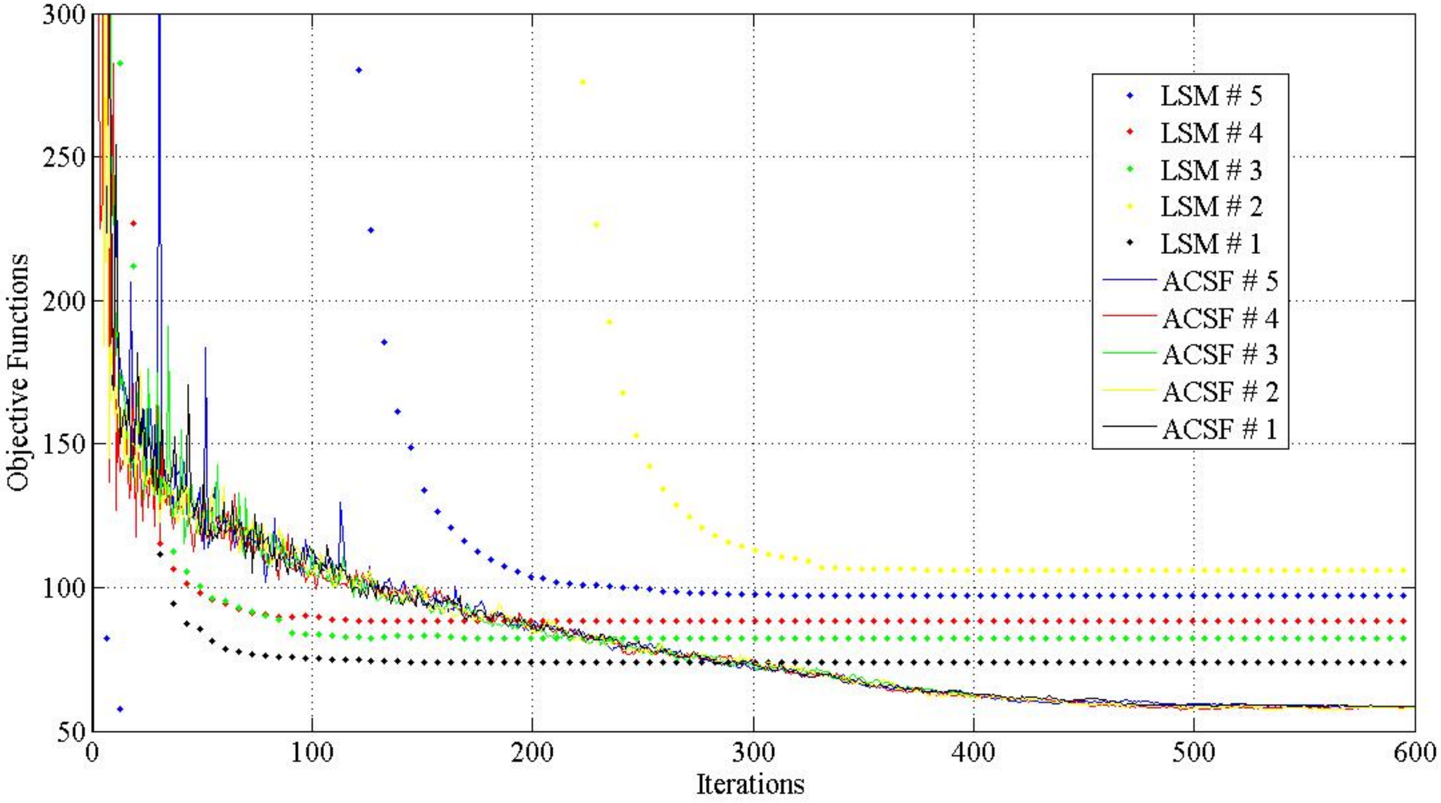


Figure 4.12: Convergence pattern of ACSF for case study 2

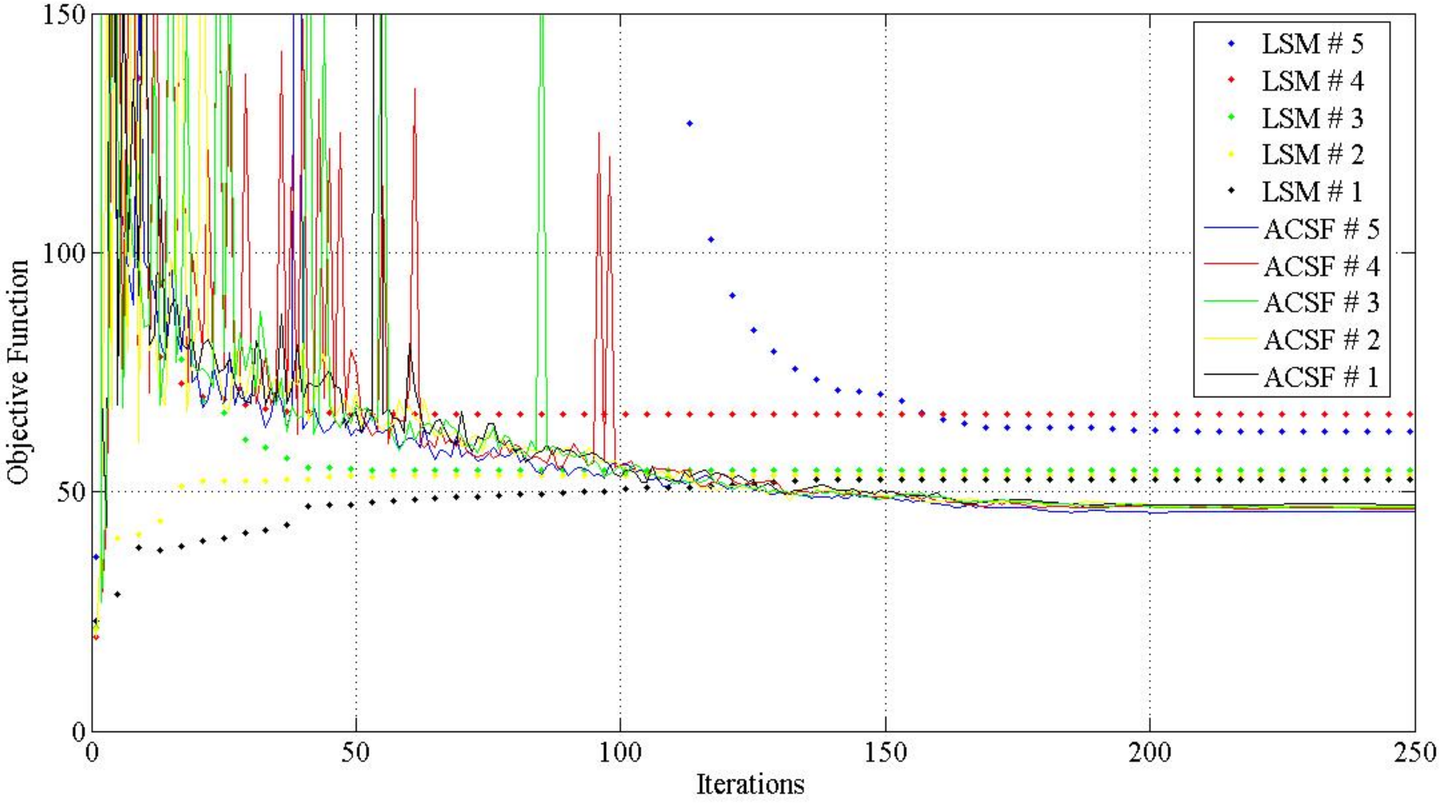


Figure 4.13: Convergence pattern of ACSF for case study 3

Table 4.5: Dependency of level set method on initial guess in Bridge problem

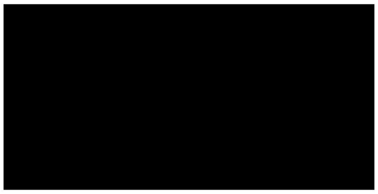
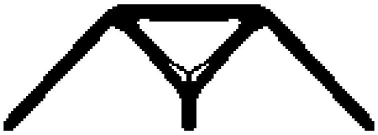
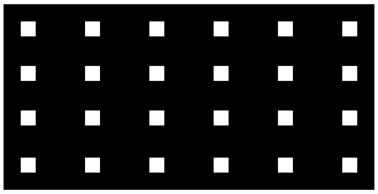
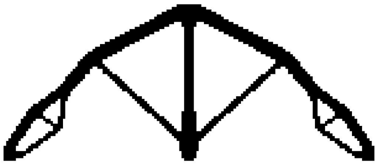
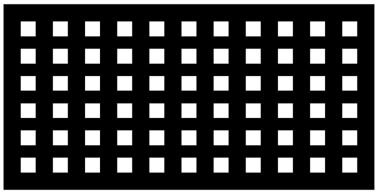
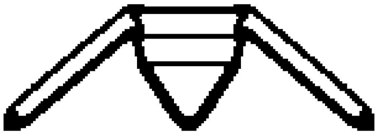
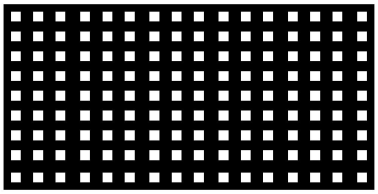
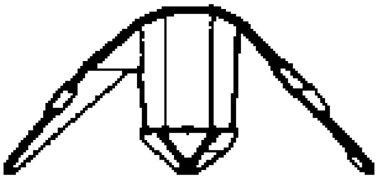
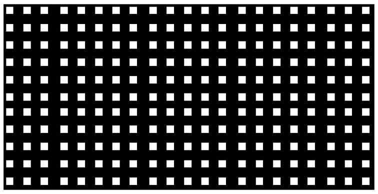
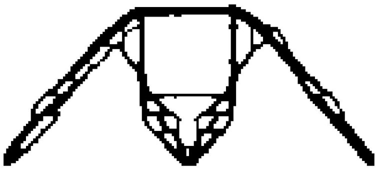
Number	Initial guess	Final solution
1		
2		
3		
4		
5		

Table 4.6: The 5 most different shapes (out of 18 simulations) with the same topologies

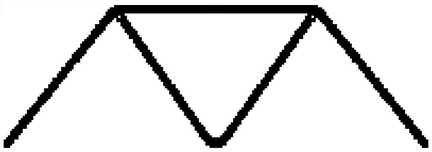
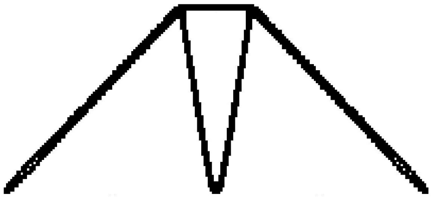
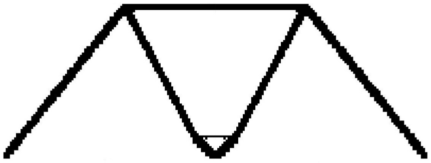
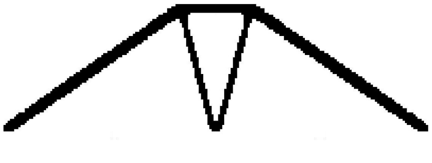
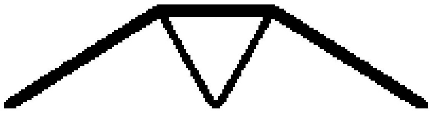
Simulation #	Shape
1	
2	
3	
4	
5	

Table 4.7: Comparison of LSM and ACSF with different initial guesses for case study 1

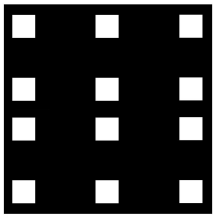
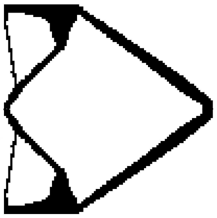
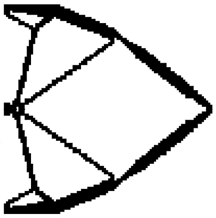
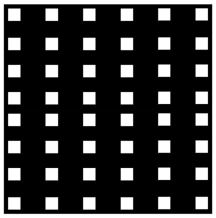
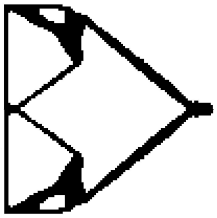
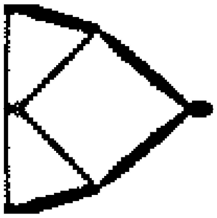
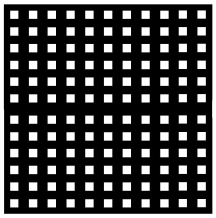
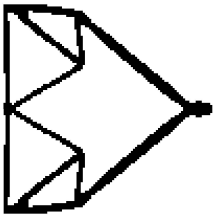
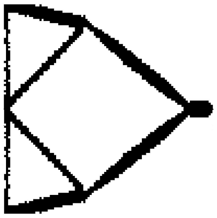
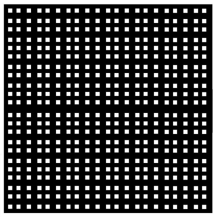
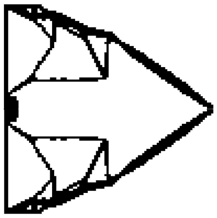
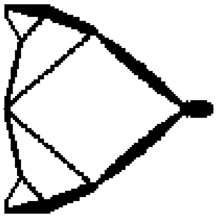
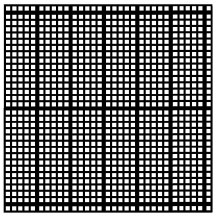
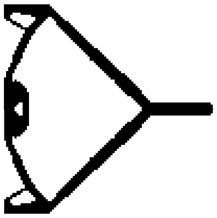
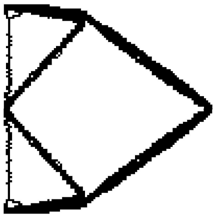
Initial Guess	Classic Level Set Method	ACSF Solution
		
		
		
		
		

Table 4.8: Comparison of LSM and ACSF with different initial guesses for case study 2

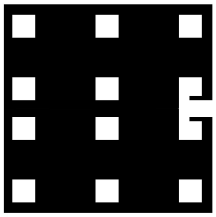
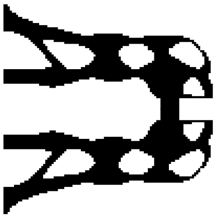
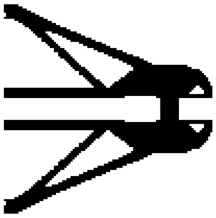
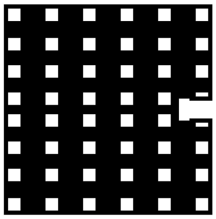
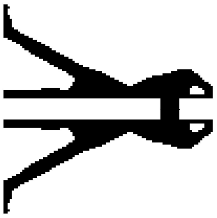
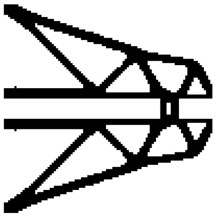
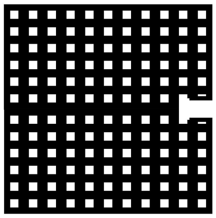
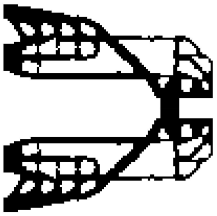
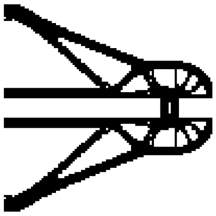
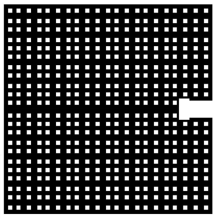
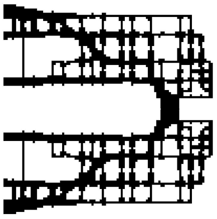
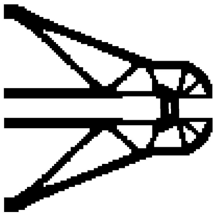
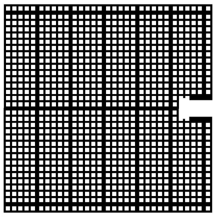
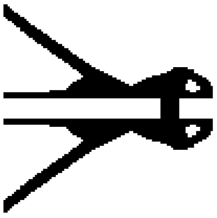
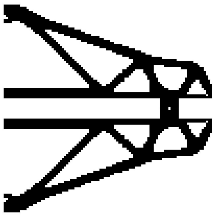
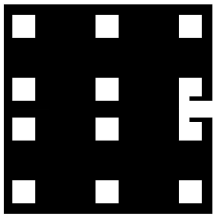
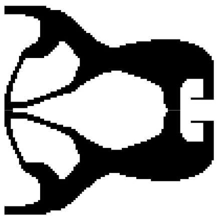
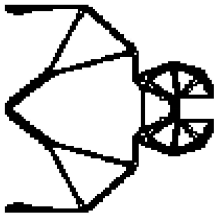
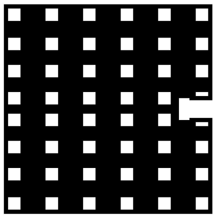
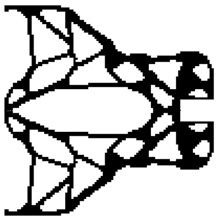
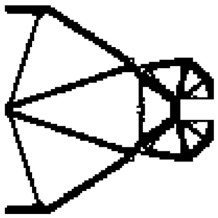
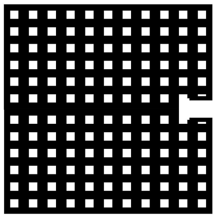
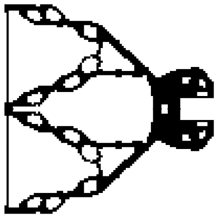
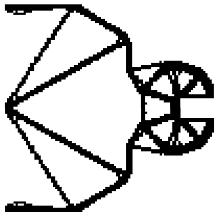
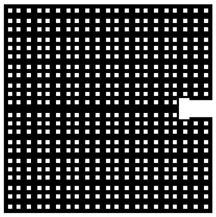
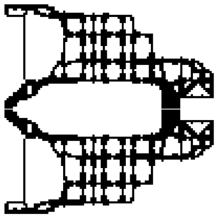
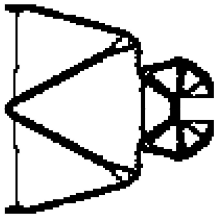
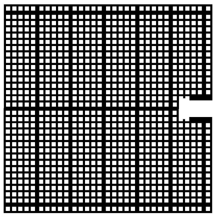
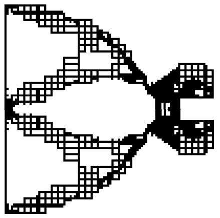
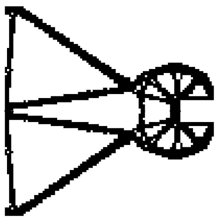
Initial Guess	Classic Level Set Method	ACSF Solution
		
		
		
		
		

Table 4.9: Comparison of LSM and ACSF with different initial guesses for case study 3

Initial Guess	Classic Level Set Method	ACSF Solution
		
		
		
		
		

4.3 Application of Stochastic Flows in Optimal Path Planning Problem

Inspired by electric arc discharge, the Evolving-Arc (E-ARC) algorithm is proposed in current work in order to deal with serious difficulties. Technically, E-ARC employs the stochastic level set method [57] in open curve fashion [138]. Improving efficiency, and balancing accuracy and computational costs of the method, a flexible structure is suggested that allows a compromise between conflicting targets, depending upon the availability of computational resources and the required precision. This section proceeds as follows: a brief explanation on the physics of electric arc discharge and its analogy with optimal path planning are presented in Section 4.3.1). The E-ARC algorithm along with simulated annealing are described and compared in Section (4.3.2), followed by results (4.3.3) and conclusions (4.3.4).

4.3.1 Electric Discharge and Analogy with Path Planning Problem

In any electric discharge the electricity flows through a path with minimum resistance [139]-[141]. Electric discharge has two main examples. The first is lightning; the second is electric arc. Lightning as a wonderful natural phenomenon is a transient electric discharge between thunderclouds and ground surface. The main difference between lightning and electric arc is that lightning is a very brief phenomenon, while the electric arc lasts until the power supply is connected. Electric arc is a luminous streamer, which is an optimum path in terms of conductivity, but it can evolve as the environmental variables change while time passes. For example, even the convection can result in changing the minimum resistance path [140]. Just as with lightning strokes, sometimes multiple paths are required to relieve the potential charge. Figure (4.14) shows an electric arc including several strokes. As can be seen, one of these strokes is more enlightening than the other, which basically has the best conductivity amongst paths shown.

Depending on environmental variables, an electric arc evolves in order to find the optimum path with maximum conductivity. Obviously, as long as the variables do not change, the optimum steamer settles down in a stationary condition, but usually they change in a stochastic fashion. As a result, one can see that an electric arc evolves randomly [139], and consequently is capable of finding a better path of conductivity. In an optimal path planning problem, one is looking for a path that has the best performance. Depending on how complicated the performance is defined, one should choose an algorithm resulting in

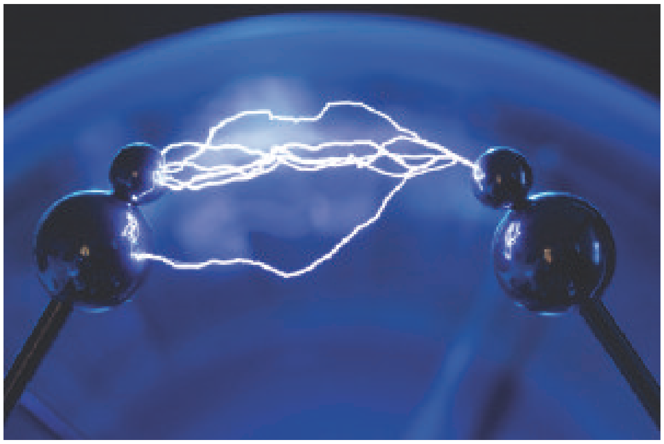


Figure 4.14: A Real Electrical Arc, image courtesy of Jonny O'Callaghan [142]

a best possible solution for the problem. The present algorithm in this section, is inspired by electric arc evolution and designed to find such a path. Optimality of the path and the stochastic behavior of the evolution is governed by using open curve [138] stochastic level set method [57].

4.3.2 Algorithm

Open Curve Level Set Method

Usually the level set method works in cases where implicit closed curves are required. Image segmentation and topology optimization are some examples of this purpose. In a path planning problem, an open curve should be evolved to find the optimal path. As there is no robust way to employ an open curve as a zero level set of a continuous function, P. Smereka [138] has presented a modified level set approach, usable for both open and closed curves. The interacting spiral crystal growth has been modeled by this approach [138]. Modeling the crystal growth needs open curve to define the screw dislocations in crystal. The classic level set method with one evolving surface has been modified by adding another level set function in order to include screw dislocations. In Figure (4.15), the initial step line is defined by:

$$\Gamma(0) = \{\phi(x,0) = 0, \psi(x,0) > 0\} \tag{4.35}$$

Therefore, the middle part of the horizontal line $\Gamma(0)$ intersected by two vertical lines

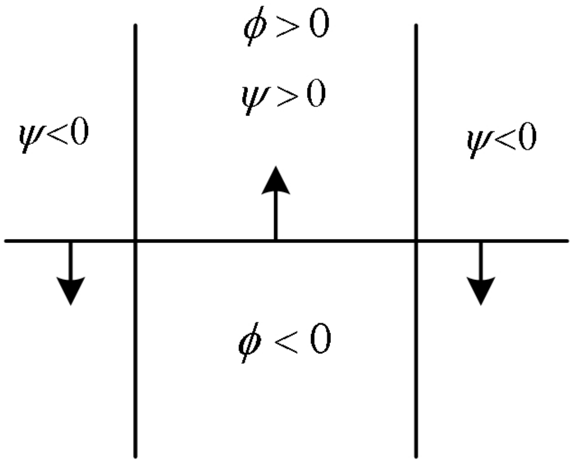


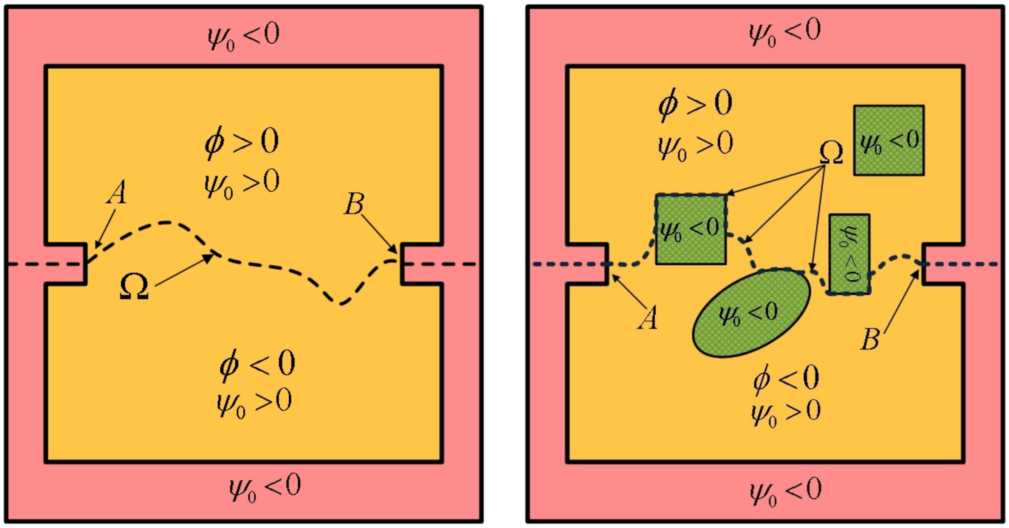
Figure 4.15: The way to implement single level set

is the open curve required for spiral crystal growth modeling. In this thesis almost the same approach has been employed, except that in our case there is no need to update the second level set and it is only used to show the poles, boundaries and obstacles.

Problem Setup

Before struggling with the main problem, it is necessary to discover how the path, boundaries, hazardous regions and obstacles are defined. As explained, the path should be represented as an open curve level set [138], and by using almost the same idea, the boundaries and hazardous regions and/or obstacles will be implemented.

Definition of Path According to Section (4.3.2), a path Ω can be defined by an evolving surface ϕ and a stationary surface ψ_0. This open curve, Ω, is shown in Figure (4.16(a)). Here, only ϕ is evolving, and the fixed surface ψ_0 is employed only in order to describe the two ends of the path and ϕ can evolve in those regions that $\psi_0 > 0$. The path evolves based on the algorithms that will be explained in detail in the following sections, but two ends of the path (A and B) will not move, because they are located in the regions that $\psi_0 \leq 0$.



(a) Open Curve As a Simple Path

(b) Hazardous Regions and/or Obstacles Defined in E-ARC

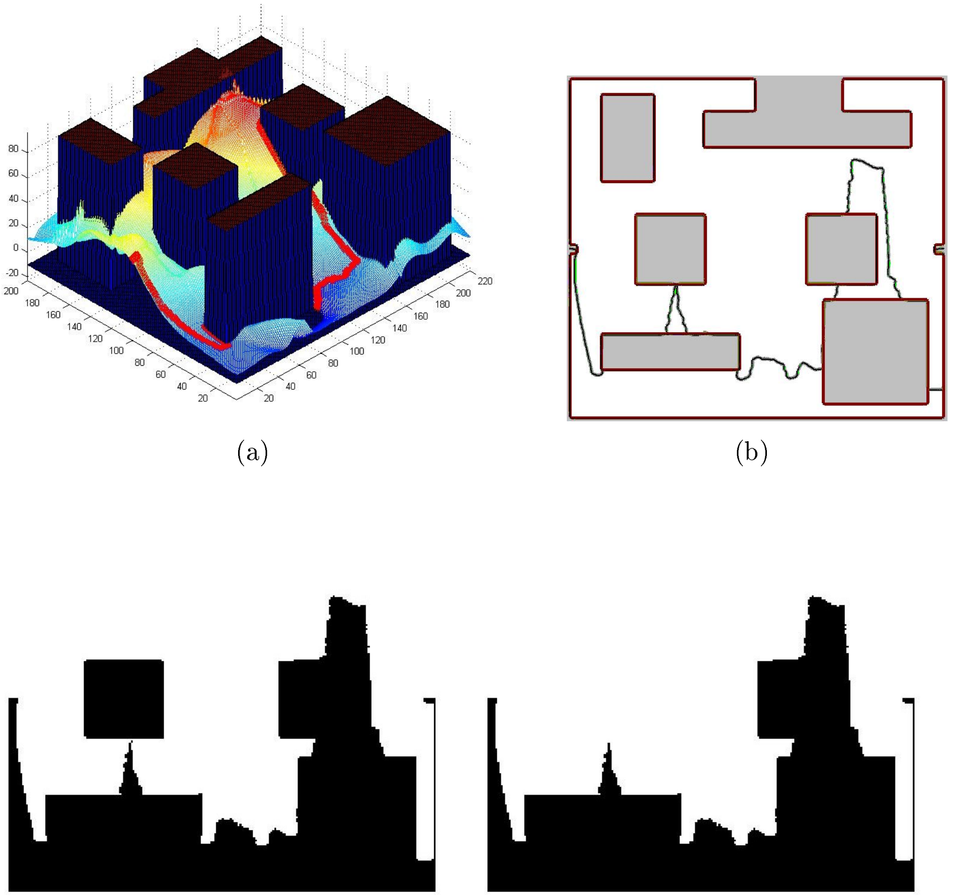
Figure 4.16: Open Curve Definition of Path

Definition of Boundaries To define an evolving open level set, one can employ another fixed level set, as in Figure (4.15). Now the problem becomes how to define the boundaries for the region in which the path should evolve. Figure (4.16(a)) shows the idea to define proper boundaries for the problem. The idea is similar to the way that the poles are defined, in that the curve evolves within the region that is defined by the boundary, and gets frozen outside it. By adding this extra constraint, the path looks like Figure (4.16(a)).

Hazardous Regions and/or Obstacles The standard way to implement hazardous regions and/or obstacles is to penalize those paths that are passing over these regions. Thus, the optimization algorithm neglects these regions in order to minimize the objective function. To penalize the path, the objective function should be magnified by some multipliers over hazardous regions and/or obstacles. Another valid approach is to utilize the method introduced in Section (4.3.2) to define the open path. It has been noted that to define the evolving curve that defines the path, one should employ another level set that does not evolve and is only used in order to define the start and end of the path. This idea extends to define the hazardous area and obstacles encountered over the path. Trying out the idea, assume that the hazardous areas and obstacles are marked by the same level set employed to define the ends of the path. In this definition, while the path is evolving within regular regions, it cannot evolve in marked areas because the updated path gets frozen in aforementioned regions. Consequently, the evolving path will not enter this area and only evolves in regular regions to optimize the objective function. Figure (4.16(b)) depicts this idea. As evident, while the path is evolving, it cannot enter the signed area $\psi < 0$. The potential problem arises when the evolving procedure completely surrounds the hazardous region and the path cross each other: the surrounding part of path becomes unnecessary and should be removed. It cannot be obtained automatically by classic formulation. To do so, the Flood Fill algorithm, a popular algorithm in image processing, [144] is employed to remove the undesired holes in the image. Considering Equation 2.1, one can conceive the entire region as a black and white image in which the black and white regions are defined by the evolving curve. When the curve crosses itself, Figure (4.17(b)), a hole will be created in the aforementioned image. Using the image processing algorithms such as the Flood Fill algorithm [144], one can remove these holes and mitigate the unnecessary part of the path that surrounds the hazardous regions or obstacles. After removing the hole(s), the evolution of the path can proceed as usual.

Description of the Evolving Algorithms

Basically, the E-ARC algorithm employs stochastic level set approach with open curve fashion. In addition, some modifications are required in order to improve the efficiency of the algorithm. In this thesis, three steps are designed: first, the classic level set method with open curve fashion will be explained. This case is only practically applicable when the objective function has a closed form equation as a differentiable function of the path. In most engineering applications this closed form equation is not available and hence this case is not studied in detail and only a general formulation required for further steps is presented. Instead another approach, which is very similar to simulated annealing in variational framework [57], is studied. This second approach can be employed either in a situation where the closed form equation exists or the path can be evaluated numerically. Eventually the algorithm will be modified to overcome some major drawbacks, such as slow convergence and lack of accuracy in some part of the path. As stated, in engineering applications, the objective function is usually so complex that a closed form equation does not exist, or at most could be evaluated numerically. The third method is also usable for both circumstances, and this is the main contribution of the current work. This method is called "E-ARC" for "Evolutionary Arc," but can also recall "Electric Arc," which inspired the algorithm.



(a)

(b)

(c)

(d)

Figure 4.17: Results of Flood Fill Algorithm to Remove the Unnecessary Parts of the Path

Open Curve Classic Level Set Method As mentioned, this method is practically confined to a problem that there exists a closed form objective function which is differentiable with respect to the path. Although in many cases it is very restricted, it nevertheless results in fast convergence of the optimization process as the gradient is analytically available at each point of the path which leads the path to evolve toward the optimal path. To formulate this problem, assume that path Ω is defined between two ends A and B, by the evolving surface ϕ and stationary functional ψ_0 as shown in Figure (4.16(a)). Suppose the objective function $J(\phi)$ defined as:

$$J(\phi) = \int_{\Gamma} f(\phi)\delta(\phi)H(\psi_0)d\Gamma + \lambda \int_{\Gamma} \delta(\phi)H(\psi_0)d\Gamma \tag{4.36}$$

where Γ is the domain over which ϕ and ψ_0 are defined. $f(\phi)$ is called local energy density function, by which the objective function is represented all over the path Ω and should be evaluated at each point on the path. $\lambda > 0$ is a positive multiplier. Obviously, the second term represents the length of the path. Again, it should be noted that ψ_0 is a fixed surface and the main variable is only ϕ. $H(\psi_0)$ is employed in order to freeze the region that the path is forbidden to enter. Because along as ψ_0 is positive $H(\psi_0)$ would be equal to 1 and the path evolves along the negative direction of gradient and on the other hand, when ψ_0 is negative, $H(\psi_0)$ would be 0 which means that the path can not evolve any more. The subscript "0" in ψ_0 is chosen to show the stationary status of the surface ψ, as this surface would not evolve during the optimization process and only is employed to define the path, boundaries and/or obstacles.

The optimal path planning problem can be presented as:

$$\min_{\phi} J(\phi) \tag{4.37}$$

Usually by solving the following Euler-Lagrange equation, this minimization problem can be solved [54]- [53].

$$\phi_t = \nabla_{\phi} J \tag{4.38}$$

By a little bit of mathematical manipulation:

$$\phi_t - \nabla\phi \cdot \frac{\partial J}{\partial \phi} = 0 \tag{4.39}$$

which is exactly the Hamilton-Jacobi Equation 4.1 in which V_n is replaced by $-\frac{\partial J}{\partial \phi}$. Therefore, Equation 4.39 should be solved in order to find the optimal path. Notice that not only $\frac{\partial J}{\partial \phi}$ should exist but also it is desirable to have a closed form equation for $\frac{\partial J}{\partial \phi}$; otherwise, a huge amount of numerical computation is required. This restriction limits the real application of the problem dramatically, and therefore one only uses this formulation as a basis for the next step.

E-ARC Algorithm When analytical gradient does not exist, one may encounter two major problems. First, numerical calculation of the gradient requires significant computational resources; and second, if only the stochastic terms are used (for simulated annealing approach), some problems with convergence will arise and usually the final solution is not accurate enough in some parts. This problem arises because in stochastic gradient systems like Equation 3.62, the process converges to the solution of deterministic system [148], [149]. That is, the system has a pattern to follow towards the final solution, although the stochastic terms help to escape from local points, and the final solution is likely the global solution or at least the best possible one. In some cases when the gradient does not exist, the process should jump from one state to another in hope of finding a better solution which has a smaller probability compared to cases where gradient is available. On the other hand, because these jumps are totally random, even if a part of the optimal path is found it is possible to be lost in the next step as the simulated annealing only cares about the global value of the objective function and not the local accuracy of the path. To deal with these drawbacks, E-ARC algorithm compiles the advantages of both local and global path evaluation. In doing so, at each step N_c perturbed paths, namely $C_1, ..., C_{N_c}$, will be generated instead of only 1, Figure (4.18). Next, the straight line that connects two ends of the path is divided to N_n parts $(L_1, L_2, ..., L_{N_n})$ with random lengths in such a way that the summation of all segments is equal to the length of the original connecting line. $N_n + 1$ vertical lines are drawn such that N_c created paths are also divided into N_n parts, Figure (4.18). The random generation of all segments (each L_i) are repeated at each time interval. From there, each part of these paths should be evaluated individually. For each part of the path (L_1, L_2, ..., L_{N_n}) the best of the segmented paths (each of C_1, C_2, ..., C_{N_c}) are selected based on their fitness values. Then, the new path is made of the combination of these bests. In other words, the updated path for this step is created by connection of the best segments for each division of paths. Figure (4.18) describes how the new path is defined at each time step. The bold line in each division of Figure (4.18) is assumed to be the best one. It might raise the question that the new path has discontinuity as the best segments are not necessarily along each other, similar to the path in Figure (4.18). One should pay attention that the perturbed paths are indeed very close to each other as the step time is selected small enough and what is depicted in Figure (4.18), is magnified just to clarify the concept of the selection process. Besides, the re-initialization of the level set method makes the path continuous, so this discontinuity would not result in divergence as long as the time step is small enough. Although numerical simulation has shown that the time step necessary to satisfy the "$CourantFriedrichsLewy$" (CFL) condition [54], [53] is also suitable for this purpose, there is no mathematical proof available yet as a standard guideline. The results of this algorithm are presented in Section(4.3.3), and confirm its efficiency. To summarize, the algorithm is represented in Table (4.10) as a pseudo code.

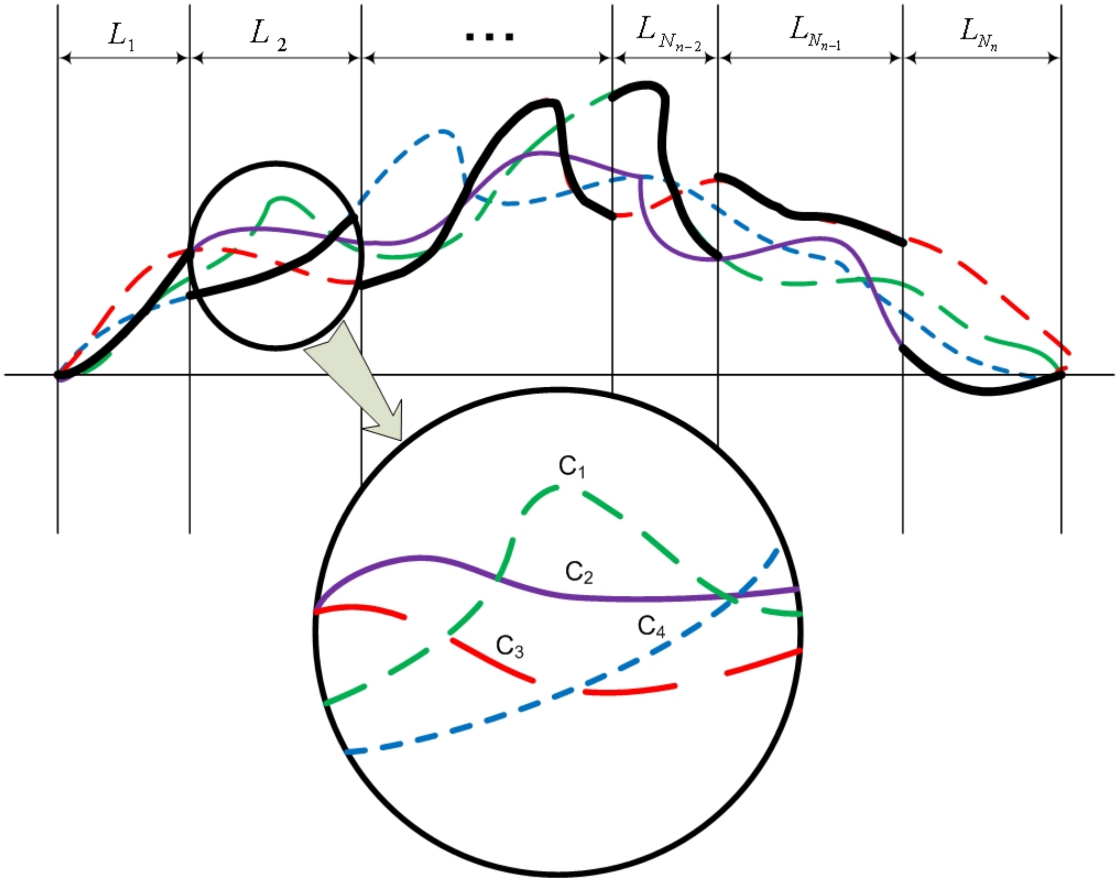


Figure 4.18: Group Gradient Concept in E-ARC

Table 4.10: Pseudo Code for E-ARC Algorithm

```
Step 1  Set $N_n$, $N_c$, $T_0$, define $\psi_0$, stopping criteria and $i = 1...N_{Itr}$

Step 2  Start with initial guess $\phi_1$ and evaluate $J(\phi_1)$

Step 3  1-If analytical gradient exists $V_{n,i} = -(\frac{\partial J}{\partial \phi})_i$
            Create $\Theta_i^j$s using
            $d\Theta_i^j = (V_{n,i}|\nabla \phi_i|dt + |\nabla \phi_i| \circ dW_i^j(t))H(\psi_0)$
            $j = 1...N_c$
          Else
            Create $\Theta_i^j$s using
            $d\Theta_i^j = (|\nabla \phi| \circ dW_i^j(t))H(\psi_0)$
            $j = 1...N_c$

        2-Divide the straight connecting line into $N_n$ random segments

        3-Divide each $\Theta_i^j, j = 1...N_c$ based on the lengths in Step (3-2)

        4-Evaluate each of the segments in step (3-3)

        5-Select the best segment of each path $\Theta_i^j$

        6-Combine the best segments to make a new path $\phi_{i+1}^\star$

        7-Update $\phi_{i+1}$by re-initialization of the $\phi_{i+1}^\star$

Step 4  Calculate $J(\phi_{i+1})$
```

Step 5	If J(ϕ_{i+1}) < J(ϕ_i)$\Longrightarrow$	Accept ϕ_{i+1}
	Else $\Longrightarrow$	Accept $\phi_{i+1}$with probability $P_{acc} = \exp(-\frac{J(\phi_{i+1})-J(\phi_i)}{T(i)})$
Step 6	Update T(i) and loop back to fulfill the stopping criteria	

4.3.3 Results of Path Planning Problem

In this section, only the general case of the algorithm shown in Table (4.10) in which the analytical gradient does not exist is investigated. First, it is shown that the results for simulated annealing algorithm, which basically illustrate the drawbacks in this algorithm for optimal path planning problem. Then, we apply the E-ARC algorithm, and show how the gradient at each point is replaced by a group motion of the points lying on a segment of a path; this method is an extension of gradient-based methods. When the gradient exists, the curve evolves along the steepest descending direction, which is opposite to the gradient vector's direction.

In E-ARC, each segment is chosen as the best one between all N_c generated segments in a particular section (one of the 1 to N_n sections). In other words, the best segment is the one that creates a higher reduction in the objective function similar to the gradient-based approach that leads each point along the steepest descending direction. If we choose N_n such that the number of generated intervals is equal to the number of nodes over each path, the two approaches become almost identical. Notably, E-ARC shows that when converging to the optimal path, there is no need to calculate the best direction for every single point over the path, and that only a limited number of intervals result in a significant improvement. Therefore, the computational cost decreases remarkably compared to calculating the gradient numerically at each point. A benchmark problem is designated in order to evaluate the algorithms. It is created by combinations of several bell shape Gaussian functions with randomly selected heights and alignments. This bumpy surface is chosen as an arbitrary objective function, and two fixed points over this surface are selected as two ends of the desired path. To have a comparable condition, all optimization processes starts from the same initial guess. The results for simulated annealing algorithm are presented in Figure (4.19). Based on these set of results, there is no guarantee to converge to the same solution even if the simulation starts from the same initial guesses, which is a serious flaw. Moreover, the solution's accuracy is unacceptable in some parts of the path. The result indicates that each path could find some part of the optimal path accurately, but the whole optimal path is not discovered. This means that the simulated annealing in the absence of gradient is not reliable enough as an algorithm for optimal path planning problems.

Figure (4.20) depicts the E-ARC algorithm's corresponding solution. All simulations from the same initial guess converge to the same solution; one of them is shown in this figure, implying more reliability of the algorithm. Figure (4.20(c)) shows convergence trends for the same objective function, both in the presence and in the absence of constraints. Moreover, the hazardous regions (or obstacles) are also represented in Figures (4.17(a)) and (4.17(b)).

A set of benchmark problems is shown in Figure (4.21). Each simulation is run several times and usually reaches the same solution. Depending upon the initial guess, the algorithm might find the different solutions, meaning that the algorithm still suffers from being trapped in the local solution. This issue can be solved by enhancing multi-agent structure, in which several paths are exploring the region simultaneously. Usually, in this swarm-like structure, each agent is searching locally on its own for the best solution; with an extra driving force, it is compelled to converge toward the best current agent [1]. The authors examine the performance of this approach [1] in some other applications. The last set of simulations is run in the presence of a constraint: namely, obstacles or hazardous regions. In the simulations shown in Figure (4.22), the hazardous regions are the same and only objective functions are different. Figure (4.17(a)) and (4.17(b)) show 3D and 2D representations of constraints respectively, which can never be violated because the evolving curve cannot enter the targeted regions. These regions are not shown in Figure (4.22) to depict the optimal paths more clearly.

4.3.4 Conclusions of Optimal Path Planning Problem

The E-ARC algorithm has been developed and investigated in this section in order to cope with the optimal path planning problem. Based on the stochastic level set method, E-ARC is an optimization algorithm in a variational framework.

This section has examined the capability of E-ARC and its strong potential to extend to other applications such as image processing and topology optimization. In this section, the open level set approach [138] has been employed in order to define the evolving path; then, its level set formulation was utilized by stochastic operators enabling to escape from locally optimal solutions. The simulations have shown that these two steps were insufficient for finding an accurate optimal solution. To solve this problem, the E-ARC algorithm provides a set of novel operators that enable local search for the optimal path along with the global evaluation of regular stochastic level set method.

Moreover, an original method to implement the constraints was introduced. Usually a significant weight was assigned to the hazardous regions or the position of obstacles which allows the optimization process to avoid entering the targeted regions; in this work, however, the aforementioned regions were frozen and the evolving path could not enter the regions. The Flood Fill algorithm [144] was employed to remove the unnecessary part of the path when the evolving path crosses itself. This algorithm was originally employed in image processing to remove unnecessary holes in images [144]. Ultimately, the E-ARC algorithm has demonstrated significant improvements to the optimal path planning techniques.

Despite E-ARC's ability to escape from local solutions, in some cases, depending on where the algorithm started and the complexity of the objective function, it might still get stuck in a local solution. Extension of E-ARC to a multi-agent structure is a solution to cope with this difficulty. The effectiveness of this multi-agent structure was shown in previous sections.

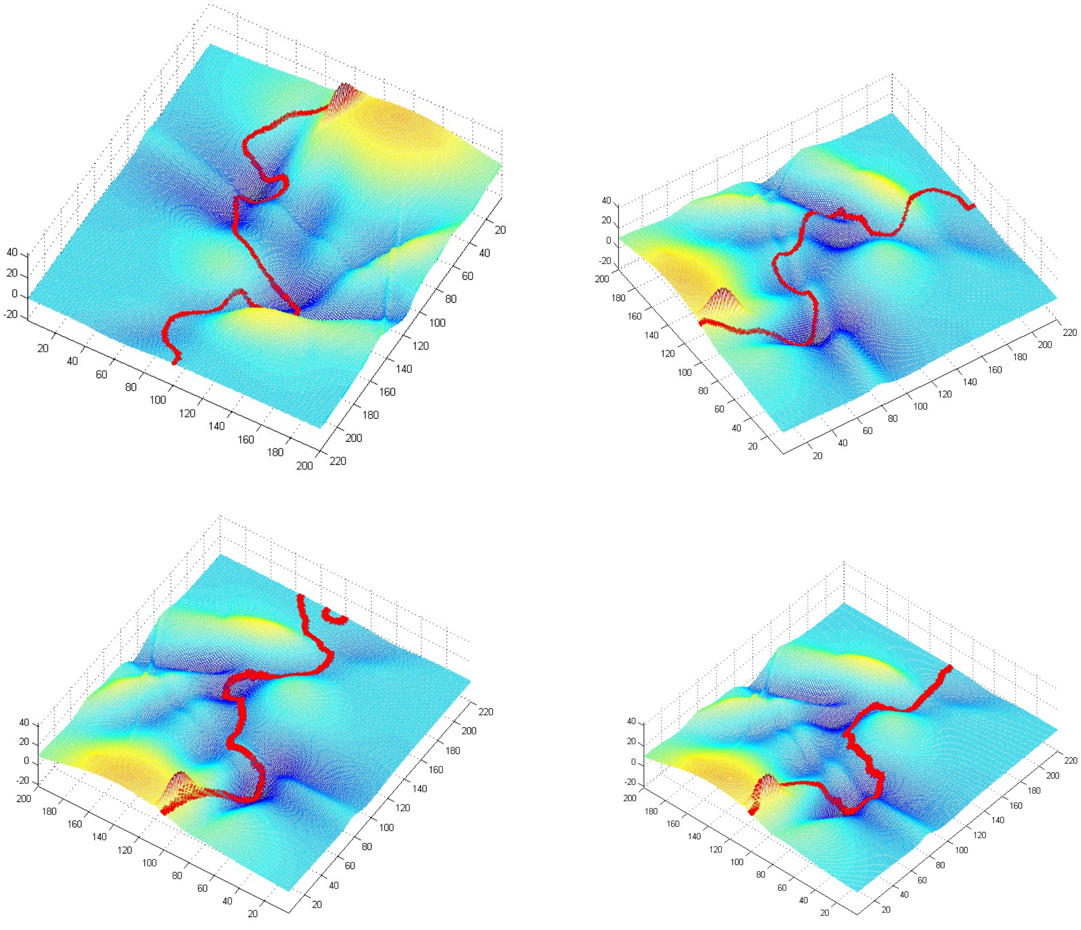
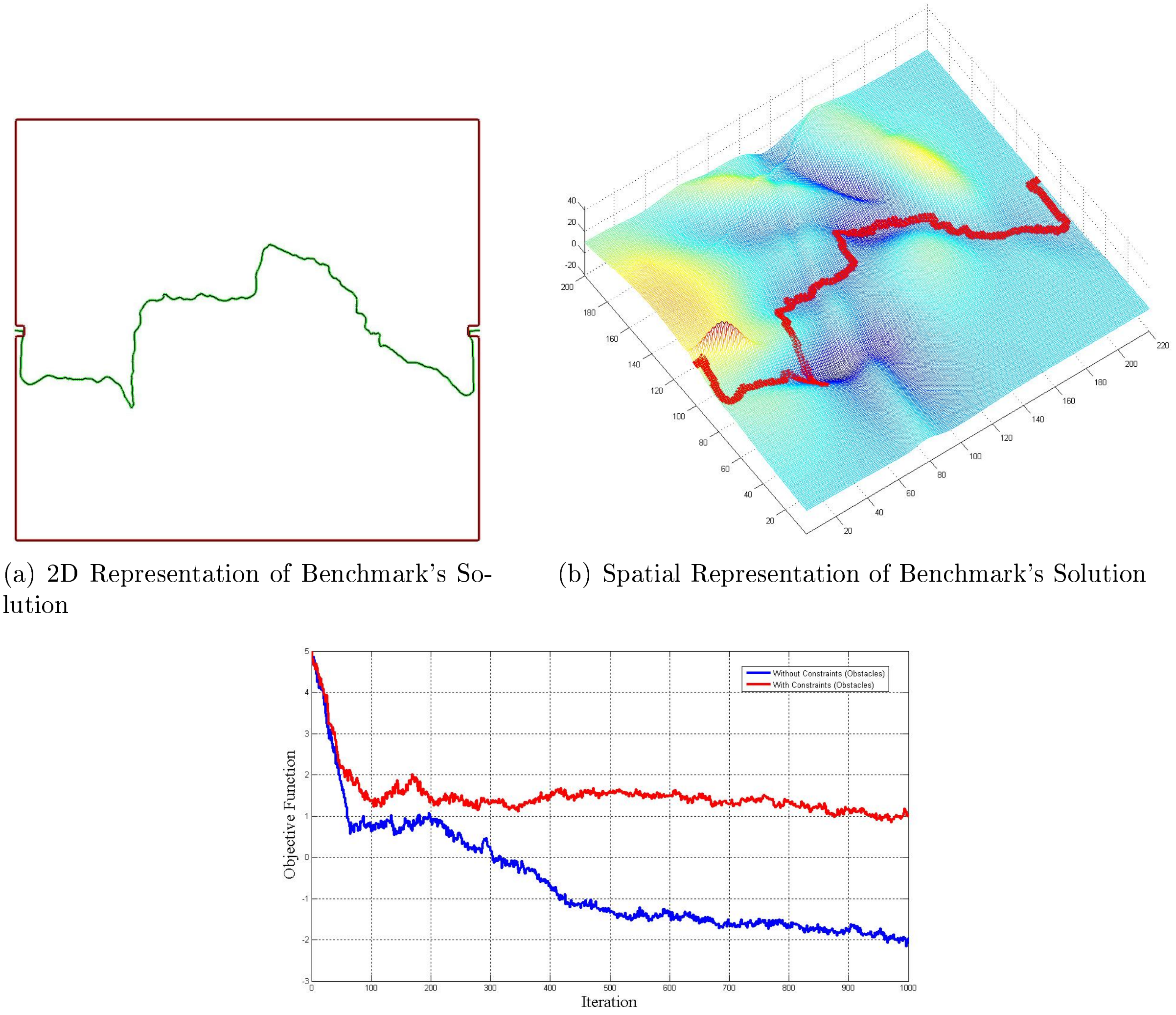


Figure 4.19: Different results initiated from the same initial guesses, produced by Simulated Annealing



(a) 2D Representation of Benchmark's Solution

(b) Spatial Representation of Benchmark's Solution

(c) E-ARC 's Convergence on the Benchmark Problem

Figure 4.20: E-ARC 's results on the benchmark shown in Figure (4.19)

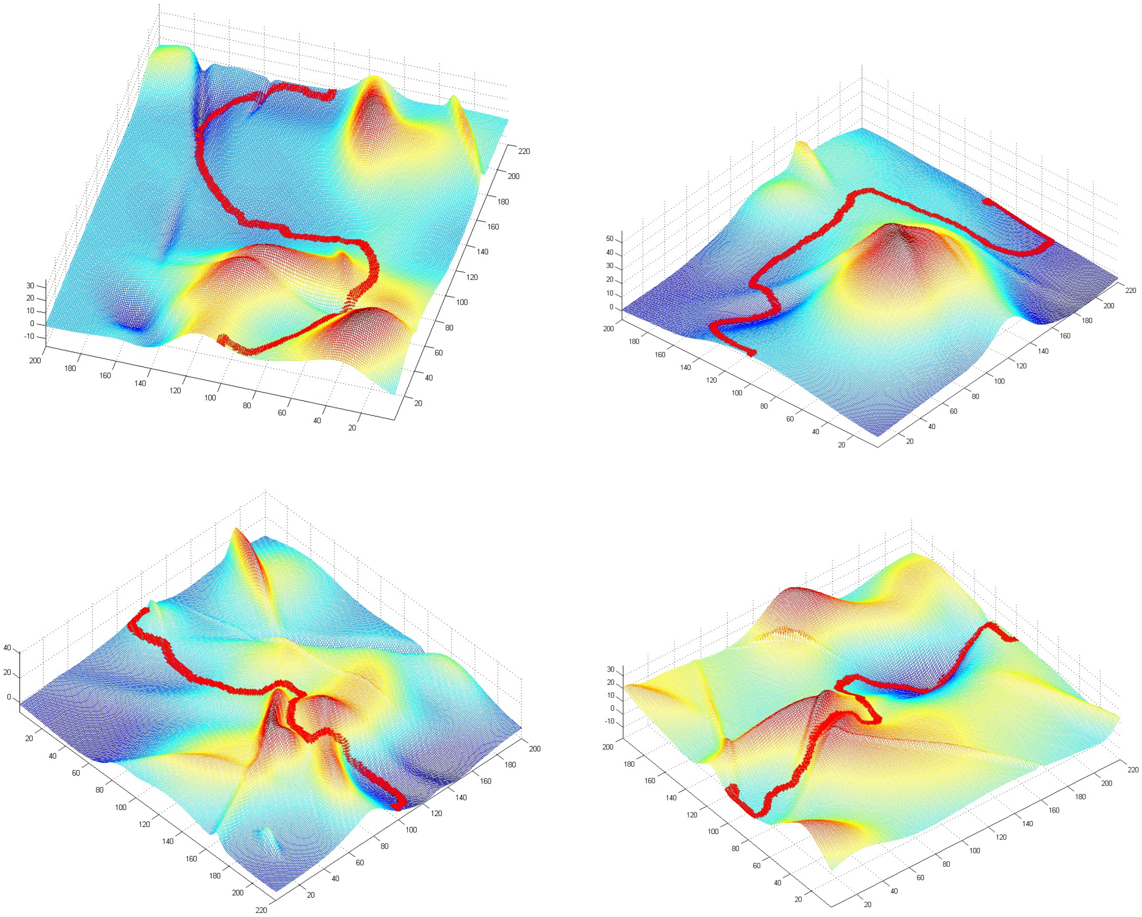


Figure 4.21: Different benchmarks explored by E-ARC

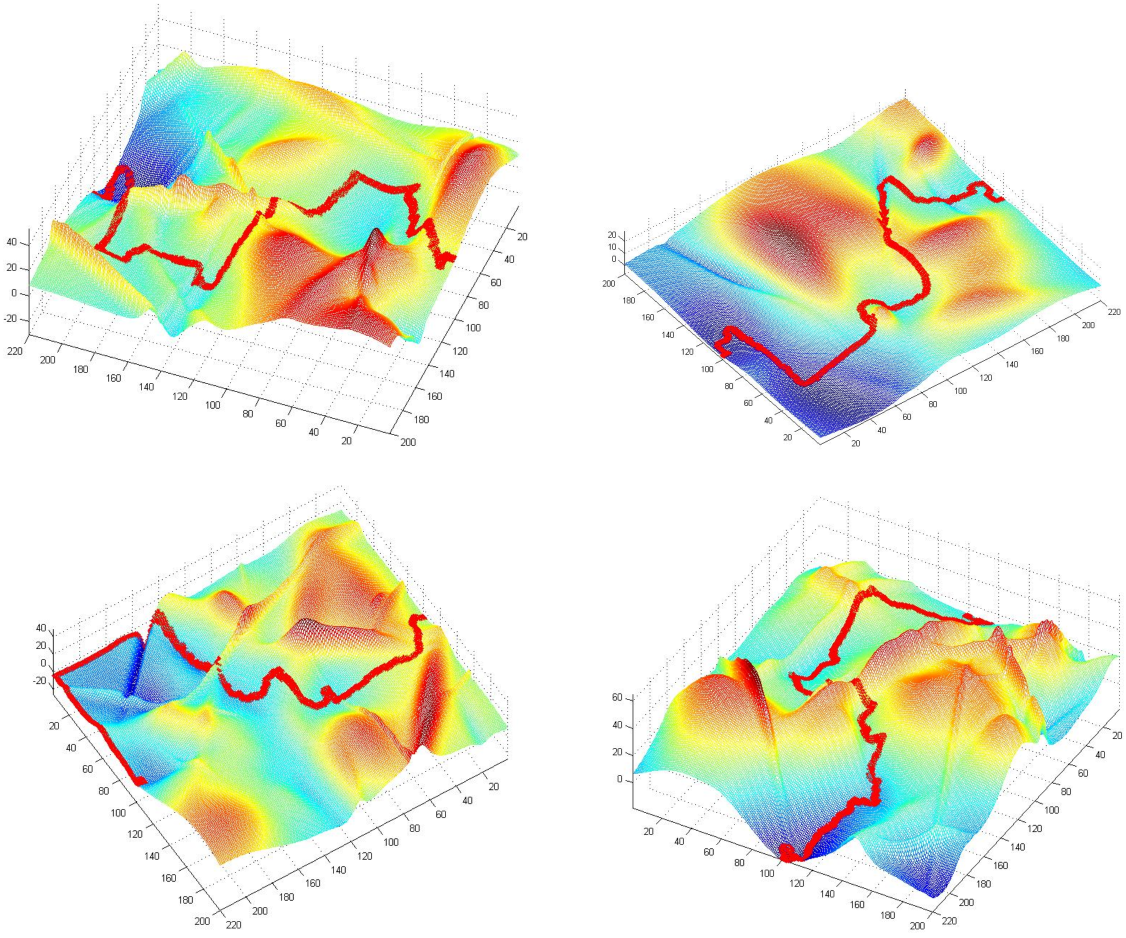


Figure 4.22: Simulation results in presence of obstacles or hazardous regions

Chapter 5

Conclusion and Future work

A set of global optimization algorithms was proposed in this study in order to cope with variational applications as well as parameter optimization problems. A simple investigation shows that the existing global parameter optimization methods cannot be easily extended to the variational framework. Therefore, a new algorithm in parameter space, SBFO, was proposed and by using the ideas learned from this approach two new algorithms, ACSF and MSLSM, in variational framework were proposed. These two algorithms are suitable for some applications, such as image processing and topology optimization, in which gradient function is analytically available. For the cases such as optimal path planning, when the gradient is not explicitly available, E-ARC algorithm was suggested. All of the aforementioned algorithms are capable of solving constraint optimization problem as well as unconstrained ones.

Inspired by bacterial aggregation process, SBFO was proposed and utilized by a local search tool, gradient operator, as well as global search approaches such as multi-agent structure and stochastic operators. SBFO can be categorized under particle swarm optimization methods and by tuning the weighting parameters; one can improve the local search or global search capability whenever needed. As a rule of thumb, the bigger the λ in Equation (3.2), the faster the convergence is and the lower the capability of local search. At the present time, tuning of this parameter is not automatic and user should define it at the beginning of the process. The performance of SBFO was tested in some benchmark problems. It demonstrates good results compared to the algorithms in its own category, as well as in evolutionary algorithms such as GA. In terms of convergence speed, as SBFO is utilized by a gradient operator, it is clearly faster than GA, although GA shows better performance when there is no limitation on number of iterations. The main advantage of SBFO is its capability of local search and additionally as motion of each agent is based on steepest descent direction; one can conclude that SBFO can be extended to variational framework where the steepest descent method is replaced by the level set method.

To deal with variational problems such as image segmentation, topology optimization and path planning problems, level set method has been widely used by researchers in different branches of science and engineering. Stochastic level set method is proposed by [57] to avoid being trapped in local solutions. The general model suggested by Juan [57] cannot be supported by the theorems proposed in [72]- [73] since the noise sources are spatial dependent. Active contour with stochastic fronts (ACSF) as a special case was designed in this thesis such that the convergence of the stochastic model towards the deterministic model was guaranteed based on [81]. The convergence analysis of ACSF was discussed in this thesis in detail. Moreover, ACSF has compact equation which is easier for implementation and less computationally expensive compared to general form, suggested by Juan [57].

The structure of ACSF allows the front to have an independent fluctuation at each point of it that helps to escape from local solutions. This independent jump is useful because during the process each point on the front can move to a better position independently if possible (possibility will be determined by annealing scheme). While dependent source noise, may create a situation in which only some points have this opportunity to select a better position and other points on the front are forced to move to the positions that are not necessarily better than before. Obviously this happens even if the overall objective function is finding its way toward a better value.

Another approach to discover the entire regions is to have a multi-agent structure that each agent searches for a solution locally and at the same time all the agents are compelled to converge to the best current solution. Multi-agent stochastic level set method was intended for this purpose. Apparently, simulations of this approach in image processing are not as good as expected. The reason is that the initial guess usually has enough holes, each hole plays a role of an independent agent; hence, basically ACSF on its own is strong enough to discover the whole feasible space. Consequently, when ACSF and MSLSM are compared over the medical image processing there is no significant improvement in results. For the same reason and this fact that topology optimization is remarkably computationally expensive, MSLSM was not applied to topology optimization problem.

Results show that ACSF is also capable of finding a consistent solution for topology optimization problems. If one compares the convergence pattern of different case studies in this problem, he/she can figure out that although the resulting topologies are not exactly the same but the objective functions are acceptably the same. This means that to distinguish between the different topologies, a new objective function should be defined;

otherwise improvement of the optimization algorithm cannot help anymore.

The performance of ACSF and MSLSM is clearly dependent on both deterministic and stochastic models, Equation (3.62). In some application, such as optimal path planing, finding an analytical form of the deterministic model is impossible or computationally expensive. To clarify this, it is worth mentioning that the selection of objective functions in optimal path planning can be as wide as fuel consumption of a rover to the amount of energy that is absorbed by solar panels that is a function of rover orientations in space. To cope with such problems, E-ARC algorithm was proposed that do not need the gradient of objective function explicitly. Instead of gradient at each point, the steepest descent direction is defined for a set of points over a portion of the front. The length of this portion is randomly selected at each time step. This helps the front to uniformly converge toward the solution. To employ the level set method in open curve fashion, the idea suggested by Smereka [138] was used. By extension of this idea, the implementation of constraint such as hazardous area becomes considerably easier and more effective compared to the other approaches for this purpose. The results show that the E-ARC algorithm is very effective for the path planning problem especially compared to stochastic level set method with the absence of a gradient function.

To improve the performance of the proposed algorithms, the following can be suggested:

As mentioned, SFBO structure has a parameter, λ in Equation (3.2), to compromise between local and global search. This parameter is now selected by trial and error at the beginning of the optimization process, but during the process the required value varies depending upon the current states of each agent. For example, if an agent is trapped in a local point, this parameter should be large enough to help the agent escape from that local solution; on the other hand, if the agent is still looking for a local solution and the length of gradient vector is big, the algorithm should allow it to find its own local point. Therefore an adaptive gain scheduling is really required to improve the performance of the algorithm. The same idea can be applied to MSLSM for the parameter that forces the agents to be similar to each other.

For image processing application, as stated, the accuracy of the algorithm in detecting the low contrast edges is limited to the capability of deterministic model. The current algorithm only helps to explore the entire image. Therefore, by using high performance models like the one suggested by Bresson et al. [69], the algorithm not only can explore the image as much as possible, but also can detect low contrast edges.

A real need for topology optimization problem is to improve the objective function such that it results in a more robust design against geometrical and loading uncertainties. As can be seen, ACSF is able to achieve almost the same objective functions with starting from different initial guesses but still the final topologies are not exactly the same. Therefore, it is necessary to have an algorithm that can choose a solution (amongst these different topologies) that is more robust. In doing so, some other terms should be added to the objective function (deterministic model) such that it can distinguish between robust and sensitive design. The current deterministic model cannot manage this problem yet.

Although the E-ARC algorithm has demonstrated acceptable results for many of the benchmark problems, there are still some cases that E-ARC cannot find the best solution. One way to improve it is to have several evolving curves instead of only one. This increases the chances of being able to find the best possible solution. By looking at Figure (4.14), which shows the real phenomenon, by which E-ARC is inspired, one can conclude that to find the best path, the real event is also using different evolving paths such that only one of them is the best, which is the most illuminated one.

References

[1] A., Kasaiezadeh, A., Khajepour, S.L., Waslander, *Spiral Bacterial Foraging Optimization Method*, American Control Conference, ACC2010, Baltimore, Maryland, USA, 2010, June 30 - July 2.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. Michigan Press, 1975.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning,* Boston, MA: Kluwer, 1989.

[4] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. Hoboken, NJ: Wiley, 1966.

[5] J. Kennedy, R. Eberhart, *Particle swarm optimization,* in Proceeding IEEE Int. Conf. Neural Network, 1995, pp. 1942–1948.

[6] M. Dorigo, L. M. Gambardella, *Ant colony system: A cooperative learning approach to the traveling salesman problem*, IEEE Trans. Evol. Comput., Apr. 1997, Vol. 1, no. 1, pp. 53–66.

[7] M. P. Song, G. C. Gu, *Research on Particle Swarm Optimization: A Review,* Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, Shanghai, China, 26-29 August, 2004, pp. 2236-2241.

[8] P.J.,Angeline, *Evolutionary optimization versus particle swarm optimization: philosophy and performance difference,* In Proceeding 7th Annu. Conf. on Evolutionary Programming, San Diego,. California USA, 1998, pp. 601-610.

[9] B. Niu, Y.L. Zhu, X.X. He, X.P. Zeng, *An improved particle swarm optimization based on bacterial chemotaxis*, in Proceeding of the Sixth World Congress on Intelligent Control and Automation, 2006, pp. 3193–3197.

[10] K.E., PARSOPOULS, V.P., PLAGIANAKOS, G.D., MAGOULAS, M.N., VRAHATIS, *Stretching technique for obtaining global minimizers through particle swarm optimization*, In Proceeding Workshop on Particle Swarm Optimization, Indianapolis USA, 2001, pp. 22-29.

[11] A., SILVA, A., NEVES, E., COSTA, *SAPPO: A simple, adaptable, predator prey optimizer,* Lecture Notes in Artificial Intelligence, 2003, Vol. 2909, pp.59-73.

[12] H. J. BREMERMANN, *Chemotaxis and optimization,* J. Franklin Inst. , 1974, Vol. 297,pp. 397–404.

[13] H. J. BREMERMANN, R. W. ANDERSON, *How the brain adjusts synapses may be in Automated Reasoning*, Essays in Honor of Woody Bledsoe, R. S. Boyer, Ed. Norwell, MA: Kluwer, 1991, pp. 119–147.

[14] R.W. ANDERSON, *Biased random-walk learning: A neurobiological correlate to trial-and-error,* in Neural Networks and Pattern Recognition, O. M. Omidvar and J. Dayhoff, Eds. New York: Academic, 1998, pp. 221–244.

[15] S. D. MÜLLER, J. MARCHETTO, S. AIRAGHI, AND P.KOUMOUTSAKOS, *Optimization based on bacterial chemotaxis,* IEEE Trans. Evol. Comput., Feb.2002, Vol. 6, pp. 16–29.

[16] K.M. PASSINO, *Biomimicry of bacterial foraging for distributed optimization and control*, IEEE Control Syst. Mag., Jun. 2002, Vol. 22, no. 3, pp. 52–67.

[17] Y. LIU AND K. M. PASSINO, *Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviors*, J. Optim.Theory Appl., 2002, Vol. 115, No. 3, pp. 603–628.

[18] S. DAS, A., ABRAHAM, A. KONAR, *On Stability of the Chemotactic Dynamics in Bacterial-Foraging Optimization Algorithm,* IEEE Transanctions on Systems, Man, and Cybernetics Part A: Systems and Humans, May 2009, Vol. 39, No. 3.

[19] *http://www.icsi.berkeley.edu/ storn/code.html*

[20] A., BISWAS, S., DASGUPTA, S., DAS, AND A. ABRAHAM, *A Synergy of Differential Evolution and Bacterial Foraging Algorithm for Global Optimization*, Neural Network World, 2007, Vol. 17, No. 6, pp.607-626.

[21] A., Biswas, S., Dasgupta, S., Das, A., Abraham, *Synergy of PSO and Bacterial Foraging Optimization: A Comparative Study on Numerical Benchmarks*, Second International Symposium on Hybrid Artificial Intelligent Systems (HAIS 2007), Advances in Soft computing Series, Springer Verlag, Germany, Corchado, E. et al. (Eds.): Innovations in Hybrid Intelligent Systems, ASC 44, 2007, pp. 255-263.

[22] A. Abraham, A. Biswas, and S. Dasgupta et al. *Analysis of reproduction operator in bacterial foraging optimization algorithm*, In CEC 2008: IEEE World Congress on Computational Intelligence, Hong Kong, June, 2008. IEEE Press. pp. 1476–1483.

[23] H., Chen, Y., Zhu, K., Hu, X., He, B., Niu, *Cooperative Approaches to Bacterial Foraging Optimization,* In: ICIC (2), 2008, pp. 541–548.

[24] W., Korani, *Bacterial foraging oriented by particle swarm optimization strategy for PID tuning*, GECCO 2008: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, 2008, pp. 1823–1826.

[25] H., Shen, Y., Zhu, X., Zhou, H., Guo, C., Chang, *Bacterial Foraging Optimization Algorithm with Particle Swarm Optimization Strategy for Global Numerical Optimization,* Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, 2009, pp. 497-504.

[26] S., Yasushi, U., Masahiro, *Cell Signaling Reactions: Single-Molecular Kinetic Analysis*, Springer, 2011.

[27] C.J. Tomlin, J.D. Axelrod, *Biology by numbers: mathematical modeling in developmental biology*, Nature Rev Gen, 2007, Vol. 8, pp. 331–340.

[28] P., Venkataraman, *Applied Optimization with Matlab Programming*, Wiley, 2009.

[29] S.S. Rao, *Engineering Optimization: Theory and Practice* , John wiley & Sons, New York. 1996.

[30] O., Bozorg Haddad, A., Afshar, M.A., Mariano, *Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization*, 2006, Water Resources Management, Vol. 20, pp. 661-680.

[31] Y., Ermoliev, *Stochastic Quasi-gradient Methods and Their Application to System Optimization*, 1983, Stochastic Vol. 9, pp. 1-36.

[32] Yu., M., Ermoliev , E.A. Nurminski, *Stochastic quasi-gradient algorithms for minimax problems*, Edited by M. Dempster. Proceedings of the International Conference on Stochastic Programming, London: Academic Press. 1980.

[33] B.T., Poljak, *Nonlinear programming methods in the presence of noise, Mathematical Programming*, Vol. 14, 1978.

[34] J.N., Tsitsiklis, D.R, Bertsekas, M., Athans, *Distributed Deterministic and Stochastic Gradient Optimization Algorithms*, IEEE Transactions on Automatic Control. 1986, Vol. 31, pp. 803-812.

[35] A. N. Nakonechnyi, *Stochastic gradient processes: a survey of convergence theory using Lyapunov second method*, Cybernetics and Systems Analysis, 1995, Vol. 31, No. 1, pp. 46–62.

[36] A., Shapiro, Y. Wardi, *Convergence analysis of gradient descent stochastic algorithms,* Journal of Optimization Theory and Applications, 1996, Vol. 91, pp. 439–454.

[37] J. C., Fort, G., Pages, *Convergence of stochastic algorithms: from Kushner-Clark theorem to the Lyapounov functional method*, Adv. in Applied Probability. 1996, Vol. 28, pp. 1072-1094.

[38] M. V., Solodov, B. F., Svaiter, *Descent Methods with Line Search in the Presence of Perturbations*, J. Computational and Applied Mathematics, 1997, Vol. 80, pp. 265-275.

[39] M. V., Solodov, S. K., Zavriev, *Error Stability Properties of Generalized Gradient-Type Algorithms*, J. Opt. Theory and Appl., 1998, Vol. 98, No. 3, pp. 663–680.

[40] V.S., Borkar, S.P., Meyn, *The O.D.E method for convergence of stochstic approximation and reinforcement learning*, Siam J. Control, 2000, Vol., 38, No. 2, pp.447–69.

[41] B., Bharath, V., Borkar, *Stochastic Approximation Algorithms: overview and Recent Trends*, Sadhana, 1999, Vol. 24, pp. 425–452.

[42] D.P., Bertsekas, J.N., Tsitsiklis, *Gradient convergence in gradient methods with errors*, SIAM J. Optim., 2000, Vol., 3, pp. 627–642.

[43] J. Z., Shi, *Convergence of Line-Search Methods for Unconstrained Optimization*, Applied Mathematics and Computation, 2004, Vol., 157, pp. 393-405.

[44] M., Li, C.,Wang, *Convergence property of gradient-type methods with non-monotone line search in the presence of perturbations*, Applied Mathematics and Computation, 2006, Vol. 174, Issue 2, pp. 854-868.

[45] S., Boyd, L., Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK., 2004.

[46] Y. Law, H. Lee, and, A. Yip, "A Multi-resolution Stochastic Level Set Method for Mumford-Shah Image Segmentation." IEEE Transactions on Image Processing, 17(12):2289–2300, 2008.

[47] R. C. Gonzalez and R. E.Woods, Digital Image Processing, 2nd ed.Englewood Cliffs, NJ: Prentice-Hall, 2002.

[48] J. Freixenet, X. n. Muñoz, D. Raba, J. Martí, and X. Cufí, "Yet another survey on image segmentation: Region and boundary informationintegration," in Proc. ECCV, 2002, vol. 2352, pp. 408–422, Lecture Notes Comput. Sci.

[49] V. Shrimali, R. Anand, and V. Kumar, "Current trends in segmentation of medical ultrasound b-mode images: A review," IETE Technical Review, vol. 26, no. 1, pp. 8-17, January 2009.

[50] J. Revell, M. Mirmehdi, D. McNally, "Applied review of ultrasound image feature extraction methods". In Houston,A., Zwiggelaar, R., eds.: The 6th Medical Image Understanding and Analysis Conference, BMVA Press pp. 173–176, 2002.

[51] E. D. Angelini, Y. Jin, and A. Laine, "State-of-the-art of level set methods in segmentation and registration of medical imaging modalities," in Handbook of Medical Image Analysis: Advanced Segmenation and Registration Models, Kluwer Academic, New York, NY, USA, 2004.

[52] S. Osher, R. Fedkiw, "Level set methods and dynamic implicit surfaces", Springer, Berlin Heidelberg New York, 2002.

[53] J.A. Sethian, "Level-Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science", Cambridge University Press, Cambridge, MA, 1999.

[54] S. Osher and N. Paragios. "Geometric Level Set Methods in Imaging, Vision, and Graphics", Springer Verlag, 2003.

[55] J. S. Suri, K. Liu, S. Singh, S. N. Laxminarayan, X. Zeng, and L. Reden, "Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review," IEEE Transactions on Information Technology in Biomedicine, vol. 6, pp. 8 -28, 2002.

[56] J. S. Suri and K. Liu, "Level set regularizers for shape recovery in medical images," in the Proceedings of IEEE Symposium on Computer-Based Medical Systems, pp. 369 - 374, Bethesda, MD, USA, 2001.

[57] O. Juan, R. Keriven, and G. Postelnicu, "stochastic motion and the level set method in computer vision: stochastic active contours," IJCV, vol. 69, no. 1, pp. 7–25, 2006.

[58] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in the Proceedings of IEEE International Conference in Computer Vision, pp. 810-815, Cambridge, MA, USA, 1995.

[59] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in the Proceedings of International Conference on Computer Vision, pp. 694-699, Cambridge, MA, USA, 1995.

[60] W. J, Niessen, B. M. t. H. Romeny, and M. A. Viergever, "Geodesic deformable models for medical image analysis," IEEE Transactions on Medical Imaging, vol. 17, pp. 634 - 641, 1998.

[61] V. Caselles, F. Catte, T. Coll, and F. Dibos, "A geometric model for active contours," Numerische Mathematik, vol. 66, pp. 1-31, 1993.

[62] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," International Journal of Computer Vision, vol. 1, pp. 321-331, 1987.

[63] D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems" Communications on Pure and Applied Mathematics, 42(5): 577-685, 1989.

[64] T. F. Chan and L. A. Vese, "Active contour and segmentation models using geometric PDE's for medical imaging," in Geometric Methods in Bio-Medical Image Processing, Mathematics and Visualization, R. Malladi, Ed.: Springer, 2002.

[65] T. F. Chan and L. A. Vese, "A level set algorithm for minimizing the Mumford-Shah functional in image processing," in the Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision, pp. 161-168, Vancouver, BC, Canada, 2001.

[66] T. F. Chan and L. A. Vese, "An efficient variational multiphase motion for the Mumford-Shah segmentation model," in the Proceedings of Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, pp. 490-494, Pacific Grove, CA, USA, 2000.

[67] T. F. Chan and L. A. Vese, "Active contours without edges," IEEE Transactions on Image Processing, vol. 10, pp. 266 - 277, 2001.

[68] L. A. Vese and T. F. Chan, "A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model", International Journal of Computer Vision 50(3): 271-293, 2002.

[69] X., Bresson, and S., Esedoglu, and P., Vandergheynst, and J.P., Thiran, and S., Osher, Fast global minimization of the active contour/snake model," Journal of Mathematical Imaging and Vision, 28:2, 2007.

[70] J.B., Walsh, “An introduction to stochastic partial differential equations”, Ecole d’Eté de Prob. de St. Flour XIV, 1984, Lect. Notes in Math 1180, Springer Verlag (1986).

[71] N.K. Yip, “stochastic motion by mean curvature”, Arch. Rational Mech. Anal., 144:331–355, 1998.

[72] Lions, P. and P. Souganidis: 2000a, ‘Fully nonlinear stochastic partial differential equations with semilinear stochastic dependence’. C.R. Acad. Sci. Paris Ser. I Math 331, 617-624.

[73] Lions, P. and P. Souganidis: 2000b, “Uniqueness of weak solutions of fully nonlinear stochastic partial differential equations”. C.R. Acad. Sci. Paris Ser. I Math 331, 783-790

[74] A. Lang. “Simulation of stochastic Partial Differential Equations and stochastic Active Contours.” Dissertation, University at Mannheim, Institut fur Mathematik, Lehrstuhl fur Mathematik V (2007).

[75] S. Chen and R.J. Radke, “Markov Chain Monte Carlo Shape Sampling using Level Sets, Second Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)”, in conjunction with International Conference on Computer Vision 2009, September 2009.

[76] Y. Pan, J. D. Birdwell, S. M. Djouadi, “Probabilistic Curve Evolution Using Particle Filters”, Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 Seville, Spain, December 12-15, 2005.

[77] G., Unal, H., Krim, A., Yezzi, “Stochastic differential equations and geometric flows”, Image Processing, IEEE Transactions on, 11:2, 2002.

[78] G.B., Arous , A., Tannenbaum, O., Zeitouni, “Stochastic approximations to curve-shortening flows via particle systems”, Journal of Differential Equations, 195:1, 2003.

[79] R. Malladi, J. A. Sethian, and B. C. Vemuri, “A topology independent shape modeling scheme,” in Proc. SPIE Conf. Geometric Methods Computer Vision II, vol. 2031, San Diego, CA, 1993, pp. 246–258.

[80] Gelasca, E. D. , Byun,J., Obara, B. and Manjunath, B. S., Evaluation and benchmark for biological image segmentation, in IEEE International Conference on Image Processing. San Diego, CA, USA: IEEE Signal Processing Society, 12-15 October 2008, pp. 1816-1819.

[81] Caruana, M., Friz, P. K., and Oberhauser, H., 2011. “A (rough) pathwise approach to a class of non-linear stochastic partial differential equations.”. pp. 27–46. journal: Annales de l’Institut Henri Poincaré. Analyse Non Linéaire; Rainer Buckdahn (Brest).

[82] Friz, P. K., and Oberhauser, H., 2009. Rough path limits of the wong zakai type with a modified drift term.

[83] Friz, P., Berlin, T., Berlin, W., Oberhauser, H., and Berlin, T., 2010. Rough path stability of spdes arising in non-linear filtering.

[84] Gard, T. C., 1988. *Introduction to stochastic differential equations*, Vol. 114. M. Dekker, New York. Thomas C. Gard.; Bibliography: p. 219-225.; Includes index.

[85] Higham., D. J., 2001. “An algorithmic introduction to numerical simulation of stochastic differential equations”. pp. 525–546. journal: SIAM Rev.

[86] Karatzas, I., and Shreve, S. E., 1991. *Brownian motion and stochastic calculus*, 2 ed., Vol. 113. Springer-Verlag, New York.

[87] Kunita, H., 1990. *Stochastic flows and stochastic differential equations*, Vol. 24. Cambridge University Press, Cambridge ; New York. Hiroshi Kunita. -; Bibliography : p. 340-344.

[88] Schaffter, T., 2010. Numerical integration of SDEs: A short tutorial. Tech. rep. WingX.

[89] Lyons T. J., 1998. *Differential equations driven by rough signals*, Revista Mat. Iberoamericana, Vol. 14, p. 215-310

[90] Grayson, M., and Grossman, R., 1989. *Vector Fields and Nilpotent Lie Algebras.* Society for Industrial and Applied Mathematics, Philadephia, PA.

[91] Bendsoe, M. P., 1989. "Optimal shape design as a material distribution problem". pp. 193–202. journal: Structural and Multidisciplinary Optimization.

[92] Bendsoe, M. P., 1995. *Optimization of structural topology shape and material.* Springer-Verlag, Berlin. Martin P. Bendsoe.; Bibliography: p. [237]-260.

[93] Hassani, B., and Hinton, E., 1999. *Homogenization and structural topology optimization: theory, practice, and software.* Springer, London ; New York. Behrooz Hassani and Ernest Hinton.

[94] Allaire, G., Jouve, F., and Toader, A.-M., 2004. "Structural optimization using sensitivity analysis and a level-set method". *Journal of Computational Physics,* **194**(1), 2/10, pp. 363–393.

[95] Osher, S., and Sethian, J. A., 1988. "Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations". *JOURNAL OF COMPUTATIONAL PHYSICS,* **79**(1), pp. 12–49.

[96] Osher, S. J., and Santosa, F., 2001. "Level set methods for optimization problems involving geometry and constraints - i. frequencies of a two-density inhomogeneous drum". *J.Comput.Phys,* **171**, pp. 272–288.

[97] Allaire, G., 2002. *Shape optimization by the homogenization method*, Vol. 146. Springer, New York. Gregoire Allaire.; Includes bibliographical references (p. [427]-452) and index.; Applied mathematical sciences (Springer-Verlag New York Inc.) ; v. 146.

[98] Jia, H., Beom, H. G., Wang, Y., Lin, S., and Liu, B., 2011. "Evolutionary level set method for structural topology optimization". pp. 445–454. journal: Comput. Struct.

[99] Allaire, G., and Jouve, F., 2006. *Coupling the Level Set Method and the Topological Gradient in Structural Optimization*, Vol. 137 of IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, Springer Netherlands.

[100] Allaire, G., Allaire, G., Toader, A.-M., and Jouve, F., 2002. A level-set method for shape optimization.

[101] Wang, M. Y., Chen, S., Wang, X., and Mei, Y., 2005. “Design of multimaterial compliant mechanisms using level-set methods”. *Journal of Mechanical Design, Transactions Of the ASME,* **127**(5), pp. 941–956. Cited By (since 1996): 59.

[102] Wang, M. Y., and Wang, X., 2005. “A level-set based variational method for design and optimization of heterogeneous objects”. *Computer-Aided Design,* **37**(3), 3, pp. 321–337.

[103] Kong, T. Y., and Rosenfeld, A., 1989. “Digital topology: introduction and survey”. *Comput. Vision Graph.Image Process.,* **48**(3), December, pp. 357–393.

[104] Liu, Z., Korvink, J.G., and Huang, R., 2005. “Structure topology optimization: Fully coupled level set method via femlab”. *Structural and Multidisciplinary Optimization,* **29**(6), pp. 407–417. Cited By (since 1996): 22.

[105] Zhao, H. K., Chan, T., Merriman, B., and Osher, S., 1996. “A variational level set approach to multiphase motion”. *Journal of Computational Physics,* **127**(1), pp. 179–195. Cited By (since 1996): 394.

[106] Sage, A. P., 1968. *Optimum Systems Control.* N.J, Englewood Cliffs.

[107] van Dijk, N., Yoon, G., van Keulen, F., and Langelaar, M., 2010. “A level-set based topology optimization using the element connectivity parameterization method”. *Structural and Multidisciplinary Optimization,* **42**(2), pp. 269–282.

[108] Sokolowski, J., and Zochowski, A., 2003. “Optimality conditions for simultaneous topology and shape optimization.”. pp. 1198–1221. JOURNAL: SIAM J. Control and Optimization".

[109] Amstutz, S., and Andra, H., 2006. “A new algorithm for topology optimization using a level-set method”. pp. 573–588. journal: J. Comput. Phys.

[110] G. Allaire, F. Jouve, 2006. “Coupling the level set method and the topological gradient in structural optimization”. pp. 3-12. journal: M.P. BendsÃÿe, N. Olhoff, O. Sigmund (Eds.), IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials.

[111] He, L., Kao, C.-Y., and Osher, S., 2007. “Incorporating topological derivatives into shape derivatives based level set methods”. pp. 891–909. journal: J. Comput. Phys.

[112] Dunning, P. D. and Kim, H. A., 2010. “A new hole insertion method for level set based topology optimization”. the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization, Fort Worth TX USA, Sep.

[113] Rouhi, M., Rais-Rohani, M., and Williams, T., 2010. “Element exchange method for topology optimization”. pp. 215–231. journal: Structural and Multidisciplinary Optimization; 10.1007/s00158-010-0495-9.

[114] Jia, H., Beom, H. G., Wang, Y., Lin, S., and Liu, B., 2011. “Evolutionary level set method for structural topology optimization”. *Comput. Struct.,* **89**, March, pp. 445–454.

[115] Rong, J. H., and Liang, Q. Q., 2008. “A level set method for topology optimization of continuum structures with bounded design domains”. *Computer Methods in Applied Mechanics and Engineering,* **197**(17-18), 3/1, pp. 1447–1465.

[116] Luo, Z., and Tong, L., 2008. “A level set method for shape and topology optimization of large-displacement compliant mechanisms”. *International Journal for Numerical Methods in Engineering,* **76**(6), pp. 862–892. Cited By (since 1996): 14.

[117] Yulin, M., and Xiaoming, W., 2004. “A level set method for structural topology optimization and its applications”. *Adv.Eng.Softw.,* **35**(7), jul, pp. 415–441.

[118] Ananthasuresh, G. K., 2003. *Optimal synthesis methods for MEMS*. Kluwer Academic Publishers, Boston. G.K. Ananthasuresh.; Includes bibliographical references and index.

[119] Challis, V., 2010. “A discrete level-set topology optimization code written in matlab”. pp. 453–464. journal: Structural and Multidisciplinary Optimization; 10.1007/s00158-009-0430-0.

[120] Chen, S. K., 2006. Compliant mechanisms with distributed compliance and characteristic stiffness: A level set approach.

[121] Howell, L. L., 2001. *Compliant Mechanisms*, 1 ed. Wiley-Interscience, jul.

[122] Yamada, T., Yamasaki, S., Nishiwaki, S., Izui, K., and Yoshimura, M., 2011. “Design of compliant thermal actuators using structural optimization based on the level set method”. p. 011005. journal: Journal of Computing and Information Science in Engineering.

[123] W. Choi, D. Zhu, and J. C. Latombe, "Contingency-tolerant robot motion planning and control," in *Intelligent Robots and Systems '89. The Autonomous Mobile Robots and Its Applications. IROS '89. Proceedings., IEEE/RSJ International Workshop on*, 1989, pp. 78–86.

[124] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *Proceedings., 1993 IEEE International Conference on Robotics and Automation* , 1993, pp. 802–807 vol.2.

[125] B. Krogh and C. Thorpe, "Integrated path planning and dynamic steering control for autonomous vehicles," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol. 3, 1986, pp. 1664–1669.

[126] F. Lamiraux, D. Bonnafous, and O. Lefebvre, "Reactive path deformation for nonholonomic mobile robots," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 967–977, 2004.

[127] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristicbased path planning," in *in: Proceedings of the Workshop on Planning under Uncertainty for Autonomous Systems at The International Conference on Automated Planning and Scheduling (ICAPS*, 2005.

[128] J.-C. Latombe, *Robot Motion Planning.* Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[129] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," in *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, vol. 2, 1994, pp. 1368–1373 vol.2.

[130] L. D. Cohen and R. Kimmel, "Global minimum for active contour models: A minimal path approach," vol. 24, no. 1, pp. 57–78, August 1997, journal: Int. J. Comput. Vision.

[131] R. Kimmel, A. Amir, and A. M. Bruckstein, "Finding shortest paths on surfaces using level sets propagation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 17, no. 6, pp. 635–640, 1995.

[132] R. Kimmel and J. A. Sethian, "Optimal algorithm for shape from shading and path planning," vol. 14, p. 2001, 2001, journal: Journal of Mathematical Imaging and Vision.

[133] K. Alton and I. M. Mitchell, "Optimal path planning under defferent norms in continuous state spaces," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 866–872.

[134] B. Xu, D. J. Stilwell, and A. Kurdila, “Efficient computation of level sets for path planning,” in *Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems*, ser. IROS’09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 4414–4419.

[135] W. Na and C. Ping, “Path planning algorithm of level set based on grid modeling,” in *Computer Design and Applications (ICCDA), 2010 International Conference on*, vol. 5, 2010, pp. V5–508–V5–510.

[136] T. Cecil and D. E. Marthaler, “A variational approach to path planning in three dimensions using level set methods,” *Journal of Computational Physics*, vol. 211, no. 1, pp. 179–197, 1/1 2006.

[137] J. Canny, *The complexity of robot motion planning.* Cambridge, Mass.: MIT Press, 1988, vol. 1987, john Canny. -; Bibliography: p. [187]-195.

[138] P. Smereka, “Spiral crystal growth,” vol. 138, no. 3-4, pp. 282–301, April 2000, journal: Phys. D.

[139] B. Bickel, M. Wicke, and M. Gross, “Adaptive simulation of electrical discharges.”

[140] P. J. A. Burt, “Lightning: Physics and effects. by vladimir a. rakov and martin a. uman. cambridge university press, 2003. x + 687 pp. ISBN 0 521 583276 6,” vol. 59, no. 4, pp. 109–109, 2004.

[141] T. Kim and M. C. Lin, “Physically based animation and rendering of lightning,” in *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, 2004, pp. 267–275.

[142] J. O’Callaghan, "What is an electrical arc?", Dec2010 *http://www.howitworksdaily.com/science/question-of-the-day-what-is-an-electrical-arc*

[143] R. Malladi, “ A Topology-Independent Shape Modeling Scheme, *Ph.D. Dissertation. University of Florida, Gainesville*, FL, USA. AAI9505796. 1993.

[144] W. Burger, M. J. Burge, “Digital Image Processing: An Algorithmic Introduction using Java," *Springer*, November 2007.

[145] Y. N. Law, H. K. Lee, and A. M. Yip, “A multiresolution stochastic level set method for mumford shah image segmentation,” *Image Processing, IEEE Transactions on*, vol. 17, no. 12, 2008, pp. 2289–2300.

[146] N. K. Yip, "Stochastic motion by mean curvature," vol. 144, no. 4, pp. 313–355, 1998, journal: Archive for Rational Mechanics and Analysis.

[147] R. Carmona, H. Kesten, J. Walsh, and J. Walsh, *An introduction to stochastic partial differential equations*, ser. École d'Été De Probabilités de Saint Flour XIV - 1984. Springer Berlin / Heidelberg, 1986, vol. 1180, pp. 265–439,

[148] P.-L. Lions and P. E. Souganidis, "Fully nonlinear stochastic pde with semilinear stochastic dependence," *Comptes Rendus de Ácadémie des Sciences - Series I - Mathematics*, vol. 331, no. 8,2000, pp. 617–624, 10/15.

[149] P.-L. Lions, "Uniqueness of weak solutions of fully nonlinear stochastic partial differential equations," *Comptes Rendus de Ácadémie des Sciences - Series I - Mathematics*, vol. 331, no. 10, 2000, pp. 783–790, 11/15.