**This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.**

Self-configurable memetic algorithm

Le, Minh Nghia

2012

Le, M. N. (2012). Self-configurable memetic algorithm. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/49033

https://doi.org/10.32657/10356/49033

Downloaded on 13 Mar 2024 17:31:59 SGT

NANYANG TECHNOLOGICAL UNIVERSITY

SELF-CONFIGURABLE MEMETIC ALGORITHM

LE MINH NGHIA

SCHOOL OF COMPUTER ENGINEERING

2012

SELF-CONFIGURABLE MEMETIC ALGORITHM

LE MINH NGHIA

2012

SELF-CONFIGURABLE MEMETIC ALGORITHM

LE MINH NGHIA

LE MINH NGHIA

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2012

Abstract

To date, most successful advanced stochastic optimization algorithms involve some forms of individual learning or meme in their design. Memetic Algorithm (MA), as a form of hybridization between population-based and individual-based searches, represents one of the recent growing areas in evolutionary algorithm research. In the success and surge in interests on MAs, researchers have been exploring on various possible hybridizations of search operators towards the development and manual crafting of specialized algorithms that solve a specific problem or a set of problems effectively, using the domain knowledge obtained from human expertise. However, with so many population-based and individual-based procedures available for hybridizing, it is a tedious task, if not impossible, to design in advance an effective memetic algorithm for a given problem at hand. Furthermore, when high-fidelity analysis codes are used for evaluating design solutions in the optimization process, it is not uncommon for the single simulation process to take minutes, hours to days of supercomputer time to compute. Since the design cycle time of a product is directly proportional to the number of calls made to the costly analysis solvers, there has been practical needs for novel meta-model/surrogate-assisted memetic frameworks that can handle these forms of problems elegantly.

More recently, significantly more efforts have been directed towards formal and generic frameworks that address various design issues of MAs, leading to the field of emerging adaptive MAs, towards intelligent *self-configurable memetic search*. The image of modern evolutionary algorithms in computational intelligence can be established as a symbiosis of stochastic variation and individual learning, working in sync on the given problem in hand. In most research work on the topic, these search operators are generally studied as separate independent entities consisting of the stochastic population-based variation and individual learning phases, with much emphasis on the latter. Despite the extensive research efforts in the field, there has been limited progress made on revealing the working mechanisms responsible for the success

of memetic algorithms. Taking this cue, this dissertation presents an effort to fill this gap in the context of continuous optimization by reporting the work accomplished towards modelling the symbiosis or synergy of search operators in memetic search, which is deem to pose greatest influence on search performance, but has remained yet to be properly and formally investigated with rigor.

The symbiosis of stochastic variation and individual learning operators in MA search, labelled in this thesis as symbiosis search profile, is formalized in the form of the local optimum connectivity. To quantify the suitability of search components in creating viable, or potentially favorable solutions, the *evolvability* of symbiotic search profile or approximation model is introduced as the basis for assessment and subsequently used as the metric for adaptation in the present work. Via the effective use of statistical learning techniques on optimization data that are archived online, while the search progresses, productive symbiosis of unique search components or multiple fitness-improving approximation models are then proposed to mitigate the issues of complex problems. Empirical analyses on representative benchmark problems and case studies on real-world problems including the optimization of *OSS2 potential energy model* and *aerodynamic car rear design* produced noteworthy results, thus demonstrating the efficiency, competitiveness and robustness of self-configurable memetic search paradigm.

Acknowledgments

I would like to express my deepest gratitude to my academic supervisor in Nanyang Technological University (NTU), **Dr. Ong Yew Soon**.

His sincere encouragement during my indecisive moments and inspirational insights into the many aspects of my research direction have faithfully guided me throughout my research work. Without Dr. Ong's guidance, the fruitful achievement of this project will never be possible.

I would like to thank Stefan Menzel, Yaochu Jin, Bernhard Sendhoff and people at Honda Research Institute Europe for their beneficial comments and supports.

I would also like to thank my friends for their encouragements and the staffs in Emerging Research Lab and Center for Computational Intelligence for their technical supports.

Last but not least, special thanks to my parents and my wife for their constant support, understanding and love.

Minh Nghia Le

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optimization is a common problem arising in various domains that ranges from physics, biology, engineering designs to a plethora of other real world applications. It deals with the problem of minimizing or maximizing some metrics such as cost, energy consumption, aerodynamics, economical gains or others. The problem can be modeled either mathematically or via computational simulations, or through physical experimentations. This research focuses on the domain of continuous parametric optimization. Mathematically, the optimization problem can be expressed as to find the solution $\mathbf{x}^*$ in $\mathbf{x} = \{x_i\}_{i=1}^n$, which denotes a vector of *n* independent scalar variables such that the function $f(x_1, x_2, \ldots, x_n)$ is minimized (or maximized). Without loss of generality, the minimization problem is considered here. Function $f(\mathbf{x})$ provides a measure of the solution quality, and is commonly referred to as the *objective or fitness function*.

$$f(\mathbf{x}^*) = \min_{\mathbf{x}} f(x_1, x_2, \ldots, x_n)$$

In particular, this research considers the general non-linear programming single-objective continuous problem of the form:

$$\textbf{Minimize / Maximize:} \quad f(\mathbf{x})$$

$$\textbf{Subject to:} \quad \mathbf{x}_l^{(i)} \leq \mathbf{x}^{(i)} \leq \mathbf{x}_u^{(i)} \quad , \forall i = 1 \ldots n$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of design variables, and $\mathbf{x}_l$, $\mathbf{x}_u$ are vectors of lower and upper bounds, respectively.

For the last half of a century, the field of optimization has been studied extensively by researchers, with many algorithms and implementations that are now made available for used in the engineering community. The plethora of optimization techniques can generally be classified into three core categories, namely individual-based methods, population-based evolutionary algorithms and hybrid methods.

Individual-based method searches for an improved solution (if it exists) from a given starting point (i.e., a vector of decision variables) in a stochastic or deterministic manner. These methods have the advantage of high efficiency; however, they are sensitive to starting point selection and are more likely to stop at non-global optima than evolutionary population-based methods. Population-based (evolutionary) algorithms, on the other hand, operate on a population of individual solutions that evolves over time using nature-inspired search operators. They are less prone to stalling at local optima and have exhibited great potentials in solving non-convex, disjoint and noisy optimization problems. To date, numerous evolutionary optimization algorithms have been proposed, extending from genetic algorithms [65], evolution strategies [14], particle swarm optimization [212, 147], differential evolution [185, 156], estimation of distribution algorithm [92] to many others. These search methods have also been demonstrated with much success in many real world applications.

The third category of optimization methods is the hybrid evolutionary algorithm which has gathered remarkable increasing interest in the recent decades. From the algorithmic perspective, it is believed that two or more distinct algorithms when combined together in a synergistic manner can greatly enhance the problem-solving capacity of the derived hybrid. Inspired by Darwin's theory of natural evolution and Dawkins' notion of meme, the form of hybridization between population-based and individual-based searches that facilitates both exploration and exploitation via cooperation/competition, is known as Memetic Algorithm (MA), and represents the most popular hybridization paradigm today. Such hybrid algorithms have not only

been used successfully for solving a wide variety of engineering design problems but also shown to generate higher quality solutions more efficiently than canonical evolutionary algorithms [102, 136, 201, 119, 68, 55, 161, 35].

1.1 Research Objective

To date, most successful advanced stochastic optimization algorithms involve some forms of individual learning or meme in their design. Here, meme as the basic unit of cultural transmission is regarded by the computational intelligence community as a form of individual learning procedure capable of generating refinement on the given individual. Memetic Algorithm (MA), as a form of hybridization between population-based and individual-based searches, represents one of the recent growing areas in evolutionary algorithm research. Due to several recent and potential realizations of memes in the past two decades, MA research has further evolved into the studies of Memetic Computing, which is defined in [137] as a "*paradigm that uses the notion of meme(s) as units of information encoded in computational representations for the purpose of problem-solving*".

In the success and surge in interests on MAs, researchers have been exploring on various possible hybridizations of search operators towards the development and manual crafting of specialized algorithms that solve a specific problem or a set of problems effectively, using the domain knowledge obtained from human expertise. However, with so many population-based and individual-based procedures available for hybridizing, it is a tedious task, if not impossible, to design in advance an effective memetic algorithm for a given problem at hand. More recently, significantly more efforts have been directed towards formal and generic frameworks that address various design issues of MAs, leading to the field of emerging adaptive MAs, towards *self-configurable search*. It is expected that only autonomous self-configurable systems can turn high performance search algorithms into widely used methodologies. Such systems

should be able to configure themselves on the fly, be aware of its operational states and automatically adapt to a wide range of problem conditions. Adaptation of parameters and operators thus represents one of the most important and promising areas of research in the field. With such developments, an increasing varieties of adaptive memetic algorithms have been proposed in the literature, with many shown to solve a great variety of optimization problems more effectively and robustly than their canonical counterparts.

The image of modern evolutionary algorithms in computational intelligence can be established as a symbiosis of stochastic variation and individual learning, working in sync on the given problem in hand. In most research work on the topic, these search operators are generally studied as separate independent entities consisting of the stochastic population-based variation and individual learning phases, with much emphasis on the latter, which is evident in the works of Hinterding et al. [61] and Ong et al. [138]. Many schemes proposed for performing the adaptations however, are typically designed using a semi-ad-hoc or heuristic/ meta-heuristic paradigm that come with little theoretical rigor [132]. Few has investigated the synergistic behaviors of stochastic variations and lifetime learning in search formally. Despite the extensive research efforts in the field, there has been limited progress made on revealing the working mechanisms responsible for the success of memetic algorithms and the intrinsic properties of commonly used complex benchmark problems. Taking this cue, this dissertation presents an effort to fill this gap in the context of continuous optimization by reporting the work accomplished towards modelling the symbiosis or synergy of search operators in memetic search, which is deem to pose greatest influence on search performance, but has remained yet to be properly and formally investigated with rigor.

Last but not least, when high-fidelity analysis codes are used for evaluating design solutions in the optimization process, it is not uncommon for the single simulation process to take minutes, hours to days of supercomputer time to compute. A motivating example at Honda Research is aerodynamic car rear design, where one function evaluation involving a Computational Fluid Dynamics (CFD) simulation to calculate the fitness performance of a potential

design can take many hours of wall clock time. Besides parallelism, which is an obvious choice to achieving near linear order improvement in evolutionary search, researchers are gearing towards the approach of replacing as often as possible, calls to the costly high-fidelity analysis solvers with approximation models that are deem to be less costly to build and compute. Since the design cycle time of a product is directly proportional to the number of calls made to the costly analysis solvers, there has been practical needs for novel meta-model assisted memetic frameworks that can handle these forms of problems elegantly.

Thus, the main objectives of this research include the followings:

- providing insights to the memetic search mechanisms in continuous domain, especially on the symbiosis of stochastic variation and lifetime learning,
- developing intelligent self-configurable memetic framework that encourages productive symbiosis among unique search components, and
- incorporating advance approximation model management framework to handle computationally expensive problems in real-world optimization.

1.2 Organization of the Thesis

The rest of this thesis is organized as follows:

- In Chapter 2, the fundamentals of optimization and survey of several well-established optimization techniques are presented. Through this chapter, the background knowledge provided would serve to benefit the readers in better comprehending the main motivations and research works of this dissertation.
- Chapters 3-6 contain the core contributions of this dissertation. Particularly, Chapter 3 presents an extensive fitness landscape analysis on the local optimum structure of commonly used benchmark problems to reveal the influence of individual learning on the

Lamarckian memetic search performance. From there, Chapter 4 presents the formal model of stochastic variations and lifetime learning working in symbiosis for search, and introduces the notion of *evolvability* of symbiotic profile.

- Chapter 5 presents the proposed self-configurable memetic framework, established in this thesis as Symbiotic Evolution (SE), that facilitates the emergent and self-configuration of unique search profiles transpiring from the symbiosis of stochastic variation and lifetime learning.

- The generality of evolvability is further demonstrated in the self-configurable framework extended with use of surrogates, established as Evolvability Learning of Surrogates (EvoLS), for addressing optimization problems with computationally expensive objective function and presented in Chapter 6.

- Chapter 7 showcases two case studies of the proposed self-configurable memetic framework in solving real-world optimization problems: Symbiotic Evolution for *optimizing the OSS2 potential model* that describes $(H_2O)_n$ water clusters and EvoLS on *an aerodynamic car rear design* that involves highly computationally expensive CFD simulations.

- Last but not least, Chapter 8 concludes this thesis and outlines some potential future research directions on self-configurable search.

Chapter 2

Background

For many decades, optimization has been an important area that has been studied extensively by researchers from various disciplines, with many algorithms and implementations that are now made available and used in the different communities. In this chapter, a survey of several notable optimization approaches are presented as three core categories, namely, population-based (evolutionary) algorithms, individual-based algorithms and memetic algorithms. Subsequently, the four main design issues of memetic algorithm are discussed in details to highlight the core motivations towards self-configurable memetic search.

2.1 Population-based Methodologies

A remarkable increase of interest by researchers in the field of evolutionary computation have been witnessed in the recent decades. This form of algorithm operates on a population of individual solutions that evolves over time according to nature-inspired search operators.

As discussed by De Jong in [36], the unified model of the general evolutionary population-based algorithm is composed of the parent selection operator, stochastic variation operator(s) and replacement (or survival selection) operator, as outlined in Algorithm 1. In short, the selection and replacement schemes simulate the effect of "survival of the fittest" in nature selection. Replacement methods are similar to parent selection operators that determine which

individuals survive across generations. A great number of selection operators has been proposed in the literature [8, 50], including fitness proportional and stochastic universal selections [9], tournament selection [19] and Boltzmann selection [224]. The choice of selection operator largely depends on the desired selection pressure of the search. To some extent, the design of the stochastic variation operator(s) largely defines how population-based evolutionary algorithms differentiate from one another. In the subsequent subsections, some representative population-based algorithms, or more precisely, some notable forms of stochastic variation operators available in the literature are discussed.

Algorithm 1 Population-based Evolutionary Algorithm

1: **Initialize:** Generate the initial population;
2: **while** Stopping condition is not satisfied **do**
3: *Evaluate* individuals in the population;
4: *Select* parents pool $\mathbf{P}^t$ from the population;
5: *Apply stochastic variation* operator(s) $V(\mathbf{x}, \mathbf{P}^t)$ on $\mathbf{x} \in \mathbf{P}^t$ to generate offspring;
6: *Replace* a proportion or the entire population with the offsprings;
7: **end while**

2.1.1 Genetic Algorithms

Canonical crossover and mutation operators of Genetic Algorithm (GA) are first introduced by Holland in [65] and have since evolved, with many advanced forms of variation operators now made available, as reviewed comprehensively in [60].

For individual solutions that are represented as a vector of binary or real values, crossover operator creates offsprings from two or more parents by exchanging or recombining parts of their loci (location of the vector). Some of the common crossover schemes are one-point crossover, two-point crossover and uniform crossover [211]. For instance, in uniform crossover, the offspring are constructed from parents $\mathbf{x}_1$ and $\mathbf{x}_2$

$$\mathbf{x}_1 = \{\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)}, \ldots, \mathbf{x}_1^{(n)}\}$$

$$\mathbf{x}_2 = \{\mathbf{x}_2^{(1)}, \mathbf{x}_2^{(2)}, \ldots, \mathbf{x}_2^{(n)}\}$$

by exchanging independent ith locus of $\mathbf{x}_1^{(i)}$ and $\mathbf{x}_2^{(i)}$, $i = 1, 2, \ldots, n$ at probability p. For real-coded individual, arithmetical crossover is among some of the most widely used crossover operators [119]. In this case, the offspring are defined as $\mathbf{y}_1 = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ and $\mathbf{y}_2 = \lambda\mathbf{x}_2 + (1-\lambda)\mathbf{x}_1$ in which λ denotes a constant or varies with regard to the number of generations.

On the other hand, mutation operator randomly modifies the solution vector of an individual, subjected to a small mutation probability, thus introducing a pinch of randomness into the population. Similar to crossover operators, mutation operators are encoding scheme dependent, i.e., differing for binary or real-coded vector. Examples of real-coded mutation operators include the uniform mutation, Gaussian mutation and arithmetical mutation [119]. In Gaussian mutation, the real-coded vector of an individual is perturbed with a multivariate Gaussian distribution $\mathbf{r} \sim \mathbf{N}(\mathbf{0}, \sigma)$ to create offspring $\mathbf{x}' = \mathbf{x} + \mathbf{r}$.

Theoretical analysis of crossover and mutation in continuous domain can also be found in [157] and [158]. Advanced sampling techniques can also be considered as substitutions of the conventional crossover & mutation operators to generate the next population state. These include the Orthogonal GA [101], dynamic adaptation of crossover points [167, 207], parent-offspring analysis [206, 115], and evolvability analysis [12, 4].

2.1.2 Differential Evolution

The stochastic variation operators of Differential Evolution (DE) are designed by Storn and Price [185, 156] for solving continuous optimization problems. DE involves the phases of crossover and mutation [185, 163] to generate the subsequent population of individuals. The basic strategy of DE is described in [185] which is briefly discussed in what follows. For each parent individual $\mathbf{x}_i$, a mutated individual $\mathbf{u}_i$ is generated according to the following equation

$$\mathbf{u}_i = \mathbf{x}_{r_1} + F \times (\mathbf{x}_{r_2} - \mathbf{x}_{r_3}) \qquad \text{(Eq. 2.1)}$$

where indexes r_1, r_2, r_3 denote randomized and mutually different integers drawn from $S_i = \{1, 2, \ldots, n\} \setminus \{i\}$. F is a constant learning rate between $[0, 2]$ to control the amplification of

the differential variation $(\mathbf{x}_{r_2} - \mathbf{x}_{r_3})$. The mutated individual $\mathbf{u}_i$ is then recombined with the parent $\mathbf{x}_i$ through a form of uniform (binomial) crossover to produce offspring $\mathbf{y}_i$. More recent developments in DE algorithm design and applications are discussed in [213, 23, 214].

2.1.3 Particle Swarm Optimization

The core difference between Particle Swarm Optimization (PSO) and other population-based EAs is that each individual of the population is represented by a position vector $\mathbf{x}_i$ (i.e., design variables) and a velocity vector $\mathbf{v}_i$ [40, 41]. For each generation of the basic PSO [212, 147], the position and velocity of each individual is updated as follows:

$$\mathbf{v}_i^{t+1} = \omega \mathbf{v}_i^t + \eta_1 r_1 (\mathbf{x}_{pbest_i}^t - \mathbf{x}_i^t) + \eta_2 r_2 (\mathbf{x}_{gbest}^t - \mathbf{x}_i^t) \quad \text{(Eq. 2.2)}$$

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1} \quad \text{(Eq. 2.3)}$$

where $\mathbf{x}_i^t$ and $\mathbf{x}_i^{t+1}$ represent the current and updated position of an individual i, respectively. Similarly, the current and updated velocity of an individual are represented by $\mathbf{v}_i^t$ and $\mathbf{v}_i^{t+1}$, respectively. $\mathbf{x}_{pbest_i}^t$ and $\mathbf{x}_{gbest}^t$ are the best position found by individual i so far and the global best position found by the population so far, respectively. ω denotes an inertia weight to determine how much the current velocity is preserved. η_1 and η_2 are the acceleration constants that control the influence of $\mathbf{x}_{pbest_i}^t$ and $\mathbf{x}_{gbest}^t$ on the search. r_1 r_2 are random numbers drawn from uniform distribution $[0, 1]$. More advanced and recent versions of PSO are discussed in [98, 151, 99, 221, 33, 220].

2.1.4 Evolution Strategies

In Evolution Strategy (ES), each individual solution $\mathbf{x}_i$ is associated with a vector of mutation strengths $\mathbf{z}_i$. The mutation strengths is usually configured at a large value in the beginning so as to better explore the search space, and adaptively decreased as the search progresses. The

stochastic variation of Evolution Strategy constructs the next population state as a convolution of the current using some additive distribution $\mathbf{Z}$. For instance, the canonical Evolution Strategy as described in [13] uses an additive Gaussian [14] (or Cauchy/Lévy distributions [217, 219, 97]) with $\mathbf{y}_i = \mathbf{x}_i + \mathbf{z}_i$, where $\mathbf{y}_i$, $\mathbf{x}_i$ and $\mathbf{z}_i$ represent the offspring, corresponding parent, and random variable of distribution $\mathbf{Z}$ (named as mutation strength), respectively. Subsequently, crossover and mutation can be applied to both the decision variables and mutation strengths to create the offspring population, making ES a search with self-adaptation properties.

2.1.5 Covariance Matrix Adaptation Evolution Strategy

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) designed by Hansen *et al.* [53, 52] is an Evolution Strategy with the adaptation of the full covariance matrix of the normal mutation distribution $\mathbf{Z}$. At generation t, the offspring $\mathbf{y}$ in CMA-ES(μ, λ) are sampled independently according to

$$\mathbf{y}_i \sim \mathcal{N}\left(\mathbf{x}_W^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)}\right), i = 1, \ldots, \lambda \qquad \text{(Eq. 2.4)}$$

where $\mathcal{N}\left(\mathbf{x}_W^{(t)}, (\sigma^{(t)})^2\mathbf{C}^{(t)}\right)$ denotes a Gaussian distributed random vector with mean $\mathbf{x}_W^{(t)}$ and covariance matrix $\mathbf{C}^{(t)}$. The μ best offspring are recombined into the mean vector $\mathbf{x}_W^{(t+1)} = \sum_{i=1}^{\mu} w_i \mathbf{y}_i$ in which the positive weights w_i sum to one. The covariance matrix $\mathbf{C}^{(t)}$ and the step-size $\sigma^{(t)}$ of the normal distribution are then updated for the next generation [51]. Note that $\mathbf{x}_W^{(0)}$ and $\sigma^{(0)}$ are the user-specified parameters whose values depend on the given problem in hand. Since population size λ poses a significant impact on the performance of the algorithm, particularly when dealing with problems imbued with multi-modal properties [51], restart CMA evolution strategy with increasing population size *IPOP-CMA-ES* [5] are used to mitigate the difficulty of choice in the parameters in the absence of prior knowledge.

2.1.6 Probabilistic Search Algorithms

The stochastic variation operator in continuous probabilistic search creates offspring by sampling from an explicit probabilistic model **Z** of promising solutions instead of relying on classical crossover or mutation genetic operators [149, 26]. At each generation, the probabilistic model is constructed from a selected parent pool. In addition, prior knowledge about the problem can be expressed through distribution **Z**, such as multivariate Gaussian distributions $\mathcal{N}(\mu, \sum)$ or Bayesian networks, with assumptions on the dependencies among variables, as means of arriving at high quality search performances. Some of the popular works in this category include the Bayesian optimization algorithm (BOA) [148], Population-based Incremental Learning (PBIL) [11], Univariate Marginal Distribution Algorithm (UMDA) [120], and Estimation of Distribution Algorithms (EDA) [93, 92, 223].

2.2 Individual-based Methodologies

Individual-based search operator, also known better as local search, individual learning or lifetime learning, involves the process of searching for an improved solution (if it exists), starting from a given vector of decision variables [21]. The pseudo codes of a typical individual learning method is outlined in Algorithm 2. The search generally begins with a choice in the direction of movement. Subsequently a line search, trust region approach, or otherwise is performed to determine an appropriate next step. The process is repeated from the newly improved vector solution and the algorithm continues until a local minimum is found or the maximum computational budget allowable is reached. Individual learning methods can be distinguished by their choice of search directions and step length. Note that decisions can be made in a deterministic [45] or stochastic [184, 67] manner. For a detailed description of some representative methods in this category, the readers is referred to the Appendix B of the thesis.

Algorithm 2 Individual learning procedure $L(\mathbf{x})$

Start from some given point $x^{(1)}$.
Assign iteration counter $k = 1$.
while not converged **do**
  Calculate a search direction $D^{(k)}$, Determine an appropriate step length $\lambda^{(k)}$.
  Next search point $\mathbf{x^{(k+1)}} = \mathbf{x^{(k)}} + \lambda^{(k)} D^{(k)}$
  $k = k + 1$
end while

2.2.1 Deterministic Methods

In general, according to the use of derivative information in the procedure, continuous deterministic individual learning methods, largely referred to a group of numerical mathematical programming solvers in the literature, can be categorized as zeroth, first or second order method [168]. It is worth noting that the mathematical programming solvers in this category come with rigorous proofs of convergence to precise stationary points (i.e., 1^{st} order necessary condition [168, 103]), which exist in the form of minima, maxima or saddle points [1].

2.2.1.1 2nd-order Derivative Methods

This method requires the functional values, its first (partial) derivative vector and the second derivative matrix - the Hessian. Newton-Raphson method [82] denotes a typical example. It is based on the idea of approximating partial derivative with linear Taylor series expansion about a value. The method is powerful and simple to implement. It guarantees convergence to a stable point from any sufficiently close starting value.

2.2.1.2 1st-order Derivative Methods

In practice, the Hessian matrix may not be positive definite or may be difficult to calculate, thus making it infeasible to consider the second order methods. As a result, various techniques and variants of Newton method have been proposed to avoid calculating the exact Hessian matrix.

[1] In non-linear programming, the solution of interest lies in stationary point $\mathbf{x}$, where $\nabla F(\mathbf{x}) = 0$ or $\nabla F(\mathbf{x}) < \epsilon$, with ϵ denoting some arbitrarily small value.

Such approaches are known as first-order methods that employ the function values and first (partial) derivative vector to decide its moves, i.e., search directions and step length.

The second derivative of the objective function is then approximated using either finite differencing or via some iterative techniques. Methods in which the Hessian matrix is updated in an iterative manner are categorized generally as Quasi-Newton methods. The strategies of Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) are some notable examples of the Quasi-Newton method [153].

2.2.1.3 Zero-order Derivative Methods

Zero-order methods, also known as direct search techniques, use only the objective function values or the relative rank of objective values in the move operator to decide on the search structure. These are often very useful when the derivative information, including both Hessian and gradients of the function, are either unavailable, unreliable or expensive to compute [80]. Simplex search [127], pattern search [103], and methods with adaptive sets of search directions, such as Davies, Swann, and Campey with Gram-Schmidt orthogonalization (DSCG) [168], constitute to the three main categories of direct search.

2.2.2 Stochastic Methods

To deal with the scale-up in problem complexity, such as high multi-modality or noisy landscapes, stochastic refinement methods have been proposed to enhance search diversity in the neighborhood. While a majority of stochastic methods has been focused on solving combinatorial problems [67], there is an increasing trend of their presence in the continuous domain. Some of the early popular methods include simulated annealing [29, 216, 175, 198], tabu search [174, 24, 25], and the Solis Wets method [180] that operate on stochastic neighborhoods.

When first (partial) derivatives or gradients are not readily or cheaply available, especially on problems with computationally expensive objective and constraint functions, it may be ad-

vantageous to use approximated stochastic gradients than finite differencing [27]. The simultaneous perturbation stochastic approximation (SPSA) introduced by Spall in [183, 182] denotes one of the recent approaches with stochastic gradients.

Evolution strategy (ES) and covariance matrix adaptation evolution strategy (CMA-ES) that are configured to exploit local structure have also been demonstrated as reliable and highly competitive stochastic individual learning procedures for search [7, 123, 95]. Other notable approaches that received recent attention also include the real-coded hill-climbing crossover-based local search (XLS) [112, 134].

2.3 Memetic Methodologies

Memetic Algorithm (MA), also commonly known as hybrid EAs, Baldwinian EAs, Lamarkian EAs, or genetic local search, represents an emerging field that has attracted increasing research interest in recent times, with a growing number of publications appearing in a plethora of international journals and conference proceedings. The earliest form of Memetic Algorithms [176, 124, 125, 215] was first introduced as a marriage between population-based global search and individual learning, where the latter is often referred to as local search or a meme, capable of local refinement. Taking fundamentals from Darwinian principles of natural evolution and Dawkins notion of a meme, many modern evolutionary algorithms in the field of computational intelligence have been designed and crafted specifically for addressing particular problems or domains, and with significant success reported [102, 201, 68, 86, 139, 69, 141, 188, 195, 22, 189]. Memetic algorithms have been used successfully to solve a wide variety of engineering design problems and often shown to generate higher quality solutions more efficiently than canonical evolutionary algorithms [205, 102, 136, 201, 119, 68, 55, 161, 35]. A discussion on the different depictions of MAs inspired from Dawkins's theory of Universal Darwinism is provided in [131].

It is well established that the main purpose of designing a successful MA hybrid search lies in balancing well between generality (through stochastic variations) and problem specificity (through individual learning) [125, 57, 161, 132, 143]. In the next subsections, representative works in memetic research are discussed according to the four main design issues, namely level of hybridization, modes of learning, memetic operators and algorithmic parameters.

2.3.1 Level of Hybridization

It is worth noting that individual learning can be incorporated as a form of population initialization, *before* the population-based search, to enhance the search performance as contrast to a random initialization scheme. For *interleaved* hybrid procedures, on the other hand, individual learning is conducted after undergoing the stochastic variation or reproduction operator(s). In other hybrids, refinement is incorporated *after* the population-based search as a form of post-processing to fine-tune or improve the precision of the solution found by the EA.

An outline of the basic Memetic Algorithm is provided in Algorithm 3, where it can be seen that besides the stochastic operators, an individual learning phase (line 7) is included to refine the individuals.

Algorithm 3 Memetic Algorithm

1: *Generate* an initial population
2: **while** Stopping conditions are not satisfied **do**
3: *Evaluate* all individuals in the population
4: *Select* individuals for the parents pool $\mathbf{P}^t$ via selection operator $S(.)$
5: **for** each individual $\mathbf{x}$ in $\mathbf{P}$ **do**
6: *Evolve* $\mathbf{x}$ to offspring $\mathbf{y}$ according to stochastic variation operators $V(\mathbf{x}, \mathbf{P}^t)$
7: *Refine* $\mathbf{y}$ to $\mathbf{z}$ through individual learning operator $L(\mathbf{y})$
8: *Proceed* in the spirit of Lamarckian or Baldwinian learning
9: **end for**
10: **end while**

2.3.2 Modes of Learning

In the literature, two basic modes of individual learning (or inheritance schemes) are often discussed, namely, *Lamarckian* and *Baldwinian* learning [138] (line 8 in Algorithm 3). Lamarckian learning forces the genotype to reflect the result of improvement in individual learning by placing the locally improved individual back into the population to compete for reproductive opportunities [136, 68, 85]. In diverse contexts, Lamarckian memetic algorithms have also been used under the name of hybrid evolutionary algorithm, Lamarckian evolutionary algorithm, or genetic local search. Baldwinian learning, on the other hand, only alters the fitness of the individuals and the improved genotype is not encoded back into the population. Let $\mathbf{x}$ and $\mathbf{x}^{opt}$ denote the initial and improved solutions after undergoing refinement. Algorithmically, Lamarckian learning returns $(\mathbf{x}^{opt}, f(\mathbf{x}^{opt})$ to the population while Baldwinian learning return $(\mathbf{x}, f(\mathbf{x}^{opt})$ instead. Although Lamarck's theory of evolution has generated controversies and doubts from biology, the potentials and contributions of Lamarckian learning in computational evolutionary systems have been significant [70, 64]. It is worth emphasizing that most successful MAs to date are designed in the spirit of Lamarckian learning which exhibits clear advantage on problems in non-changing environments [118, 117, 208, 94]. On the other hand, Baldwinian learning is deemed as more appropriate for problems in dynamic or uncertain environments [145, 141, 166, 170].

2.3.3 Memetic Operators

In MA, an effective combination of search operators has to be found, both to combine different characteristics and strengths, and to design efficient communication mechanisms. Researchers have been exploring on various hybridizations of search operators towards the development and manual crafting of specialized algorithms that solve a specific problem or a set of problems effectively, using the domain knowledge of human expertise.

For example, the hybridizations of genetic operators with individual-based search methods have manifested as hybrid real-coded Genetic Algorithm with female and male differentiation (RCGA-FMD) [46], approximate probabilistic memetic framework based on GA-DSCG (APrMF) [132], and memetic algorithm with local search chaining (MA-LSCh-CMA) [122]. A review of different hybridizations of genetic algorithm with diverse individual learning strategies that employ gradient information is reported in [104]. On the other hand, accelerating differential evolution using an adaptive local search (DEahcSPX) [134] represents an example of combining DE's stochastic operators with local search to accelerate the search progress. Particle Swarm CMA-ES [126] denotes an example of hybrid MA in which CMA-ES is employed as an individual learning procedure with the population-based search PSO. Another notable example is the estimation of distribution algorithm (EDA) with ant-miner local search proposed in [2] for solving the nurse rostering problem.

Under the design issue of memetic operators, much effort has also been spent on the studies of adapting the appropriate individual learning procedures or otherwise often referred in the literature as local search(es) or meme(s) to employ, when solving specific problems or domains, by means of vertical inheritance [88, 131] and meme selection via heuristics [136, 31, 87]. It is worth noting that different learning procedures can be instances of the same strategy working on different parameters settings. Adaptation of local search parameters for real-coded memetic algorithm can be achieved through some heuristics, such as the adaptive hill-climbing XLS in [134], or via co-evolution mechanism [177, 152, 178]. For instance, in co-evolving memetic algorithm (COMA) proposed by Smith [177], neighborhood region associated with each individual solution in local search is adaptively updated based on a set of rules $\mathcal{R}$ represented by two bit strings. The bit-string rules are then evolved separately in another population using Darwinian evolutionary mechanisms (i.e. cooperative co-evolutionary system).

2.3.4 Algorithmic Parameters

From a survey of the field, the basic configuration of a memetic algorithm can be summarized as three core parameters

- Subset individuals of the population that should undergo individual learning
- Frequency of individual learning, which denotes how often individual learning is applied on the population
- The computational budget allocated for the individual learning phase defines how long each learning process should proceed for.

One of the conventional topics pertinent to the MA hybrid design is to identify *which individuals of the search population should undergo individual learning*, where for instance fitness and distribution-based strategies have been proposed by Land [91] and Nguyen *et al.* [133]. On the question pertaining to how often individual learning should be used, the effect of individual learning on MA search were investigated by Hart in [55] where various configurations at different stages of the search were considered. On the other hand, [90] suggested to apply learning on every individual when the computational complexity of the learning procedure is low. Schemes to adapt the *frequency of individual learning* based on search diversity and fitness distribution criteria have also been considered by Molina *et al.* in [112, 121].

To address the overall balance of stochastic variation and individual learning in search, a theoretical upper bound on *the computational budget* to allocate was proposed by Nguyen *et al.* in [132]. The bound provided a means to adapt various design issues of MA simultaneously, and at runtime, from which individuals that should undergo individual learning, to the amount of computational budget allocated for learning. In addition, the concept of local search chains to adapt the intensity of individual learning was also introduced in [123]. To alleviate the potentially high intensity and computational budget incurred in individual learning, especially when

dealing with real world complex problems plagued with computationally expensive objective functions, management schemes to adapt the use of approximation models or surrogates in lieu of the original objective functions [75] have also been considered.

Chapter 3

Landscape Analysis in Lamarckian Memetic Algorithms

To date little progress on the analytical study of Lamarckian MA in non-linear programming has been made [118, 117, 208, 146], in contrast to the greater effort reported on enhancing the understandings of Baldwinian MAs [62, 17, 144, 143]. Ironically, the majority of memetic algorithms that have experienced significant success on real-world problems happened to adopt the Lamarckian mode of learning in their design.

Given the limited progress made on revealing the working mechanisms of Lamarckian memetic algorithms in general non-linear programming and the intrinsic properties of commonly used complex benchmark problems, this chapter introduces the concepts of local optimum structure and provides a systematic study on the local optimum structure of several commonly used benchmark problems for non-linear programming to gain clues into the success or failure of MAs. Note that the empirical studies also serve to address the lack of analysis on benchmark problems that are commonly used in the literature.

The chapter is organized in the following manner. The influence of individual learning on the selection pressure in Lamarckian memetic algorithm is analyzed based on the concept of local optimum structure. Case studies on the local optimum structure of five continuous optimization benchmark problems of diverse properties are presented in Section 3.2. Finally, a brief conclusion of the chapter is provided in Section 3.3.

Nomenclature

$f(\mathbf{x})$	=	Objective or fitness function
$\mathbf{x}^*$	=	Global optimum
$\mathbf{x}^{(i)}$	=	*i-th* element of vector $\mathbf{x}$
$d(\mathbf{x}, \mathbf{y})$	=	Euclidean distance $\|\mathbf{x} - \mathbf{y}\| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$ between $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{\Psi}$	=	A set of local optimums
$E[.\|\mathbf{P}]$	=	Expectation of a measure conditioned to population $\mathbf{P}$
n	=	Number of dimensions
N	=	Population size
$S(.)$	=	Selection operator

3.1 Local Optimum Structure

In non-linear programming, the solution of interest lies in stationary point $\mathbf{x}$, where $\nabla F(\mathbf{x}) = 0$ or $||\nabla F(\mathbf{x})|| < \epsilon$, with ϵ denoting some arbitrarily small value. Stationary points exist in the form of minima, maxima or saddle points. In Lamarckian MAs, individual learning procedures based on mathematical solvers have been well established to converge to precise stationary points efficiently (i.e., 1^{st} order necessary condition [168, 103]). Practically, if $\mathbf{\Psi}$ is the complete local optimum set of a given problem and individual learning is performed until local optimality condition is satisfied, all populations of Lamarckian MA search, with the exception of the initial population, are subsets of $\mathbf{\Psi}$. The study on the search dynamics of Lamarckian MAs thus evolves around the set of local optimum solutions. For this reason, it is crucial to uncover the properties and characteristics of the local optimum set $\mathbf{\Psi}$ when analyzing Lamarckian MAs.

Since the search operators of Lamarckian MAs evolve around the set of local optimum solutions $\mathbf{\Psi}$, the local optimum structure or distribution of local optimum solutions represents one of the key property of the problem landscape to bring insights into the search and the algorithm that effects it. To study and analyze the influence of local optimum structure on the performance of Lamarckian MA search, particularly as a result of the stochastic selection

operator, the notion of *constructive* and *obstructive* local optimum structure and other basic definitions and theorems used are first introduced in what follows.

Let $\mathbf{P}^t = \{\mathbf{x}_k \in \mathbf{\Psi}\}_{k=1}^N$ denotes the population of N locally optimum individuals at generation t and $\mathbf{P}_S^t = S(\mathbf{P}^t) = \{\mathbf{x}_k^{'} \in \mathbf{\Psi}\}_{k=1}^N$ denotes the resultant population of optimum individuals after undergoing the stochastic selection operation.

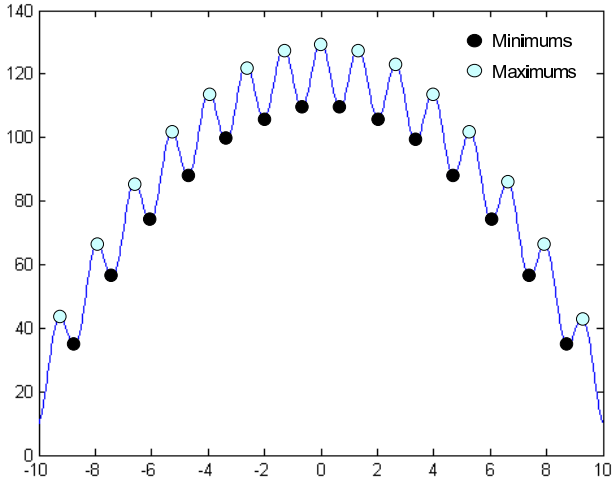
Definition 1: The selection progress rate of the search θ_S at generation t is defined as the expected change in *distance to global optimum* as a result of the selection operator $S(.)$.

$$\begin{aligned}\theta_S(\mathbf{P}^t) &= D - E[D'|\mathbf{P}^t] \\ &= \frac{1}{N} \times \sum_{k=1}^{N} d(\mathbf{x}_k, \mathbf{x}^*) - \frac{1}{N} \times \sum_{k=1}^{N} E[d(\mathbf{x}_k^{'}, \mathbf{x}^*)|\mathbf{P}^t] \end{aligned} \quad \text{(Eq. 3.1)}$$

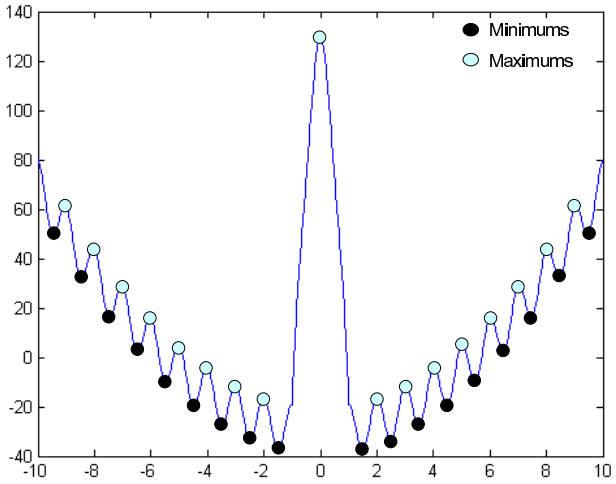
Definition 2: A *'constructive'*/ *'obstructive'* local optimum structure exhibits property of local optima with *'increasing'*/ *'decreasing'* fitness quality $f(\mathbf{x})$ when being closer to the global optimum, i.e., $d(\mathbf{x}, \mathbf{x}^*)$.

Let $\mathbf{\Psi} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_M\}$, where $|\mathbf{\Psi}| = M$, denote the finite set of local optimum solutions. Let d_i and f_i be the abbreviations for $d(\mathbf{x}_i, \mathbf{x}^*)$ and $f(\mathbf{x}_i)$, respectively. Without loss of generality, members of set $\mathbf{\Psi}$ are sorted in an ascending order of distance to the global optimum $d(\mathbf{x}_i, \mathbf{x}^*)$ such that $0 < d_1 \leq d_2 \leq d_3 \ldots \leq d_M$. For a maximization problem, the definition implies $f_1 \geq f_2 \geq f_3 \ldots \geq f_M$ for *constructive* local optimum structure and $f_1 \leq f_2 \leq f_3 \ldots \leq f_M$ for *obstructive* correlated structure of local optimums. Illustrative examples of problems with a fitness landscape imbued with properties of *constructive* and *obstructive* correlated local optimum structure are depicted in Figures 3.1.a and 3.1.b, respectively.

In what follows, an analysis on the search mechanisms of the stochastic selection operator in Lamarckian MAs, by taking into considerations the local optimum structure of a problem landscape, is presented. In particular, the well-established stochastic fitness-proportional selection scheme [50], which represents one of the selection operators commonly used in evolutionary computation, is analyzed.



3.1.a: Problem Landscape with Constructive Local Optimum Structure



3.1.b: Problem Landscape with Obstructive Local Optimum Structure

Figure 3.1: Illustrations of ‘constructive’/ ‘obstructive’ landscapes in maximization problem

For a Lamarckian MA with stochastic fitness-proportional selection operator, the probability of individual $\mathbf{x}_k \in \mathbf{P}^t = \{\mathbf{x}_k\}_{k=1}^N$ to survive in the reproduction pool, can be derived as $\frac{f_k}{\sum_{j=1}^N f_j}$. The selection progress rate can then be derived as

$$\begin{aligned}\theta_S(\mathbf{P}^t) &= \frac{\sum_{k=1}^N d_k}{N} - \frac{1}{N} \times \sum_{k=1}^N E[d(\mathbf{x}_k^{'}, \mathbf{x}^*)|\mathbf{P}^t] \\ &= \frac{\sum_{k=1}^N d_k}{N} - \frac{1}{N} \times N \times (\frac{\sum_{u=1}^N f_u d_u}{\sum_{j=1}^N f_j}) \\ &= \frac{\sum_{k=1}^N d_k}{N} - \frac{\sum_{u=1}^N f_u d_u}{\sum_{j=1}^N f_j} \\ &= \frac{(\sum_{j=1}^N f_j)(\sum_{k=1}^N d_k) - N \times \sum_{u=1}^N f_u d_u}{N \sum_{j=1}^N f_j}\end{aligned} \quad \text{(Eq. 3.2)}$$

Without loss of generality, the progress rate of the stochastic selection operator on maximization problems with fitness landscape containing a *constructive* or *obstructive* local optimum structure can be inferred from Eq. 3.2, using the Chebyshev's sum inequality theorem [54]. Note that on *constructive* local optimum structure, two sequences $\{f_j\}_{j=1}^N$ and $\{d_k\}_{k=1}^N$ are of an inverted order. According to the Chebyshev's sum inequality theorem[1], we have

$$\begin{aligned}(\sum_{j=1}^N f_j)(\sum_{k=1}^N d_k) &\geq N \times \sum_{u=1}^N f_u d_u \\ (\sum_{j=1}^N f_j)(\sum_{k=1}^N d_k) - N \times \sum_{u=1}^N f_u d_u &\geq 0\end{aligned} \quad \text{(Eq. 3.3)}$$

Since the left-hand side of Eq. 3.3 is the numerator of $\theta_S(\mathbf{P}^t)$ in Eq. 3.2, we have $\theta_S(\mathbf{P}^t) \geq 0$. Thus, for a maximization problem fitness landscape containing a *constructive* correlated local optimum structure, the selection progress rate is positive, i.e., $\theta_S(\mathbf{P}^t) \geq 0, \forall \mathbf{P}^t$.

Conversely, for a maximization problem fitness landscape containing *obstructive* correlated local optimum structure, the selection progress rate can be shown to be negative, i.e., $\theta_S(\mathbf{P}^t) \leq$

[1]**Chebyshev's sum inequality theorem**: considering two sequences, $\{f_j\}_{j=1}^N$ and $\{d_k\}_{k=1}^N$ of the same order, $f_1 \leq f_2 \leq \ldots \leq f_N$ and $d_1 \leq d_2 \leq \ldots \leq d_N$, then $(\sum_{j=1}^N f_j)(\sum_{k=1}^N d_k) \leq N \times \sum_{u=1}^N f_u d_u$ Similarly, if two sequences are of an inverted order, $f_1 \geq f_2 \geq \ldots \geq f_N$ and $d_1 \leq d_2 \leq \ldots \leq d_N$, then $(\sum_{j=1}^N f_j)(\sum_{k=1}^N d_k) \geq N \times \sum_{u=1}^N f_u d_u$

$0, \forall \mathbf{P}^t$. Similarly, note that on obstructive local optimum structure, two sequences $\{f_j\}_{j=1}^N$ and $\{d_k\}_{k=1}^N$ are of the same order. According to the Chebyshev's sum inequality theorem, we have

$$(\sum_{j=1}^{N} f_j)(\sum_{k=1}^{N} d_k) \leq N \times \sum_{u=1}^{N} f_u d_u$$
$$(\sum_{j=1}^{N} f_j)(\sum_{k=1}^{N} d_k) - N \times \sum_{u=1}^{N} f_u d_u \leq 0 \qquad \text{(Eq. 3.4)}$$

Since the left-hand side of Eq. 3.4 is the numerator of $\theta_S(\mathbf{P}^t)$ in Eq. 3.2, we have $\theta_S(\mathbf{P}^t) \leq 0$.

To summarize, the stochastic selection operator in MA exhibits the property of "pulling" the population towards the global optimum on problem landscapes imbued with a *constructive* correlated local optimum structure, thus advancing the search towards the global optimum. In contrast, the stochastic selection operator exhibits the effect of "pushing" the population away from the global optimum on problem landscapes with an *obstructive* correlated optimum structure.

3.2 An Analysis on the Local Optimum Structure of Representative Benchmark Problems

In non-linear programming, a number of complex benchmark minimization problems having diverse properties are commonly used [186]. Although the set of problems has been used extensively for benchmarking evolutionary [210, 101, 218] and memetic algorithms [55, 136, 130, 131, 134], little effort has been made to reveal their properties in the context of memetic algorithms thus far, particularly on the local optimum structure of these benchmark problems.

In this section, a systematic analysis on the local optimum structure of five commonly used continuous parametric benchmark problems is presented. These benchmarks including the Ackley, Rastrigin, Griewank, Weierstrass and Rosenbrock functions, represent classes of multimodal, 'epistatic'/ 'non-epistatic' optimization problems [136], as detailed and summarized in Appendix A.

To study the local optimum structure of the benchmark problems, the random multi-start individual learning (local search) which forms the baseline optimization algorithm is used to search on them [2]. In the experiments, the sets of local optimum solutions for the five problems [3] are attained using the popular Davidon-Fletcher-Powell (DFP) strategy, which belongs to the family of the quasi-Newton approach. The stopping criteria of DFP is defined by the Cauchy's convergence test [4] (i.e. $|\mathbf{x}_{n+1} - \mathbf{x}_n| \leq \epsilon$ for $N > N_0$ and precision ϵ set to 1E-8). In each run, m solutions $\mathbf{x}$ are randomly sampled to undergo individual learning based on DFP to arrive at the corresponding local optimums. Here, m is configured to 4000 and the local optimums attained are analyzed to reveal the properties of the original problem fitness landscape and the transformed landscape due to individual learning.

Fitness distance correlation (FDC) analysis is a common method to reveal the correlated structure of a landscape [190, 79, 118]. Correlation coefficient measure, ϱ, gives the level of correlation between the fitness and distance to global optimum, which is defined by

$$\varrho(f, d) = \frac{Cov(d, f)}{\sigma(d) \times \sigma(f)} \qquad \text{(Eq. 3.5)}$$

Alternatively, ϱ can be estimated from the set of m solutions $\{(d_i, f_i)\}_{i=1}^{m}$ as follows

$$\varrho(f, d) \approx \frac{1}{\sigma(f)\sigma(d)} \frac{1}{m} \sum_{i=1}^{m} (f_i - \overline{f})(d_i - \overline{d}), \qquad \text{(Eq. 3.6)}$$

where $\overline{f}$ and $\sigma(f)$ refer to the mean and standard deviation, respectively. Similar notations are used for $\overline{d}$ and $\sigma(d)$. Using Eq. 3.6, the statistical measure ϱ_L of the transformed problem landscape can be estimated from the set of m local optimums. In addition to the fitness distance correlation coefficient ϱ, the scatter plot $(f(\mathbf{x}), d(\mathbf{x}, \mathbf{x}^*))$ of fitness and distance to global

[2] The random multi-start individual learning (local search) algorithm is the simplest form of stochastic algorithm that uses a basic random search as a global optimizer. Being an unbiased global optimizer, it allows an accuracy revelation on the local optimum structure of the problem landscape

[3] $\mathbf{\Psi}_{Ackley}$, $\mathbf{\Psi}_{Rastrigin}$, $\mathbf{\Psi}_{Griewank}$,$\mathbf{\Psi}_{Rosenbrock}$ and $\mathbf{\Psi}_{Weierstrass}$

[4] Cauchy's convergence test for sequence $\{\mathbf{x}_i\}$ can be described as: for every $\epsilon > 0$, there is a number N, such that for all $n, m > N$ holds $\|\mathbf{x}_m - \mathbf{x}_n\| < \epsilon$

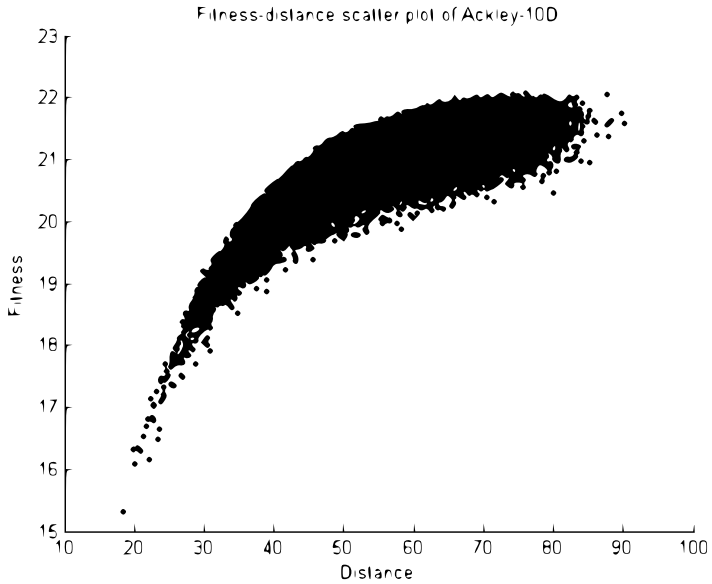
Benchmark Problem	**Original Landscape Structure ($\varrho \pm \sigma$)**	**Transformed Landscape or Local Optimum Structure ($\varrho_L \pm \sigma_L$)**
Ackley	$0.780676 \pm 6.539E-3$ *Weak Constructive Correlation*	$0.920672 \pm 5.791E-3$ *Strong Constructive Correlation*
Rastrigin	$0.719241 \pm 6.925E-3$ *Poor correlation*	$0.994019 \pm 0.306E-3$ *Strong Constructive Correlation*
Griewank	$0.994486 \pm 0.247E-3$ *Strong Constructive Correlation*	$0.995402 \pm 0.297E-3$ *Strong Constructive Correlation*
Rosenbrock	$0.621473 \pm 9.374E-3$ *Poor correlation*	$0.999912 \pm 0.228E-3$ *Strong Constructive Correlation*
Weierstrass	$0.877178 \pm 3.749E-3$ *Weak Constructive Correlation*	$0.709783 \pm 8.725E-3$ *Poor correlation*

Table 3.1: Mean (ϱ_L/ϱ) and Variance (σ_L/σ) of Landscape Structure Correlation

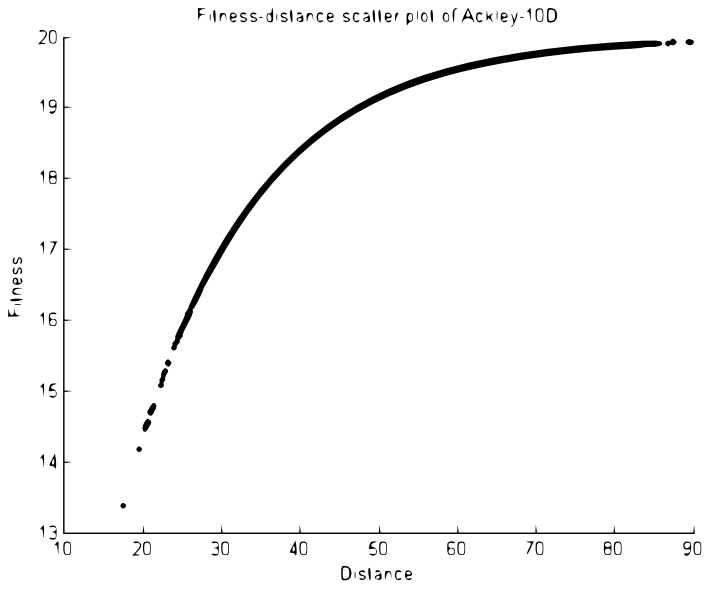
optimum, which can aid visual analysis on the correlation property of a problem landscape, is also considered here.

Using the results on 10-dimensional benchmark problems across 30 independent runs from the experimental study, the fitness distance scatter plots and correlation coefficients, i.e., ϱ_L and ϱ, are presented in Figures 3.2-3.6 and Table 3.1, respectively. From the scatter plots in Figures 3.2 and 3.3, strong *constructive* correlated structure of local optimums are observed on both Ackley and Rastrigin functions. The left and right panels of the figures show the scatter plot of local optimums and initial sampled points, or the transformed and the original landscapes, respectively. Further, the results in Table 3.1 indicate higher mean-variance structure correlation value for the transformed fitness landscape than the original fitness landscape on Ackley ($\varrho_L = 0.920672 \pm 5.791E-3 > \varrho = 0.780676 \pm 6.539E-3$) and Rastrigin ($\varrho_L = 0.994019 \pm 0.306E-3 > \varrho = 0.719241 \pm 6.925E-3$). This highlights the significant impact of individual learning on the search space, where the phenomenon of reinforcement in fitness distance correlation is observed in the transformed fitness landscape. [5]

[5]On a minimization problem, the local optimum solutions of the landscape is considered to possess strong *constructive* correlation property if *FDC-L* $\varrho_L \approx 1$, i.e., the closer an individual is to the global optimum, the lower (better) is its fitness value. On the other hand, the local optimum structure is considered to display strong obstructive correlation property if *FDC-L* $\varrho_L \approx -1$.

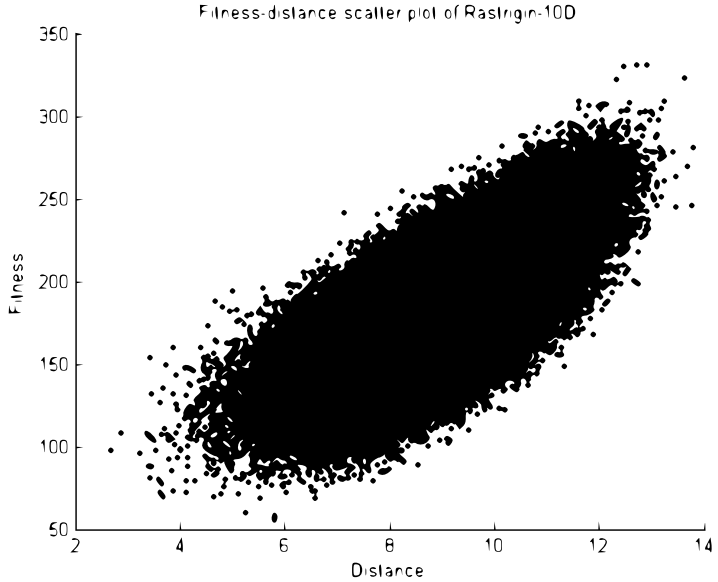


3.2.a: Ackley(10D)

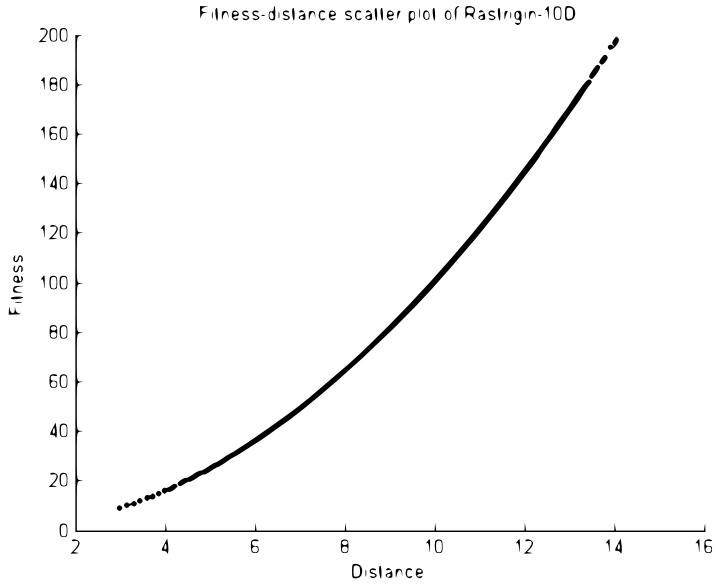


3.2.b: L-Ackley(10D)

Figure 3.2: Fitness distance scatter plots of Ackley function



3.3.a: Rastrigin(10D)



3.3.b: L-Rastrigin(10D)

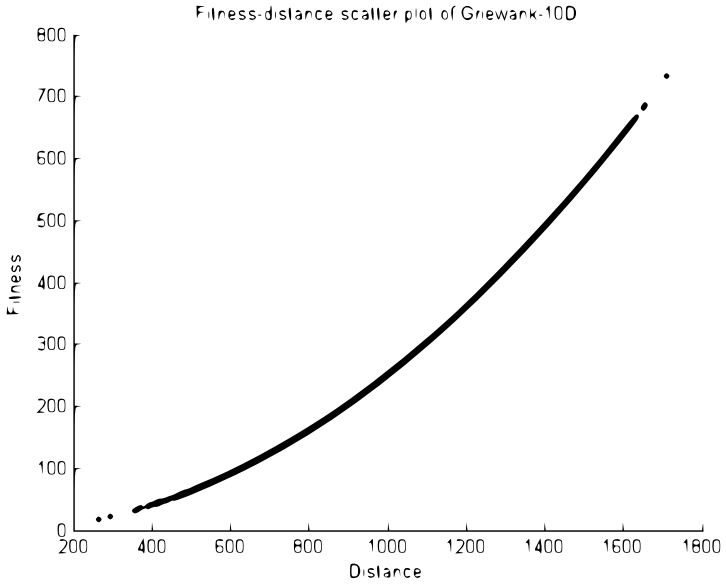
Figure 3.3: Fitness distance scatter plots of Rastrigin function

On the Griewank function, however, the scatter plot in Figure 3.4 shows a similar degree of *constructive* correlation structure in both the original and transformed landscape, i.e., ($\varrho_L = 0.995402 \pm 0.297E-3 \approx \varrho = 0.994486 \pm 0.247E-3$). It is likely that the decreased level of ruggedness in Griewank function for increasing problem dimension that led to the similar level of structure correlations observed.

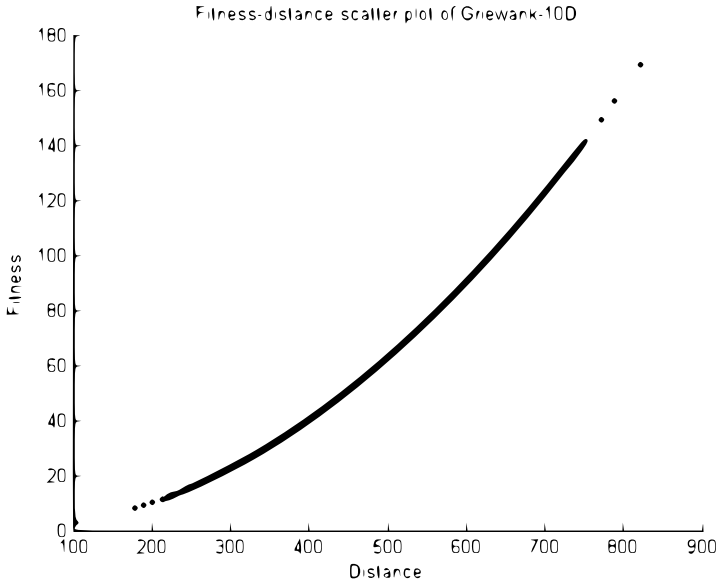
For the 10-dimensional Rosenbrock function, 3 stationary points have been found in the experiment study, which is consistent to the results reported in [172]. The scatter plot in Figure 3.5 indicates the presence of *constructive* correlated structure among the local optimums in the transformed Rosenbrock landscape,which is significantly higher than that exhibited by the solutions of the original landscape ($\varrho_L = 0.999912 \pm 0.228E-3 \gg \varrho = 0.621473 \pm 9.374E-3$).

Last but not least, based on the scatter plot in Figure 3.6, Weierstrass function is found to possess a weak *constructive* correlated structure of local optimums, even though visually the original problem landscape may appear to be relatively structured. The resulting *FDC* coefficient ϱ of Weierstrass, based on original fitness landscape is found to be higher ($\varrho = 0.877178 \pm 3.749E-3$) than the local optimum structure correlation ($\varrho_L = 0.709783 \pm 8.725E-3$). Interestingly, the Weierstrass function thus possesses an unique property that differs from the other benchmark functions considered.

From the structure correlation profile of the five representative benchmark problems summarized in Table 3.1, it is worth noting that most of the problems investigated and analyzed here displayed increased *constructive* correlations in the transformed landscape than the original fitness landscape, i.e., $\varrho_L > \varrho$. Based on the proof presented in Section 3.1, this increased *constructive* structure correlation of the transformed landscapes is expected to benefit the Lamarckian MA due to the strong "pulling" effect of the stochastic selection operator on the population of individuals, thus enhancing the search towards the global optimum. On the other hand, it is also revealed that the degraded *constructive* correlated structure of the Weierstrass transformed landscape, i.e., $\varrho_L < \varrho$, would lead to poorer search performance when

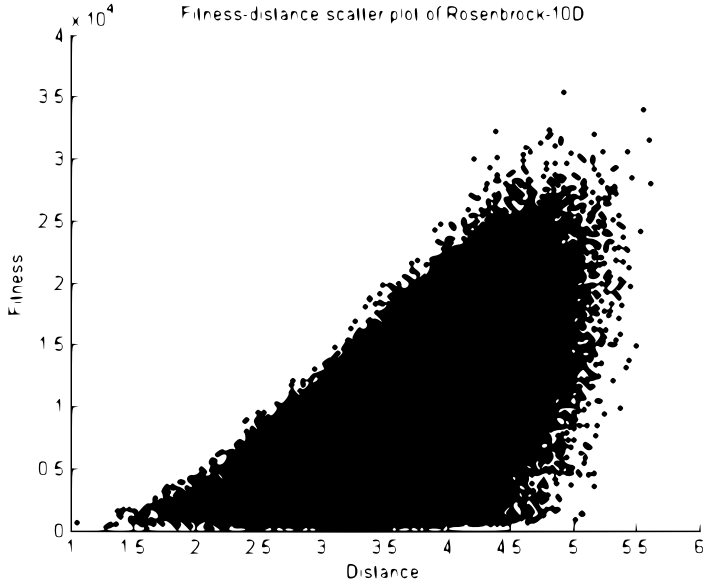


3.4.a: Griewank(10D)

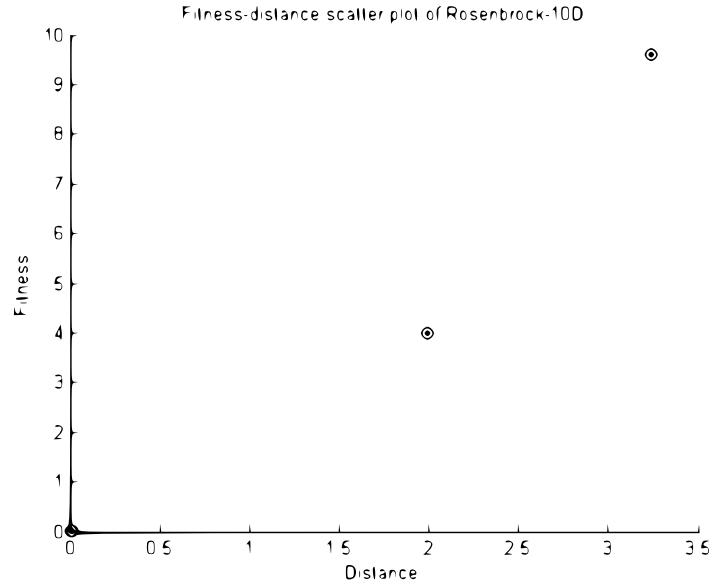


3.4.b: L-Griewank(10D)

Figure 3.4: Fitness distance scatter plots of Griewank function

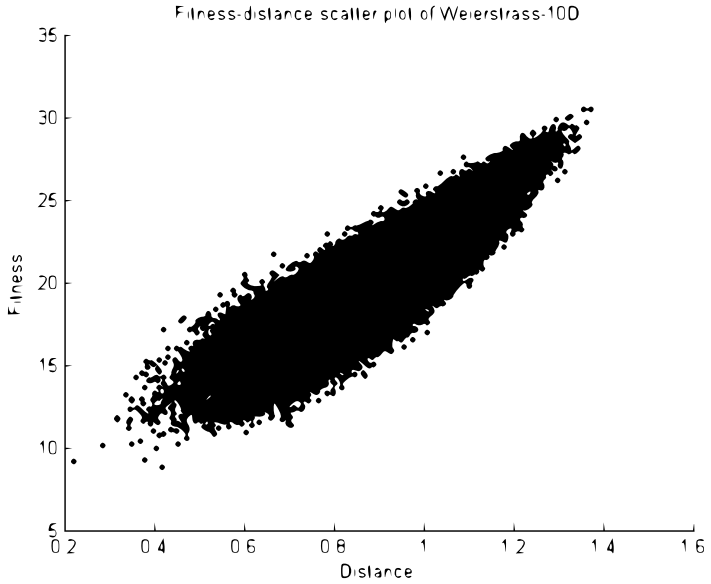


3.5.a: Rosenbrock(10D)

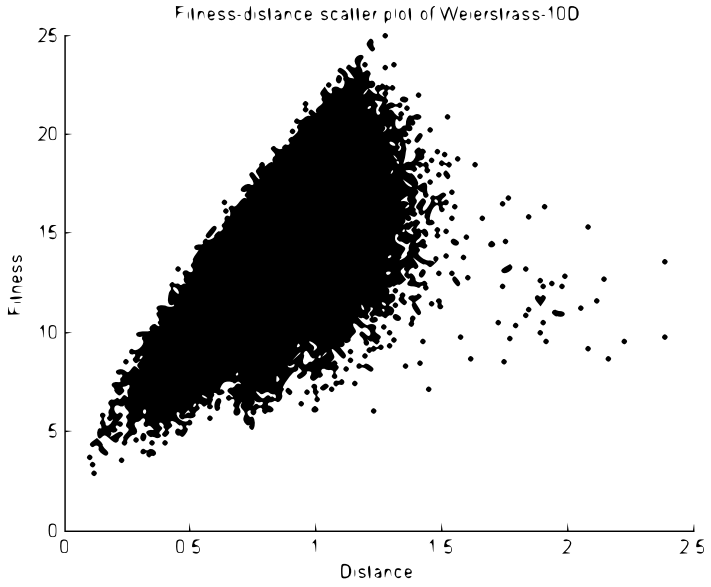


3.5.b: L-Rosenbrock(10D)

Figure 3.5: Fitness distance scatter plots of Rosenbrock function



3.6.a: Weierstrass(10D)



3.6.b: L-Weierstrass(10D)

Figure 3.6: Fitness distance scatter plots of Weierstrass function

using MA-DFP as compared to a simple random search. In such event, the DFP individual learning should not be used.

3.3 Conclusions

In this chapter, the notion of '*constructive*'/ '*obstructive*' local optimum structure has been introduced and defined as a useful property for analyzing the transformed fitness landscape brought about by the individual learning procedure of MA. Subsequently, the influence of individual learning on the selection pressure in Lamarckian memetic algorithm is analyzed to demonstrate how memetic algorithm benefits from the individual learning process in advancing the search towards the global optimum. The results of improved correlated *constructive* structure observed in most of the transformed landscape relative to the original on typical benchmark problems thus highlight the positive influence of individual learning that contributes to the success of MAs recently reported in the literature.

Chapter 4

Evolvability

The image of modern memetic algorithms in computational intelligence can be established as a *symbiosis* of stochastic variation and individual learning, working in sync on the given problem in hand. In most research work on adaptive evolutionary algorithms, the symbiosis of stochastic variation and individual learning are generally studied as separate independent entities, which is evident in the works of Hinterding *et al.* [61] and Ong *et al.* [138]. Few has considered formal modelling on the combined behaviors of stochastic variations and individual learning in search. Based on the insights gained on the effect of individual learning on Lamarckian MA search in Chapter 3, this chapter proceeds to formalize and study the symbiosis [1] or synergy of search operators in memetic search, which perhaps has the greatest influence on search performance but has remained yet to be formally investigated with sufficient rigor.

Chapter 4 is organized in the following manner. The concept of symbiotic search profile and connectivity structure for analysis of MAs are first presented in Section 4.1. The notion of *evolvability* as the basis for assessment is also introduced to quantify the degree of synergy between the stochastic variation and individual learning operators. To bring about new insights into the symbiosis of search operators on Lamarckian MA design, investigations on the search mechanisms of two unique search profiles on the benchmark problems by analyzing

[1] In biology, the term '*Symbiosis*' commonly refers to the interactions of two dissimilar organisms or species, living together [38, 209].

their *evolvability* measures is further proceeded in Section 4.2. Finally, a brief conclusion of the chapter is provided in Section 4.3.

4.1 Evolvability of Symbiotic Search Profile

In what follows, we proceed to study the synergy between the stochastic variation and individual learning operators, labelled here as symbiotic search profile, using the proposed notion of *evolvability*. A search profile, ID, is defined by a pair of stochastic variation $V(\mathbf{x}, \mathbf{P}^t)$ and individual learning $L(\mathbf{x})$, i.e., $ID = (V, L)$.

The "*evolvability*" of symbiotic search profile is introduced here as the basis for assessment. Since the term "*evolvability*" has been used in different contexts[2], it is worth highlighting that here the notion of *evolvability* in this work is generalized from that of learnability in machine learning [197]. Here the evolutionary process is regarded as "*evolvable*" on a given optimization problem if the progress in search performance is observed for some moderate number of generations. Hence *evolvability* of a symbiotic search profile is referred here to the propensity of the stochastic variation and individual learning in creating viable, or "potentially favorable" individuals that leads towards the global optimum.

The *evolvability* of search profile ID for a given solution $\mathbf{x}$ at generation t, denoted here as $Ev_{ID}(\mathbf{x}, t)$, may be defined using a variety of criteria or cost functions. An intuitive and common quantitative measure of *evolvability* in search would be to consider the expected fitness improvement (FI) with respect to the expected computational cost (C) incurred. Without loss of generality, minimization problems are considered throughout this chapter. The *evolvability* of symbiotic search profile $Ev_{ID}(\mathbf{x}, t)$, for a potential solution $\mathbf{x}$ in population $\mathbf{P}^t$, may be formulated in terms of (FI) and (C) as follows:

[2] In [203], "evolvability" is defined as the genome's ability to produce adaptive variants when acted upon by the genetic system. Others have generally refer the term to the ability of stochastic or random variations to produce improvement for adaptation to happen [138].

$$Ev_{ID}(\mathbf{x},t) = \frac{FI(\mathbf{x},t)}{C(\mathbf{x},t)} \tag{Eq. 4.1}$$

$$\begin{aligned} FI(\mathbf{x},t) = E[\triangle f|\mathbf{P}^t,\mathbf{x}] &= \int\int \triangle f \times P(\mathbf{y}|\mathbf{P}^t,\mathbf{x})P(\mathbf{z}|\mathbf{y})d\mathbf{y}d\mathbf{z} \\ &= \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times \int_{\mathbf{z}} \triangle f \times P(\mathbf{z}|\mathbf{y})d\mathbf{z}d\mathbf{y} \end{aligned} \tag{Eq. 4.2}$$

$$\begin{aligned} C(\mathbf{x},t) = E[\triangle C|\mathbf{P}^t,\mathbf{x}] &= \int\int \triangle C \times P(\mathbf{y}|\mathbf{P}^t,\mathbf{x})P(\mathbf{z}|\mathbf{y})d\mathbf{y}d\mathbf{z} \\ &= \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times \int_{\mathbf{z}} \triangle C \times P(\mathbf{z}|\mathbf{y})d\mathbf{z}d\mathbf{y} \end{aligned} \tag{Eq. 4.3}$$

where

$f(\mathbf{x})$	=	Objective or fitness function
$\mathbf{P}^t$	=	Reproduction pool at generation t
$\mathbf{x}$	=	Solution individual in the reproduction pool, i.e., $\mathbf{x} \in \mathbf{P}^t$
$\mathbf{y}$	=	Offspring produced by stochastic variation, i.e., $\mathbf{y} = V(\mathbf{x}, \mathbf{P}^t)$
$\mathbf{z}$	=	Resultant individual from offspring through individual learning, i.e., $\mathbf{z} = L(\mathbf{y})$
$P(\mathbf{z}\|\mathbf{y})$	=	Density function of individual learning L applied on offspring $\mathbf{y}$
$P(\mathbf{y}\|\mathbf{P}^t,\mathbf{x})$	=	Density function of variation operator V applied on parent $\mathbf{x}$
$\triangle f$	=	Fitness improvement $f(\mathbf{x}) - f(\mathbf{z})$ on minimization problem
$\triangle c$	=	Computational cost or the number of function evaluations incurred by variation and individual learning operators.

Let $Gain(\mathbf{y},\mathbf{x}) = \int_{\mathbf{z}} \triangle f \times P(\mathbf{z}|\mathbf{y})d\mathbf{z}$ and $Cost(\mathbf{y},\mathbf{x}) = \int_{\mathbf{z}} \triangle c \times P(\mathbf{z}|\mathbf{y})d\mathbf{z}$ then Eq. 4.2 and Eq. 4.3 becomes

$$FI(\mathbf{x},t) = \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times Gain(\mathbf{y},\mathbf{x})d\mathbf{y} \tag{Eq. 4.4}$$

$$C(\mathbf{x},t) = \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times Cost(\mathbf{y},\mathbf{x})d\mathbf{y} \tag{Eq. 4.5}$$

With the substitution of Eq. 4.4 and Eq. 4.5 into Eq. 4.1, we arrive at

$$Ev_{ID}(\mathbf{x},t) = \frac{\int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times Gain(\mathbf{y},\mathbf{x})d\mathbf{y}}{\int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t,\mathbf{x}) \times Cost(\mathbf{y},\mathbf{x})d\mathbf{y}} \tag{Eq. 4.6}$$

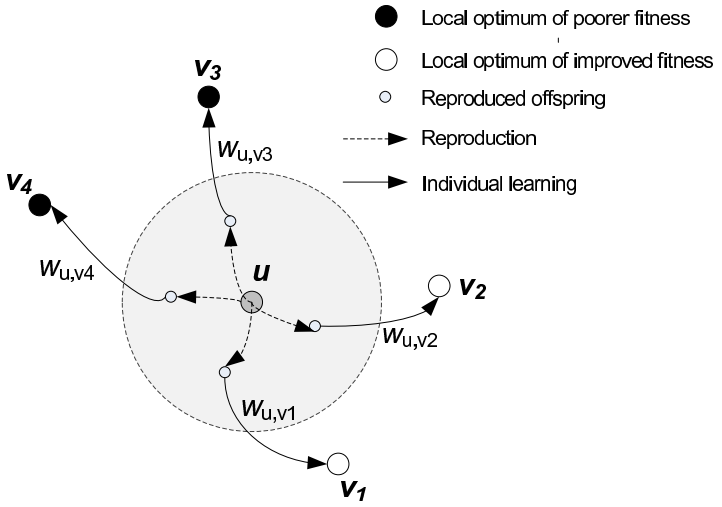


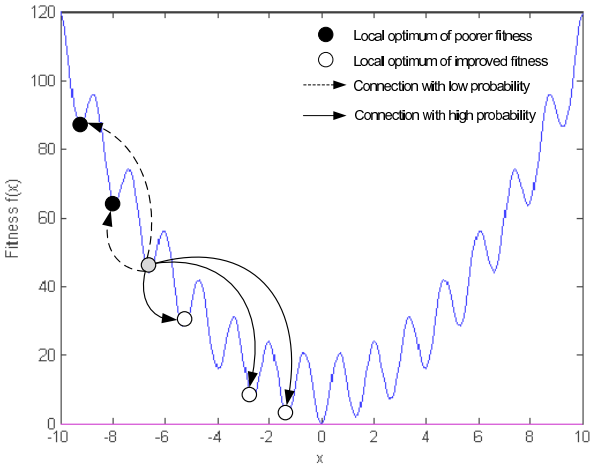
Figure 4.1: Connectivity of local optimums

By this definition, *evolvability* measure of a symbiotic search profile $Ev_{ID}(\mathbf{x}, t)$ indicates the expectation of how a developed offspring, after undergoing individual learning, is improved over its initial parent with the computational cost taken into consideration.

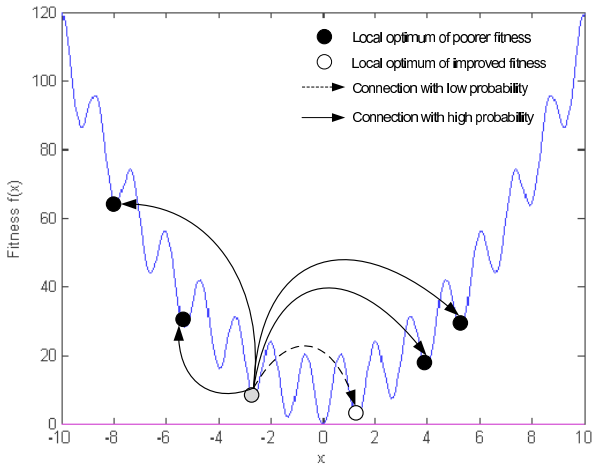
Next, a special case of MA in which the individual learning strategy is performed until convergence to local optimum (under sufficient computational budget) is considered and discussed for further illustration. Subsequently, the solver's behavior that emerged from the stochastic variation and individual learning mechanisms in search can be modelled as a directed graph that defines the *connectivity probability* of local optimum solutions, more precisely, the probability of "jumping" from one local optimum to another as determined by the symbiotic search profile used [94]. Local optimum $\mathbf{u}$ is connected to local optimum $\mathbf{v}$ if and only if $\mathbf{v}$ is reachable from $\mathbf{u}$, i.e., the probability of $\mathbf{v} = L(V(\mathbf{u}))$ is non-zero, or $P(\mathbf{v} = L(V(\mathbf{u}))) > 0$. The connectivity of local optimum solutions is then modeled as a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. Vertex $\mathbf{V}$ represents the local optimum solutions (i.e. $\mathbf{V} = \mathbf{\Psi}$). A directed edge $e_{\mathbf{u},\mathbf{v}}$ represents that $\mathbf{v}$ is reachable from $\mathbf{u}$ via the stochastic variation V and individual learning L operators. Figure 4.1 depicts possible connections between a local optimum $\mathbf{u}$ to other local optimum solutions of a improved or inferior quality, as illustrated by vertex $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3, \mathbf{v}_4$, respectively.

For minimization problems, positive or high *evolvability* measure on a local optimum $\mathbf{x}$, i.e., $Ev_{ID}(\mathbf{x}, t) \geq 0$, indicates a *constructive* connectivity of the given local optimum. On the other hand, negative *evolvability* measure of a local optimum $\mathbf{x}$, i.e., $Ev_{ID}(\mathbf{x}, t) < 0$, defines an *obstructive* connectivity. In other words, a *'constructive'*/ *'obstructive'* connectivity indicates that a local optimum is more likely to connect to other local optimum solutions of 'improved'/ 'inferior' quality. Note that the concepts of *'constructive'*/ *'obstructive'* connectivity does not involve the notion of distance to global optimum used in the concepts of *'constructive'*/ *'obstructive'* local optimum structure, presented in Chapter 3. As the distance to global optimum of solution $\mathbf{x}$ is generally not available in practice, here the fitness function $f(\mathbf{x})$ is used instead to evaluate the solution quality. In addtion, it is worth noting that the connectivity characteristic of a local optimum, i.e., being *'constructive'* or *'obstructive'*, also depends on the time factor t.

Examples on *constructive* and *obstructive* connectivity of local optimum solutions are depicted in Figure 4.2.a and 4.2.b, respectively. Note that the 'solid'/ 'dotted' lines in the figures refers to the 'high'/ 'low' probability connections of local optimums. For the *constructive* connectivity property illustrated in Figure 4.2.a, as most of the local optimums $\mathbf{x}_k$ of an optimization problem possess a high probability of connecting to improved quality solutions at low learning expense,i.e., high or positive *evolvability* $Ev_{ID}(\mathbf{x}, t)$, Lamarckian MAs will search effectively and efficiently towards the global optimum. On the other hand, for problems imbued with *obstructive* connectivity where most local optimums are likely to connect to low-quality solutions at the expenses of high learning costs, i.e., negative *evolvability* $Ev_{ID}(\mathbf{x}, t)$, a limited progress rate can be expected. In the extreme case, where $\mathbf{x} = L(V(\mathbf{x})) = \mathbf{z}$, no search improvement can be achieved, which leads to the well-known problem of premature convergence or stagnation in MA [208].



4.2.a: Constructive Connectivity of Local Optimums



4.2.b: Obstructive Connectivity of Local Optimums

Figure 4.2: Illustrations of ‘constructive’/ ‘obstructive’ connectivity in minimization problem

4.2 Analysis of Symbiotic Profiles in MA

Given the restricted theoretical knowledge available in this area, an attempt to provide some insights into the success of MAs by analyzing the *evolvability* of two symbiotic search profiles is presented in what follows. In the experimental study, a MA that uses a search profile of Gaussian mutation operator as the stochastic variation $V(\mathbf{x})$, and the Davidon-Fletcher-Powell (DFP) or Davies, Swann, & Campey with Gram-Schmidt orthogonalization (DSCG) for individual learning $L(\mathbf{x})$, and are notated hereafter as MA-DFP and MA-DSCG, respectively. DSCG represents a form of direct search method that has been demonstrated to have good performance over some derivative-based numerical methods on the set of continuous benchmark problems in [136, 130]. DFP, on the other hand, is a popular quasi-Newton based individual learning method, which has also been used in Section 3.2 to obtain the sets of local optimums. Here the stopping criteria of the individual learning strategies is defined by the Cauchy's convergence test[3], thus allowing the learning strategies to arrive at local optima.

Using the five sets of local optimum solutions for the benchmark problems [4] (obtained previously in Chapter 3), the *evolvability* measure $Ev_{ID}(\mathbf{x}, t)$ of each search profile for each local optimum $\mathbf{x}$ are subsequently estimated to provide insights into the search mechanisms of the Gaussian mutation and individual learning operators, i.e., DFP or DSCG in the MAs. For each search profile, the fitness improvement $FI(\mathbf{x}, t)$ in Eq. 4.4 is estimated from a simulation of $T = 10 \times n$ iterations (n denotes the number of dimensions) on each local optimum $\mathbf{x}$: $FI(\mathbf{x}, t) \approx \sum_{i=1}^{T} (f(\mathbf{x}) - f(\mathbf{z}_i))/T$. In each iteration i, $\mathbf{y}_i = V(\mathbf{x})$ is an offspring of $\mathbf{x}$ reproduced using a mutation with normal distribution $\mathbf{N}(\mathbf{0}, \mathbf{1})$ and $\mathbf{z}_i = L(\mathbf{y}_i)$ denotes the resulting individual or local optimum obtained after learning. Using a similar procedure, the computational cost expectation of the search operators in Eq. 4.5 can also be estimated as $C(\mathbf{x}, t) \approx \sum_{i=1}^{T} C_i/T$, where C_i denotes the total computational cost incurred by $L(\mathbf{y}_i)$ in

[3] $|\mathbf{x}_{n+1} - \mathbf{x}_n| \leq \epsilon$ for $N > N_0$ and precision ϵ set to 1E-4

[4] $\mathbf{\Psi}_{\mathbf{Ackley}}$, $\mathbf{\Psi}_{\mathbf{Rastrigin}}$, $\mathbf{\Psi}_{\mathbf{Griewank}}$, $\mathbf{\Psi}_{\mathbf{Rosenbrock}}$ and $\mathbf{\Psi}_{\mathbf{Weierstrass}}$

each iteration. Here, the computational cost is defined by the number of function evaluations incurred until local optimality convergence.

Subsequently, the *evolvability* measure of each search profile on local optimum $\mathbf{x}$ is calculated as $Ev_{ID}(\mathbf{x}, t) = FI(\mathbf{x}, t)/C(\mathbf{x}, t)$. Note that t can omitted in this case since the fitness improvement and cost in the search profiles are averaged across the stochasticity of $V(\mathbf{x})$ which is independent of time. The details of the simulation procedure used is outlined in Algorithm 4. Simulation results obtained on the five 10-dimensional benchmark problems are then summarized using scatter plot $\{f(\mathbf{x}), Ev_{ID}(\mathbf{x})\}$, labelled here as *Ev-plot*, in Figures 4.3-4.7. Note that in the figure, *x*-axis and *y*-axis represent the fitness $f(\mathbf{x})$ and the *evolvability* measure $Ev_{ID}(\mathbf{x})$ of the local optimum, respectively.

Algorithm 4 Simulation Procedure

for $\mathbf{x}$ in $\mathbf{\Psi}$ **do**
  $FI(\mathbf{x}) = 0$, $C(\mathbf{x}) = 0$
  for $i = 1$ to $T = 10 \times n$ **do**
    Produce offspring $\mathbf{y}_i$ according to $V(\mathbf{x})$
    Produce individual $\mathbf{z}_i$ by individual learning $L(\mathbf{y}_i)$
    Fitness improvement $\triangle f(\mathbf{x}, \mathbf{z}_i) = f(\mathbf{x}) - f(\mathbf{z}_i)$
    $FI(\mathbf{x}) = FI(\mathbf{x}) + \triangle f(\mathbf{x}, \mathbf{z}_i)$
    $C(\mathbf{x}) = C(\mathbf{x}) + C_i$
  end for
  $FI(\mathbf{x}) = FI(\mathbf{x})/T$
  $C(\mathbf{x}) = C(\mathbf{x})/T$
  $Ev_{ID}(\mathbf{x}) = FI(\mathbf{x})/C(\mathbf{x})$
end for
Provide *Ev-plot*

4.2.1 Ackley function

The *Ev-plots* for symbiotic profiles *(M, DFP)* and *(M, DSCG)* on the Ackley function are illustrated in Figure 4.3. On the lower panel of Figure 4.3, *(M, DSCG)* is observed to exhibit positive *evolvability* $Ev_{ID}(\mathbf{x})$ on each local optimum $\mathbf{x}$. In contrast, the upper panel of Figure 4.3 indicates that *(M, DFP)* shows negative *evolvability* for local optimum with fitness value

under 8 (see *x*-axis). Further, the negative *evolvability* phenomenon is shown to prevail for most of the local optimums even at higher fitness ranges. The results thus highlighted a significantly higher *evolvability* $Ev_{ID}(\mathbf{x})$ on the local optimum resulting from search profile *(M, DSCG)* than *(M, DFP)*. Based on the notions of *constructive* and *obstructive* connectivity described in Section 4.1, the property of *constructive* connectivity, i.e., local optimums of Ackley function are likely to connect to other optimums of higher fitness, is shown for *(M, DSCG)*. In contrast, *obstructive* connectivity is observed on profile *(M, DFP)*.

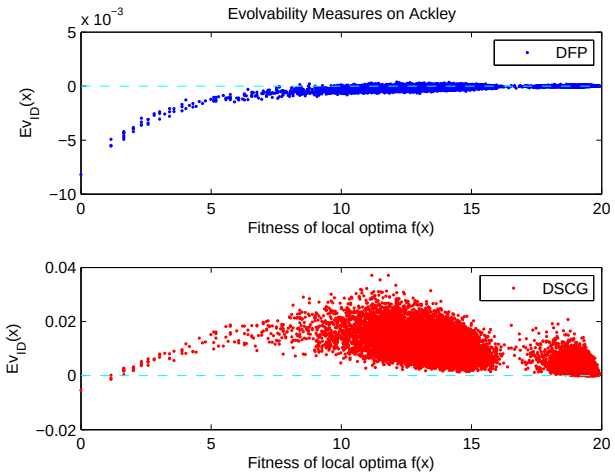


Figure 4.3: Evolvability Analysis on Ackley(10D) function.

4.2.2 Rastrigin function

As shown in Figure 4.4, both *(M, DFP)* and *(M, DSCG)* bring about positive *evolvability* measures on majority of the local optimum, but negative *evolvability* on some high quality local optimums of the Rastrigin function. Hence a *obstructive* connectivity can be inferred for local optimums that are closer to the global optimal. Comparing *(M, DFP)* with *(M, DSCG)*, however, *(M, DSCG)* exhibits significantly higher *evolvability* on most of the local optimums than *(M, DFP)*. This implies that *(M, DSCG)* displays a stronger *constructive* connectivity profile than *(M, DFP)* on Rastrigin.

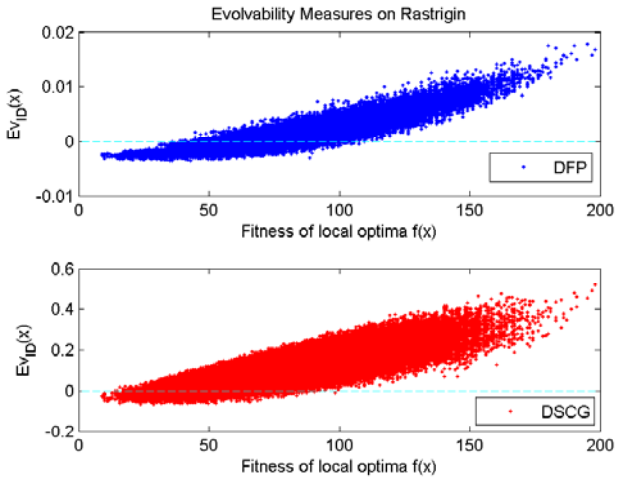


Figure 4.4: Evolvability Analysis on Rastrigin(10D) function

4.2.3 Griewank function

Next, the simulation results on the Griewank function is analyzed and discussed. Visually although the local optimum landscape of the Griewank function appears similar to that of the Rastrigin function, as presented in Chapter 3, it is worth noting that in contrast to the latter, both *(M, DFP)* and *(M, DSCG)* exhibit positive *evolvability* on nearly all the local optimums, as observed in Figure 4.5. Further, *(M, DSCG)* shows a higher *evolvability* measure than *(M, DFP)* for the same local optimums. Particularly, for local optimums with fitness value of 100, *(M, DSCG)* contributed *evolvability* measures that are in the range of 0.1 to 0.2, while *(M, DFP)* showed *evolvability* measures lower than $2E-3$.

4.2.4 Rosenbrock function

With respect to the Rosenbrock function, it is worth noting that the sparseness of the scatter plot in Figure 4.6 indicates a small number of local optimums existing in the fitness landscape. The obtained *evolvability* measure as shown in Figure 4.6 clearly depicts the strong *constructive* connectivity of the local optimums to global optimum.

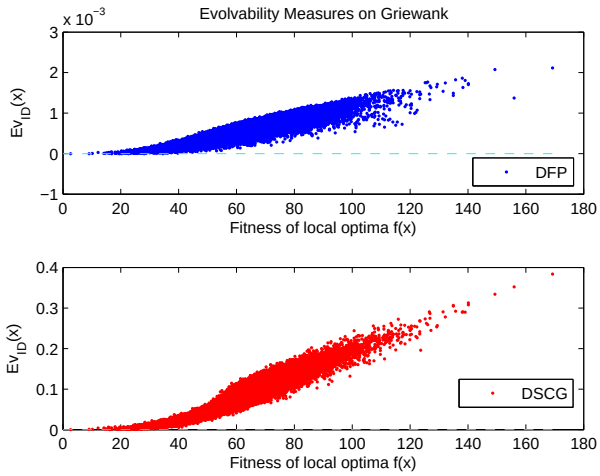


Figure 4.5: Evolvability Analysis on Griewank(10D) function

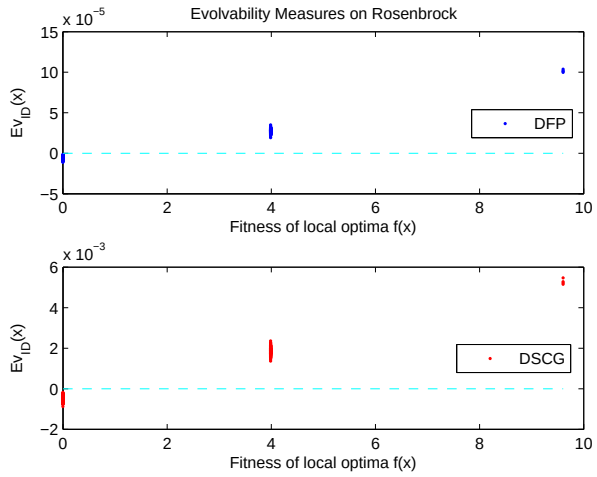


Figure 4.6: Evolvability Analysis on Rosenbrock(10D) function

4.2.5 Weierstrass function

(M, DSCG) shows a positive *evolvability* on nearly all local optimums of the Weierstrass function. A linearly increasing *evolvability* measure is observed in the lower panel of Figure 4.7 for *(M, DSCG)*. *(M, DFP)*, on the other hand, exhibits an entirely opposite behavior, with most local optimums giving negative *evolvability*, as shown in Figure 4.7, i.e., for those local optimums with fitness value under 15.

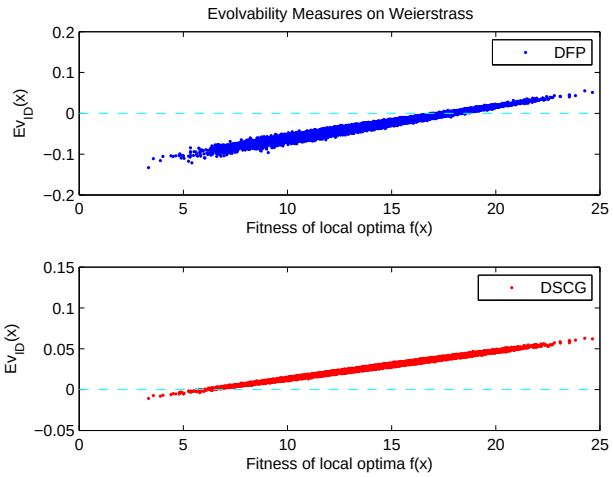


Figure 4.7: Evolvability Analysis on Weierstrass(10D) function

As summarized in Table 4.1, the results of analyzing the *evolvability* measures indicate that the search profile *(M, DSCG)* possesses a *constructive* local optimum connectivity on most of the representative benchmark functions, while *(M, DFP)* exhibits an *obstructive* connectivity of local optimums when approaching the global optimum of the Ackley, Rastrigin and Weierstrass functions. Further, it is worth noting that local optimum connectivity of a higher *evolvability* measure is observed in *(M, DSCG)* as compared to *(M, DFP)* on four out of the five problems considered, thus suggesting explanation into the wide success of hybrid MA-DSCG reported in the literature [136, 130].

Benchmark Problem	Local Optimum Connectivity	
	MA-DSCG	**MA-DFP**
Ackley	Constructive	Obstructive
Rastrigin	Constructive	Constructive
Griewank	Constructive	Constructive
Rosenbrock	Constructive	Constructive
Weierstrass	Constructive	Obstructive

Table 4.1: Local Optimum Connectivity Profiles of Benchmark Problems

4.3 Conclusions

In this chapter, the proposed notions of *symbiotic search profile* is first introduced. The symbiosis of stochastic variation and individual learning operators in MA is then formalized in the form of local optimum connectivity and quantified via the notion of *evolvability*. The concepts of '*constructive*'/ '*obstructive*' connectivity is also introduced as an important property for revealing the working mechanisms of MA in search. Further analysis on the symbiotic profile of *(M, DSCG)* and *(M, DFP)* based on their *evolvability* measure highlighted the unique local optimum connectivity properties and their influences on MA search performance, thus bringing about new understandings to the superior performance and success of MA-DSCG reported in previous studies.

Chapter 5

Symbiotic Evolution

Several adaptive memetic algorithms have been proposed in recent years, with many shown to solve a great variety of optimization problems more effectively and robustly than their canonical counterparts. Existing schemes for performing the adaptations however, are mostly designed based on semi-ad-hoc or heuristic/ meta-heuristic methods that comes with limited theoretical rigor [132]. More importantly, few has explicitly modelled the dynamics of stochastic variations and individual learning, or the symbiotic search profile, for adaptation in search. In Chapter 4, the notion of *evolvability* as the basis for assessment of symbiotic search profile has been introduced to quantify the degree of synergy between the stochastic variation and individual learning operators. Through the study, the *evolvability* of symbiotic search profiles was demonstrated to have great influence on search performance of MA. Taking this cue, Chapter 5 proceeds to present Symbiotic Evolution (SE) as the proposed memetic framework that facilitates the emergence and self-configuration of productive search profiles, transpiring from the symbiosis of stochastic variation and individual learning, working in sync on solving the given problem in hand. Last but not least, it is worth noting that the core interest of taking such a pursue of research is nonetheless motivated by the potential benefits brought about from the constructive synergy of stochastic variation and individual learning in accelerating search performances.

The chapter is organized as follows: Section 5.1 introduces Symbiotic Evolution (SE) as

a realization of the presented study on symbiosis in the context of self-configurable memetic algorithm, for solving complex problems. Taking a data-centric paradigm in the spirit of Optinformatics [96], SE performs the statistical learning of *evolvability*, at runtime, to infer the most productive symbiotic search profile for use on a given solution individual, leading to a self-configuring solver that adapts to search on the given problem in hand. Section 5.2 presents a numerical study on the search performance of SE while Section 5.3 analyzes the performances of SE with assessment made against several recent state-of-the-art modern evolutionary methods, adaptive and hybrid approaches. Finally, Section 5.4 concludes the present study with a brief discussion.

5.1 Self-configurable Memetic Search: Symbiotic Evolution

In the previous chapter, memetic solver's behavior that emerged from the symbiotic search profile has been formalized in the form of local optimum connectivity based on the notion of *evolvability*. Figure 5.1 presents an illustrative example of self-configurable memetic search where the connectivity of local optimums exhibited by three memetic algorithms are depicted on the contour plots of a problem landscape. The symbiotic search profiles, as defined by the interactions of unique stochastic variation and individual learning mechanisms, are then represented by broken and/or continuous directed edges[1] in Figure 5.1(a), (b), and (c), respectively. In the figures, search profile $\mathbb{Y}$ (represented by continuous line) is depicted as more effective for finding good quality solutions that are close to the global optimum faster than profile $\mathbb{X}$ (represented by broken line), while the latter is shown as more effective in converging to the precise global optimum, although more hops (denoting the computational effort incurred) are required.

In particular, the search behaviors of the stochastic variation and individual learning are depicted as directed graphs that model the connectivity probability of local optimum solutions

[1]Note that only the edge with highest probability of reaching a minimum among all others is depicted.

on the contour plot. (a) Connectivity Search Profile of $\mathbb{X}$ (b) Connectivity Search Profile of $\mathbb{Y}$ (c) Connectivity Search Profile of $\mathbb{X} + \mathbb{Y}$. From the graphs, the search path from solution A to the global optimum G is likely to be $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G$ when profile $\mathbb{X}$ ("dotted line") is used. Note that the "circle" and "star" symbols in Figure 5.1 denotes the local optimums and global optimum, respectively. Thus profiles $\mathbb{X}$ and $\mathbb{Y}$ display strength in solution precision and efficiency, respectively. However, when both profiles work together effectively (profile $\mathbb{X} + \mathbb{Y}$), the path becomes shorter as defined by $A \rightarrow C \rightarrow E \rightarrow F \rightarrow G$. The emergent behavior resulting from the interactions by unique stochastic variations and individual learning mechanisms working in symbiosis, i.e., search profiles $\mathbb{X}$ + $\mathbb{Y}$, as depicted in Figure 5.1(c), if appropriately harvested would lead to a higher probability of discovering the global optimum at high precision and efficiency in search, thus forms the core motivation of the present study on self-configurable memetic algorithm.

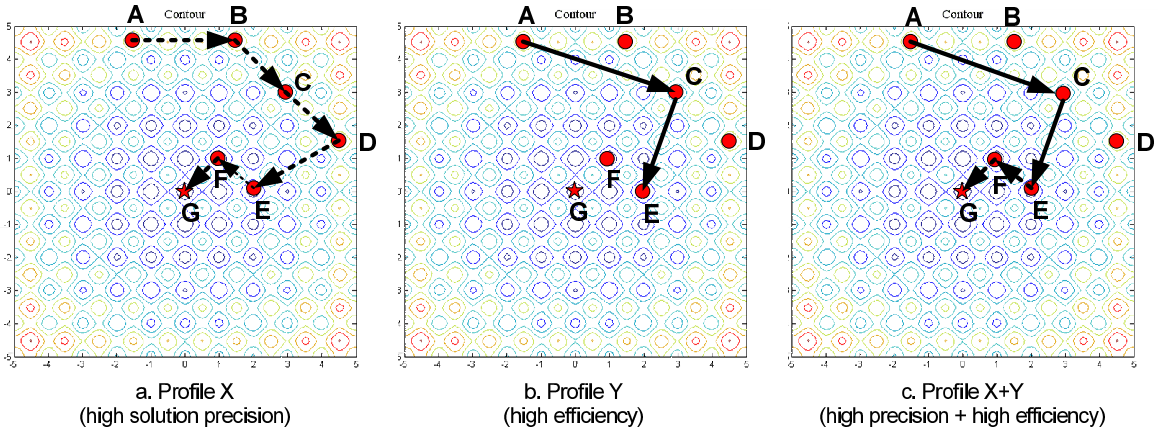


Figure 5.1: Symbiosis of stochastic variation and individual learning.

5.1.1 Statistical Learning of Evolvability

The *evolvability* measure of a symbiotic search profile $Ev_{ID}(\mathbf{x}, t)$ is introduced to indicate the expectation of how a developed offspring $\mathbf{z}$, after undergoing individual learning, is improved over its initial parent $\mathbf{x}$ with the computational cost taken into consideration. Given a multitude of search profiles composing of stochastic variation and individual learning working on

a minimization problem, the most productive profile for a given solution $\mathbf{x}$ is defined here as having the highest *evolvability* measure, i.e., $\arg\max Ev_{ID}(\mathbf{x}, t)$. Since realistic problems are seldom tractable and possess fitness landscapes that are highly complex in practice, a theoretic formulation as introduced in Chapter 4 may be deemed inappropriate. To deal with complex search problems, the concept of *Evolvability Learning* in Symbiotic Evolution is introduced in the present section. Taking a data-centric paradigm [96], here a statistical approach on the *evolvability learning* of symbiotic search profiles during runtime is proposed. In consistence with the definition of *evolvability* in Eq. 4.6

$$Ev_{ID}(\mathbf{x}, t) = \frac{\int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) \times Gain(\mathbf{y}, \mathbf{x}) d\mathbf{y}}{\int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) \times Cost(\mathbf{y}, \mathbf{x}) d\mathbf{y}}$$

the statistical learning on the *evolvability* $Ev_{ID}(\mathbf{x}, t)$ of symbiotic search profile $ID = (V, L)$ on solution individual $\mathbf{x}$ at generation t is quantitatively estimated from distinct data pairs $\Phi_L = \{(\mathbf{y}_1, \mathbf{z}_1), \ldots, (\mathbf{y}_m, \mathbf{z}_m), \ldots, (\mathbf{y}_K, \mathbf{z}_K)\}$ using Algorithm 5. Note that Φ_L denotes the database containing the vectors of solution individuals and their associated fitness uncovered along the evolution and $\mathbf{y}_i \neq \mathbf{y}_j$ if and only if $i \neq j$.

Using weighted sampling approach, based on the density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ of stochastic variation operator, each distinct sample pair of solution individuals $\{(\mathbf{y}_i, \mathbf{z}_i)\}$ is then associated with some weight $w_i(\mathbf{x})$, which defines the probability of it being selected in the estimation of Ev_{ID}. The weight $w_i(\mathbf{x})$ essentially reflects the current probability of jumping from solution $\mathbf{x}$, via stochastic variation $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$, to offspring $\mathbf{y}_i$ which was archived in the database. In other words, the weight $w_i(\mathbf{x})$ measures the relevancy of $\{\mathbf{y}_i, \mathbf{z}_i\}$ in *evolvability* learning process on solution $\mathbf{x}$. Considering $\{(\mathbf{y}_i, \mathbf{z}_i)\}_{i=1}^K$ as distinct samples from current distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$, the weights $w_i(\mathbf{x})$ associated with samples $(\mathbf{y}_i, \mathbf{z}_i)$ satisfy the equations: $\sum_{i=1}^K w_i(\mathbf{x}) = 1$ and $w_i(\mathbf{x})$ is proportional to $\int_{\mathbf{v}(\mathbf{y}_i)} P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x}) d\mathbf{y}$, in which $\mathbf{v}(\mathbf{y}_i)$ denotes the arbitrarily small bin $\mathbf{v}$ around solution $\mathbf{y}_i$ and the integral is taken over interval $[\mathbf{y}_i^{(k)} - v, \mathbf{y}_i^{(k)} + v]$ on each dimension k. Since the integration $\int_{\mathbf{v}(\mathbf{y}_i)} P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) d\mathbf{y}$ is computationally expensive,

without significant loss of precision, the weight w_i is estimated by Eq. 5.1.

$$w_i(\mathbf{x}) = \frac{P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})}{\sum_{j=1}^{K} P(\mathbf{y}_j|\mathbf{P}^t, \mathbf{x})} \quad \text{(Eq. 5.1)}$$

It is worth noting that the condition $\sum_{i=1}^{K} w_i(\mathbf{x}) < \epsilon$ (Line 4 of Algorithm 5) caters for the scenario when all samples $(\mathbf{y}_i, \mathbf{z}_i)$ are irrelevant for evolvability estimation on solution $\mathbf{x}$, i.e., $P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$ is too small. The role of ϵ thus specifies the threshold level of irrelevance for archived samples in the evolvability learning. In particular, ϵ is configured to a precision of $1E-9$ in the present study. Note that the conditional density function $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ in Eq. 5.1 is derived based on the characteristic of variation operators used, in order to reflect the current state of the search. Details of this step (Line 2 of Algorithm 5) in the current implementation will be provided in Section 5.2.1. From the K archived sample pairs[2], $FI(\mathbf{x}, t)$ and $C(\mathbf{x}, t)$ are then estimated using a weighted sampling approach defined by w_i as follows:

$$FI(\mathbf{x}, t) = \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) \times Gain(\mathbf{y}, \mathbf{x}) d\mathbf{y} \approx \sum_{i=1}^{K} w_i \times Gain(\mathbf{y}_i, \mathbf{x}) \quad \text{(Eq. 5.2)}$$

$$C(\mathbf{x}, t) = \int_{\mathbf{y}} P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) \times Cost(\mathbf{y}, \mathbf{x}) d\mathbf{y} \approx \sum_{i=1}^{K} w_i \times Cost(\mathbf{y}_i, \mathbf{x}) \quad \text{(Eq. 5.3)}$$

Using Eq. 4.6, Eq. 5.2 and Eq. 5.3, the *evolvability* of symbiotic search profile ID is estimated by Eq. 6.3

$$Ev_{ID}(\mathbf{x}, t) = \frac{\sum_{i=1}^{K} w_i \times Gain(\mathbf{y}_i, \mathbf{x})}{\sum_{i=1}^{K} w_i \times Cost(\mathbf{y}_i, \mathbf{x})} \quad \text{(Eq. 5.4)}$$

5.1.2 Symbiotic Evolution for Non-linear Programming

Next, the proposed self-configurable memetic framework, labelled here as Symbiotic Evolution (SE), for solving non-linear programming problems is presented. The essential ingredients

[2]In the implementation, the database $\Phi_L = \{(\mathbf{y}_j \rightarrow \mathbf{z}_j, Cost)\}$ is created and maintained for each individual learning strategy *L*. Each time the individual learning *L* is performed, the new entry will be inserted to Φ_L in the first-in-first-out (FIFO) manner, i.e., the outdated entries in the database will be removed when the number of entries of Φ_L exceeds the maximum size.

Algorithm 5 Statistical Evolvability Learning *SEvL(ID)*

Query archived data $\Phi_L = \{(\mathbf{y}_j \rightarrow \mathbf{z}_j, Cost)\}$ of search profile $ID = (V, L)$
Model density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ of variation operator $V(.)$, based on $\mathbf{P}^t$ and $\mathbf{x}$
Calculate weight $w_i(\mathbf{x}) = P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$ for each sample $\mathbf{y}_i$
if $\sum_{i=1}^{K} w_i(\mathbf{x}) < \epsilon$ **then**
    $w_i(\mathbf{x}) = 0$ {No relevant data is available}
    return $Ev_{ID}(\mathbf{x}, t) = -\infty$
else
    $w_i(\mathbf{x}) = w_i(\mathbf{x}) / \sum_{j=1}^{K} w_j(\mathbf{x})$ {Normalize w_i}
    return $Ev_{ID}(\mathbf{x}, t) = \sum_{i=1}^{K} w_i(\mathbf{x}) \times Gain(\mathbf{y}_i, \mathbf{x}) / \sum_{i=1}^{K} w_i(\mathbf{x}) \times Cost(\mathbf{y}_i, \mathbf{x})$ {Eq. 6.3}
end if

of the framework is composed of multiple stochastic variation *V* and individual learning *L* procedures that are governed by the *evolvability* of symbiotic search profiles. Let Ψ denote the set of symbiotic search profiles *(Stochastic Variation, Individual Learning)* considered in the SE. Based on the statistical evolvability learning scheme described in Section 5.1.1, SE adapts the search by inferring the most productive symbiotic search profile (i.e., that would lead to good search performances) to use on each individual while the search progresses online, as outlined in Algorithm 6. Without loss of generality, in the event of a minimization problem, the most productive symbiotic search profile for individual $\mathbf{x}$ is deemed as one that has the largest estimated *evolvability* measure, i.e., $\arg\max Ev_{ID}(\mathbf{x}, t)$.

It is worth noting on the generality of the proposed framework in the use of density function $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ to represent and reflect the unique characteristics of the stochastic search operators, thus not restricting to any particular type of operator. By formulating the search operator with a density function, the framework allows the incorporation of different stochastic variation operators, which depends on suitability to the given problem of interest.

In the first step, a population of individuals is initialized either randomly or using design of experiment techniques such as Latin hypercube sampling. The evaluated population of individuals then undergoes natural selection, for instance, via fitness-proportional or tournament selection. The *evolvability* of symbiotic search profiles with respect to each individual in the

population is then estimated probabilistically using the database Φ_L of solution individuals uncovered and archived along the search. Subsequently, each individual is evolved using their respective statistically inferred productive symbiotic search profile for generating the next population of solution individuals. The entire process repeats until the specified stopping criteria is satisfied.

At the initial generation, note that no data in Φ_L are available for the learning of evolvability. This case is also considered in Algorithm 5 as "no relevant data available" and the evolvability of search profiles are set to $-\infty$ (line 5, Algorithm 5). In this case, each search profile is randomly selected for use on each individual with the equal probability of $1/|\Psi|$(line 11, Algorithm 6). In the subsequent generations, the evolvability learning will be able to proceed on the database Φ_L of each symbiotic search profile which has been populated with sample data $(\mathbf{y} \rightarrow \mathbf{z}, Cost)$. For the sake of brevity, a summary of the Symbiotic Evolution is depicted in Figure 5.2.

Algorithm 6 Symbiotic Evolution

1: *Generate* an initial population
2: **while** Stopping conditions are not satisfied **do**
3: *Evaluate* all individuals in the population
4: *Update* reproduction pool size
5: *Select individuals* for the reproduction pool Ω
6: **for** each individual $\mathbf{x}$ in Ω **do**
7: **for** each $ID = (V, L) \in \Psi$ **do**
8: $Ev_{ID}(\mathbf{x}, t)$ = ***SEvL(ID)*** on $\mathbf{x}$ (see Algorithm 5)
9: **end for**
10: **if** $Ev_{ID}(\mathbf{x}, t) < 0 \; \forall \; ID$ **then**
11: Select symbiotic profile ID randomly
12: **else**
13: Select symbiotic profile ID with optimal (largest) $Ev_{ID}(\mathbf{x}, t)$
14: **end if**
15: *Evolve* $\mathbf{x}$ according to stochastic variation operator $V(.)$
16: *Perform* individual learning via operator $L(.)$
17: *Proceed* in the spirit of Lamarckian learning
18: **end for**
19: **end while**

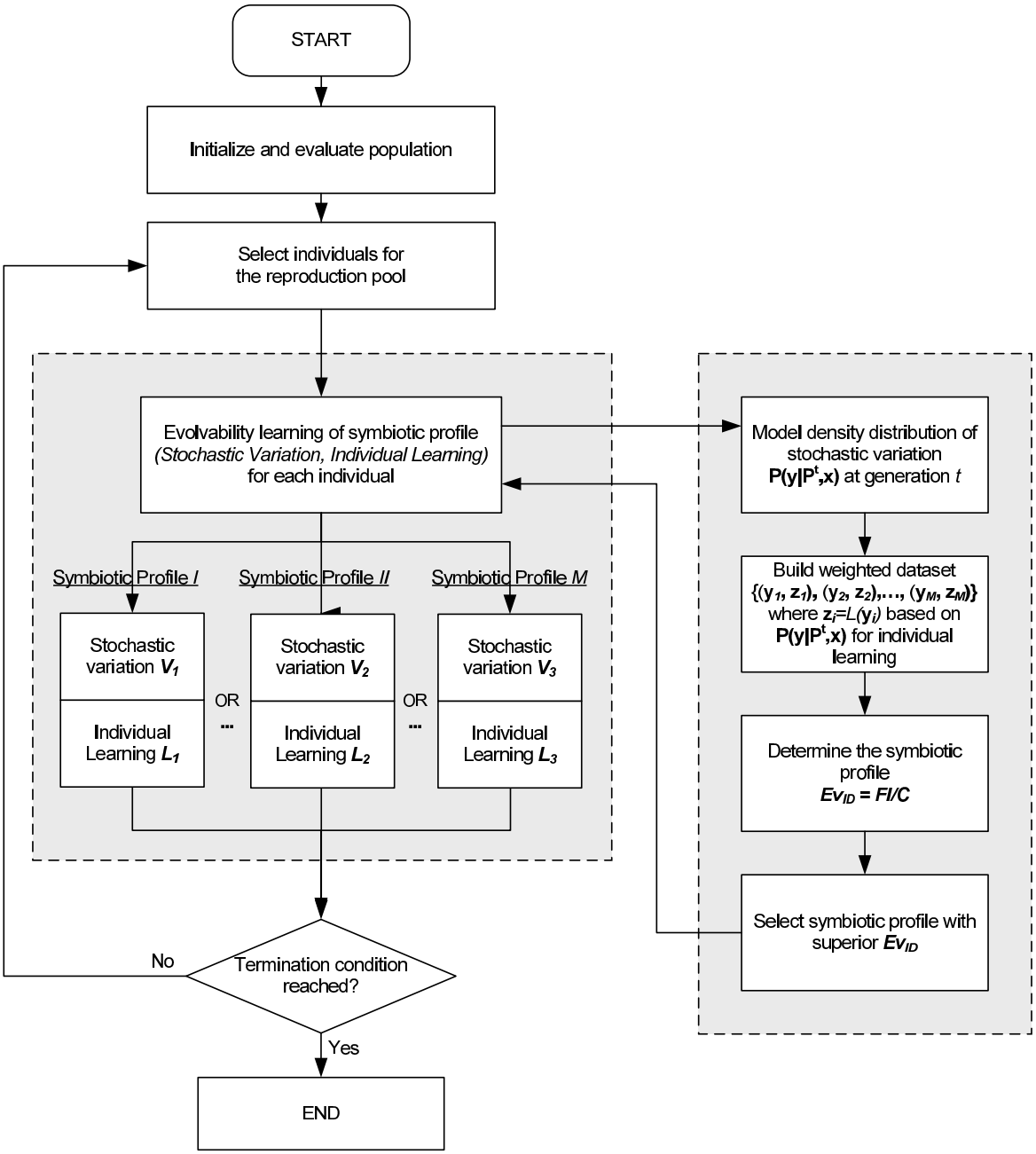


Figure 5.2: Symbiotic Evolution (SE) Framework

5.2 Empirical Study

In this section, a numerical study of the proposed SE on several commonly used continuous parametric benchmark functions is presented. To ensure a comprehensive study, diverse pairs of unique stochastic variations and individual learning procedures are considered.

5.2.1 Stochastic Variations: Mutation & Crossover

The density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ of some common stochastic variation in EA, also sometimes referred to as genetic variations, are derived and described in what follows. In the present experimental study, the stochastic variation operators considered include the conventional Gaussian mutation and the uniform crossover, which have been used widely in real-coded genetic evolution. The motivation to use simple variation operators is to provide a simple illustration that would give an ease of understanding on the generality of Symbiotic Evolution. It is worth noting that other advanced real-parameter search operators, such as that discussed in [59], can also be considered within the framework through the density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$.

In real-coded Gaussian mutation, for instance, each individual vector is perturbed with a multivariate Gaussian distribution $\mathbf{r} \sim \mathbf{N}(\mathbf{0}, \mathbf{C})$, i.e., $\mathbf{x}' = \mathbf{x} + \mathbf{r}$. In this case, the density function of mutation operator is given in [39] as

$$P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \times \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{x})^T\mathbf{C}^{-1}(\mathbf{y}-\mathbf{x})\right) \qquad \text{(Eq. 5.5)}$$

The uniform crossover procedure for a single parent $\mathbf{x}$ consists of the following steps: a) $\mathbf{x}$ selects randomly its mate $\mathbf{x}'$ in the current reproduction pool, b) uniformly crossover $\mathbf{x}$ and $\mathbf{x}'$ to create two offspring: each location i of the offspring, i.e., $\mathbf{y}_1^{(i)}$ or $\mathbf{y}_2^{(i)}$, has a value of either $\mathbf{x}_1^{(i)}$ or $\mathbf{x}_2^{(i)}$ at the crossover probability p_{cross}, and c) select randomly one of the offspring as the offspring of $\mathbf{x}$. Although there exists many variations of crossover operators in real-coded representation, such as the uniform and arithmetic crossovers, it is worth highlighting all these

variations share common property of the resultant offspring $\mathbf{y}$ bounded by $\min_{j=1...N}\{\mathbf{x}_j^{(i)}\}$ and $\max_{j=1...N}\{\mathbf{x}_j^{(i)}\}$ for each dimension[3], i.e., $\forall i = 1 \ldots n$.

Hence the density distribution of crossover operator can be modelled as a uniform distribution of bounds

$$\mathbf{R} = [\min_{j=1...N}\{\mathbf{x}_j^{(i)}\}, \max_{j=1...N}\{\mathbf{x}_j^{(i)}\}]_{i=1...n}$$

defined in Eq. 6.4, where $Vol(\mathbf{R})$ denotes the hyper-volume of hyper-rectangle $\mathbf{R}$. Note that since the hyper-rectangle $\mathbf{R}$ reduces as the search progresses, the probabilistic model of crossover operator reflects well on the refinement of the search space by crossover and selection pressure. In particular, Eq. 6.4 models the density distribution of the coordinated-based crossover considered in the present study.

$$P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) = UniformDist(\mathbf{R}) = \begin{cases} \frac{1}{Vol(\mathbf{R})} & \textit{if } \mathbf{y} \in \mathbf{R} \\ 0 & \textit{otherwise} \end{cases} \qquad \text{(Eq. 5.6)}$$

5.2.2 Individual Learning

In the present study, a comprehensive set of representative individual learning schemes is considered. These include the zero-order derivative *Davies, Swann, and Campey with Gram-Schmidt orthogonalization* (DSCG) [168], 1st-order derivative *Davidon, Fletcher and Powell* strategy (DFP) [153], and stochastic *Evolution Strategy* $(1 + n)$ (ES) (Appendix B.4).

To deal with "black-box" optimization problems that usually exist in practice, the investigation of symbiotic evolution (SE) does not assume the availability of the gradient information in analytical form. In the study, the gradient information used by DFP strategy is estimated using numerical differentiation in numerical analysis [155]. To facilitate a fair comparison to other

[3] If $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{y}$ denote the parents and the offspring then each locus of the offspring $\mathbf{y}$ satisfies the inequality

$$\min\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\} \leq \mathbf{y}^{(i)} \leq \max\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\}, \forall i = 1 \ldots n$$

evolutionary/memetic algorithms that use other forms of information in the individual learning, the exact function evaluations incurred by the estimation of gradient information were also included in the computational budget of the proposed SE[4].

5.2.3 Numerical Results of Symbiotic Evolution on Representative Complex Benchmark Problems

To instill further understanding on symbiotic evolution, various aspects of the proposed framework are investigated on a comprehensive collection of unimodal/multimodal, epistatic/non-epistatic, continuous/discontinuous and noisy benchmark problems used in literature [150]. The reader is referred to Appendix A where the benchmark test functions are tabulated with their notable characteristics. In particular, here the analysis of numerical results focuses on the potential of symbiotic evolution in providing high quality solution and efficiency across different problem types and the amount of additional computational effort incurred over and above the canonical memetic algorithms.

5.2.3.1 Search Quality & Efficiency

The parametric configurations of the experiments are summarized in Table 5.1. From the configurations of stochastic variations (i.e., Gaussian mutation or coordinated-based crossover) and individual learning procedures (i.e., DSCG, DFP or ES) considered, note that six potential combinations or unique configurations of symbiotic search profiles can transpire in the Symbiotic Evolution.

The average convergence search trends across 50 independent runs, attained by means of symbiotic evolution and canonical evolution represented by the six MAs, when solving the representative 30-dimension shifted rotated Ackley, Rastrigin, Griewank, shifted Rosenbrock and noisy shifted Schwefel functions are depicted in Figures 5.3.a-5.7.a, respectively. Here,

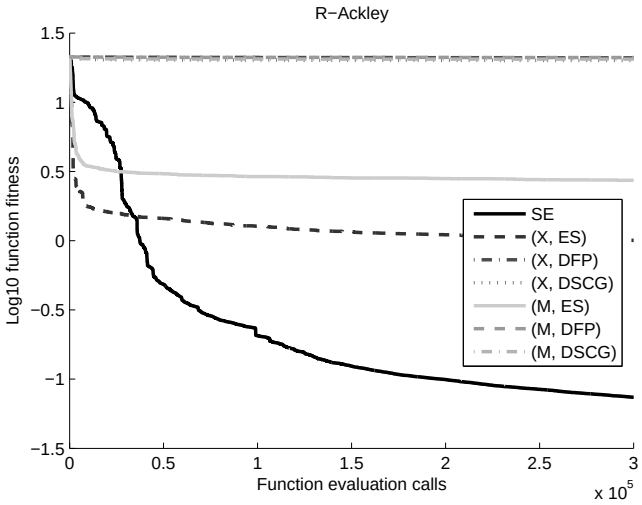
[4]For instance, the estimation of the 1st-order gradient of an individual solution $\mathbf{x}$, given $f(\mathbf{x})$, for a 30-dimension problem requires the cost of 30 additional function evaluations.

Table 5.1: Algorithm parameters setting

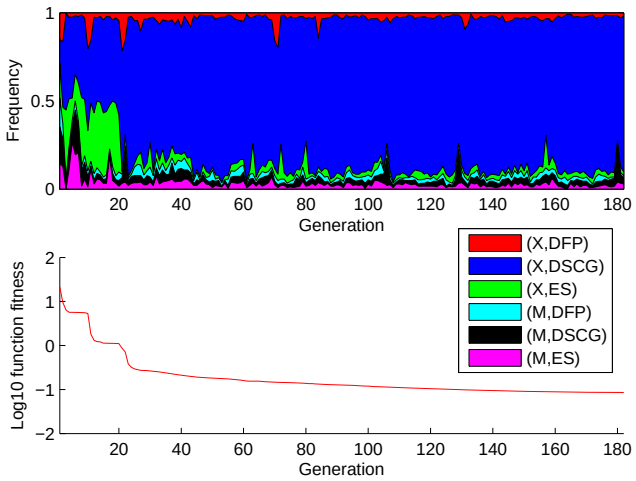
General parameters	
Encoding scheme	Real-coded
Population size	50
Selection scheme	Roulette wheel
Merging scheme	Elitism
Stopping criteria	300000 evaluations
Individual learning strategies	DSCG, DFP and ES
Initial individual learning intensity	300 evaluations
Variation operator	Uniform crossover and Gaussian mutation $N(0,1)$

each run continues until the global optimum was found or a maximum of $300,000$ function evaluations were reached. In addition, the average frequency of each symbiotic profile that is inferred as most productive on the solution individuals along the search are depicted in Figures 5.3.b-5.7.b for the respective problems considered.

Statistical results in Figures 5.3.a-5.7.a indicated that by adaptively configuring the symbiotic search profile along the search, Symbiotic Evolution exhibits search performance that outwits all the canonical MAs. It is worth noting that the search traces in Figures 5.3.a-5.7.a revealed interesting trends of slower convergence rate exhibited by symbiotic evolution during the initial stage of the search as compared to the other canonical evolutions on all the test problems. This denotes the initial learning phase of the SE in acquiring sufficient data and knowledge about the search profile before evolvability learning begins to bite. Note that when the database $\Phi_L = \{(\mathbf{y}_i, \mathbf{z}_i)\}$ containing data pairs that represents each search profile gets sufficiently populated, the statistical evolvability learning kicks in to select the most productive search profile for each unseen individual solution, leading to a faster convergence to the precise global optimum than other canonical MAs considered.

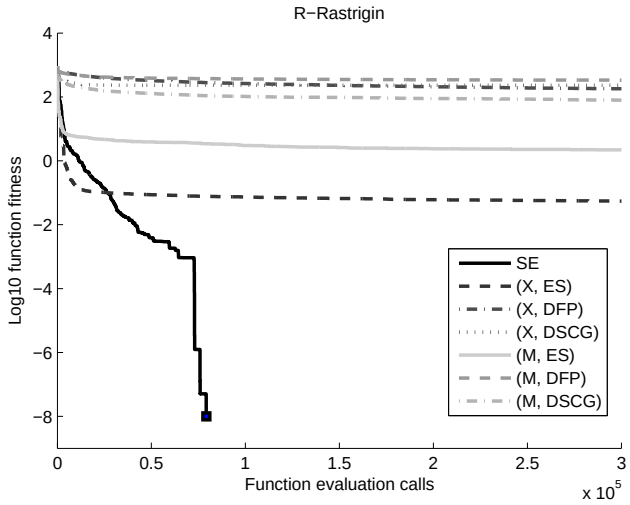


5.3.a: Convergence trend

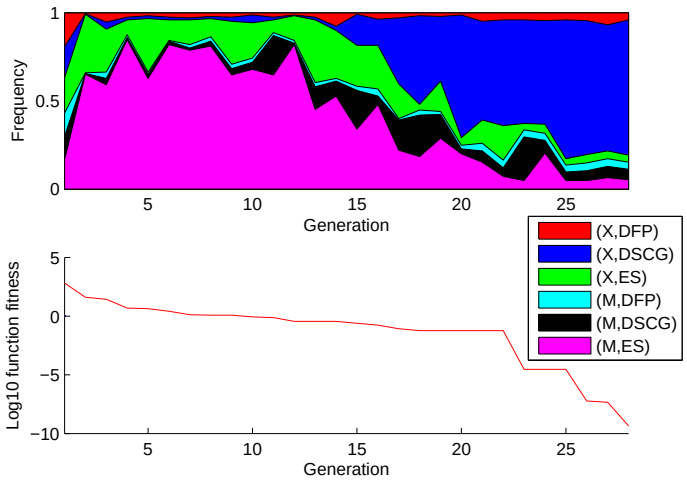


5.3.b: Profile frequency

Figure 5.3: Shifted rotated Ackley (30D) function

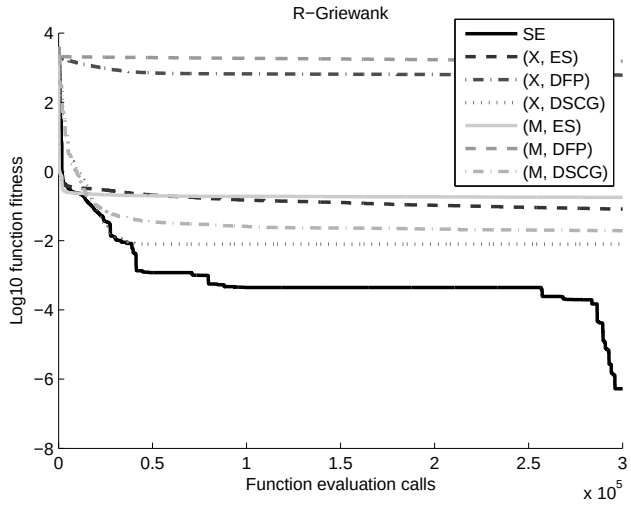


5.4.a: Convergence trend

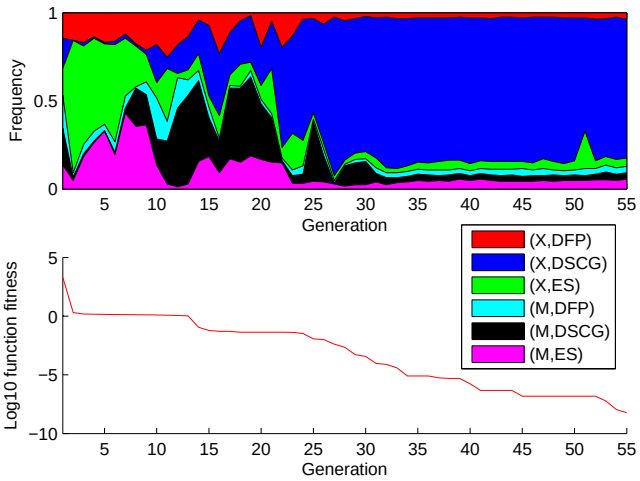


5.4.b: Profile frequency

Figure 5.4: Shifted rotated Rastrigin (30D) function

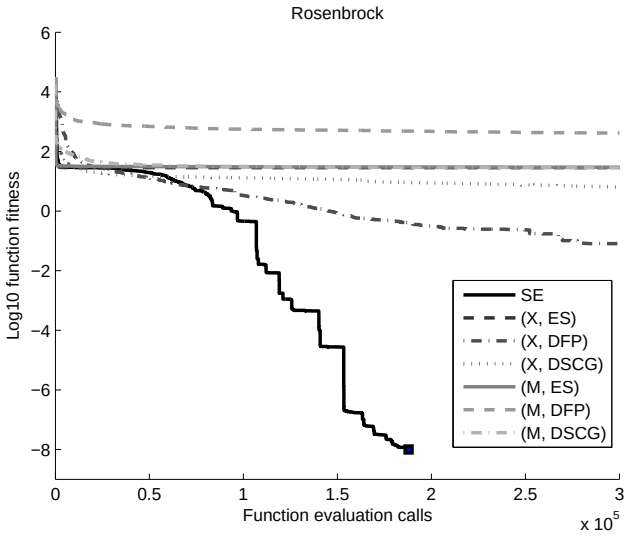


5.5.a: Convergence trend

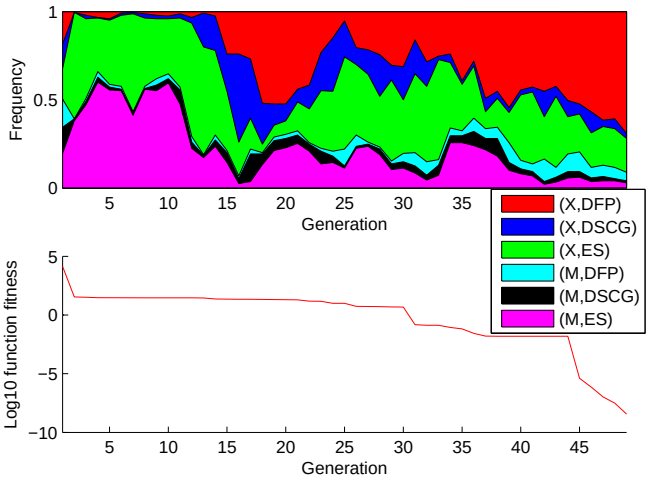


5.5.b: Profile frequency

Figure 5.5: Shifted rotated Griewank (30D) function

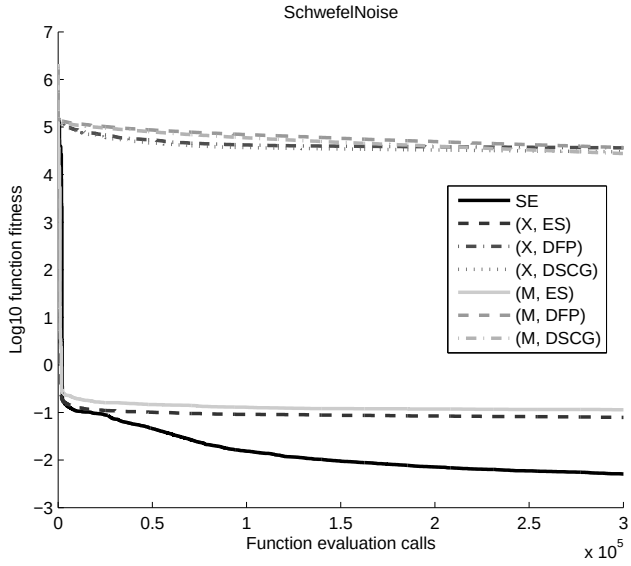


5.6.a: Convergence trend

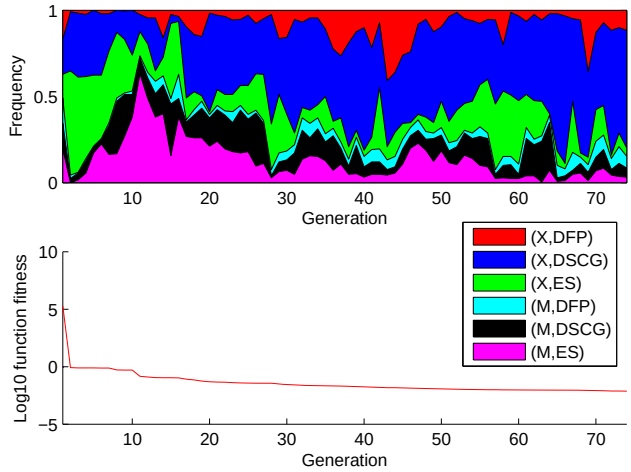


5.6.b: Profile frequency

Figure 5.6: Shifted Rosenbrock function



5.7.a: Convergence trend



5.7.b: Profile frequency

Figure 5.7: Shifted noisy Schwefel (30D) function

From Figure 5.3.b, search improvement in Symbiotic Evolution on the rotated Ackley function can be inferred as contributed mainly by profile *(Crossover, DSCG)* and *(Crossover, ES)*, while the contributions of *(Crossover, DSCG)* is more evident at later stages of the search. On rotated Rastrigin function, however, *(Mutation, ES)* is chosen for most individuals in the early generations of the SE search, before *(Crossover, DSCG)* begins to take effect as the search space is refined. For unimodal landscapes including Rosenbrock and noisy Schwefel functions, the plots in Figures 5.6.b and 5.7.b showed that decreasing frequencies of usage for symbiotic profiles with Gaussian mutation operator at the later stages of search. This appears to be due to the destructive effect of the Gaussian mutation operator when the search tries to converge to the global optimum with high precision. The observations of no fixed symbiotic configuration working as best throughout the search and across different optimization problems further confirm the motivation for introducing the concept of Symbiotic Evolution.

5.2.3.2 Computational Complexity

Algorithmically, Symbiotic Evolution differs from canonical evolution in that the former introduces a *Statistical Learning of Evolvability* phase that involves two core computations, namely

i) Modeling the density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ of stochastic variation for all individuals in the reproduction pool. The computational cost or time complexity of this estimation largely depends on the form of variation method considered. For instance, the costs of modeling the density function of the Gaussian mutation and crossover operators, described in Section 5.2.1, is of order $O\{n_{indivs}\}$ and $O\{1\}$, respectively, where n_{indivs} denotes the size of reproduction pool.

ii) Computing weights w_i of the archived solutions for all individuals in the reproduction pool. The computational cost or time complexity involved is of order $O\{n_{indivs} \times |\Psi| \times |\Phi_L|\}$, with $|\Phi_L|$ denoting the number of archived individuals.

Hence the additional time complexity for one generation of symbiotic evolution is of order $O\{n_{indivs} \times (|\Psi| \times |\Phi_L| + 1)\}$ in which the number of symbiotic search profiles $|\Psi|$ equals to $n_{IL} \times n_V$, where n_{IL} and n_V denote the number of individual learning and variation methods considered. As the size of reproduction pool n_{indivs} and the size of archived database $|\Phi_L|$ is typically fixed in advance, the time complexity of symbiotic evolution is linearly proportional to the number of search profiles involved. More importantly, since Symbiotic Evolution requires no additional function evaluation calls, the extra cost incurred may be considered to be negligible when dealing with computationally expensive problems [201].

5.3 Symbiotic Evolution: Assessment Against Other Evolutionary and Adaptive Search Approaches

In this section, a detail assessment of Symbiotic Evolution (as described in Section 5.2.3) against several recent state-of-the-art evolutionary and adaptive approaches on commonly used test suite (see Appendix A) is presented to verify the efficacy of the proposed approach.

5.3.1 Comparison to Adaptive Algorithms

To see how adapting the symbiotic search profiles affects the performance and efficiency of the search as compared to meme adaptation alone, in this subsection, the performance of Symbiotic Evolution is compared with those of other adaptive mechanisms on the choice of individual learning methods (memes) in MAs. Researchers use the terms meta-Lamarckian learning, hyperheuristic and multi-memes arbitrarily when referring to memes adaptation in adaptive MAs. In [138], adaptive mechanisms are classified into adaptation types (static, adaptive and self-adaptive rules using qualitative or quantitative feedback) and levels (external, local and global level) as summarized in Table 5.2. In addition, comparison to eight canonical MAs reported in the literature as listed in Table 5.3 are made in the present study.

Table 5.2: Classification of Adaptive MAs

Adaptation Type	
Static	No form of feedback (improvement attained by the chosen meme on the chromosome search) is used during the search
Adaptive	Qualitative or quantitative feedback influences the choice of memes at each decision point
Self-adaptive	The self-adaptation of memes is implemented in the idea of evolution
Adaptation Level	
External	No online knowledge about the memes is involved in the choice of memes
Local	Parts of the historical knowledge is involved in the choice of memes
Global	Complete historical knowledge is used to decide on the choice of memes

Table 5.3: List of Memes or Individual Learning Methods Considered

Abbreviations	**Algorithm Description**
BL	Bit climbing algorithm [34]
DP	Davis, Swan and Campey with Palmer orthogonalizational [168, 187]
FB	Schwefel library Fibonacci search [168]
FL	Fletcher's 1972 method by Siddall [44]
GL	Repeated one-dimensional Golden section search [168]
SX	Powell's strategy of conjugate directions [153]
PS	A direct search using the conjugate direction approach with quadratic convergence [66]
SK	A series of exploratory moves that consider the behavior of the objective function at a pattern of points, all of which lie on a rational lattice [127]

To be consistent, the results presented for Symbiotic Evolution are the average of 50 independent runs. For fair comparison, each run continues until the global optimum was found or a maximum of $40,000$ function evaluation calls was reached, which follows the experimental conditions considered in [138]. Using statistical T-test at 95% confidence level [162], the performance comparison of Symbiotic Evolution to the adaptive MAs on the benchmark test problems are tabulated in Table 5.4. For the detailed statistical results pertaining to each adaptive method, the reader is referred to Table 5.5. In the case where an algorithm manages to

locate the global optimum of a benchmark problem, the number of evaluation count presented indicates the effort taken to reach the optimum solution. Otherwise, the best fitness averaged over 50 independent runs is reported.

From Table 5.4, Symbiotic Evolution is shown to outperform all the canonical and adaptive MAs significantly, statistically at 95% confidence on all the problems. Note that this is a significant finding, which highlights the benefits of facilitating the self-configuring of emergence productive symbiosis search profiles in evolutionary computation instead of adapting only at the level of individual learning, as previously studied in the literature.

Table 5.4: Results of t-test at 95% confidence level for SE and other adaptive search algorithms in solving the benchmark problems (Note that s+ or s- indicates that SE is significantly statistically better or not, respectively).

Level-Type		$F_{Griewank}(10D)$	$F_{Rastrigin}(20D)$	$F_{Sphere}(30D)$	$F_{Step}(5D)$
External-Static	S-E	s+	s+	s+	s+
Local-Qualitative	QL1-L	s+	s+	s+	s+
	QL2-L	s+	s+	s+	s+
Global-Qualitative	QL3-G	s+	s+	s+	s+
Global-Quantitative	QN1-G	s+	s+	s+	s+
	QN2-G	s+	s+	s+	s+
Local-Quantitative	QN3-L	s+	s+	s+	s+
Local Self-adaptive	S-L	s+	s+	s+	s+
Canonical MAs	GA-$< . >$	s+	s+	s+	s+

5.3.2 Assessment Against Other Recent State-of-the-Art Evolutionary Approaches

In this section, the proposed Symbiotic Evolution is assessed against 13 recent state-of-the-art evolutionary approaches reported in the literature. Table 5.6 tabulates the list of diverse algorithms, consisting of Memetic Algorithms [132, 134, 122, 121], CMA Evolution Strategy [7, 5], continuous Estimation of Distribution Algorithms [223], Genetic Algorithms [46, 10], Differential Evolution [163], Particle Swarm Optimization [126], Cooperative Co-Evolution [152], and Population-based Algorithm Portfolio [150], considered here for comparisons with

Table 5.5: Success measure of the algorithms in solving the benchmark functions. Either average best fitness or average number of evaluation count incurred to locate the global optimum is presented for each function. Bold italic also highlights the best search performance.

Level-Type		$F_{Griewank}(10D)$	$F_{Rastrigin}(20D)$	$F_{Sphere}(30D)$ *(Eval. Count)*	$F_{Step}(5D)$ *(Eval. Count)*
Local-Quantitative	SE	***0***	***0***	***3429***	***1552***
External-Static	S-E	5.2500e-3	16.8760	12593	23433
Local-Qualitative	QL1-L	5.2537e-1	84.9772	> 40000	19504
	QL2-L	1.0610e-2	18.6215	8599	8942
Global-Qualitative	QL3-G	4.5000e-4	18.0530	8599	8056
Global-Quantitative	QN1-G	6.2000e-5	9.6078	8193	9653
	QN2-G	6.1060e-3	14.5241	9196	14329
Local-Quantitative	QN3-L	5.5800e-4	33.4929	10194	12007
Local Self-adaptive	S-L	2.8630e-3	14.1689	11792	28100
Canonical MAs	GA-BL	6.1370e-1	92.3340	> 40000	8588
	GA-DP	5.1600e-4	14.4480	9098	8931
	GA-FB	1.9096e+1	144.2500	> 40000	25706
	GA-FL	7.0700e-3	69.8630	6666	> 40000
	GA-GL	2.2646e+1	155.1100	> 40000	25706
	GA-PS	3.3780e-3	74.1060	12292	> 40000
	GA-SK	3.3862e-1	81.1180	40000	> 40000
	GA-SX	7.8610e-1	73.7900	> 40000	> 40000

a brief description of each algorithm provided. Using a statistical T-test of 95% confidence level, the search performances of each algorithms are pitted against the SE on solving the set of benchmark functions described in Appendix A, where the results are tabulated in Table 5.7. The detailed statistics of the different algorithms on numerical errors with respect to the global optimum, are provided separately in Table 5.8. For the sake of fair comparison, note that the accuracy level of convergence ϵ, is configured as $1E-8$, which follows the configuration used by other researchers of earlier studies as a strict guideline.

From the results in Table 5.7, Symbiotic Evolution is observed to fair competitively or significantly outperform most of the state-of-the-art methods considered, at 95% confidence level on the 30-dimensional benchmark functions, particularly on the noisy and rotated shifted multimodal functions. This highlights the robustness, efficacy and superior performance attained by taking advantage of the emergence behavior that is resulted from the interactions of stochastic

Table 5.6: Recent State-of-the-Art Evolutionary Approaches

Abbreviations	**Algorithm Description**
APrMF	Approximate probabilistic memetic framework based on GA-DSCG [132]
MA-LSCh-CMA	Memetic algorithm with local search chaining [122]
DEahcSPX	Accelerating differential evolution using an adaptive local search [134]
LR-CMA-ES	Local restart CMA evolution strategy [7]
IPOP-CMA-ES	Restart CMA evolution strategy with increasing population size [5]
EDA	Continuous Estimation of Distribution Algorithms [223]
RCMA	Adaptive local search parameters for real-coded Memetic Algorithm [121]
RCGA-FMD	Hybrid real-coded Genetic Algorithm with female and male differentiation [46]
SPC-PNX	Steady-state real parameter Genetic Algorithm SPC-PNX [10]
DE	Differential Evolution [163]
CoEVO	Real-parameter optimization using the mutation step co-evolution [152]
PS+CMA-ES	Particle Swarm CMA-ES [126]
PAP	Population-based Algorithm Portfolio based on (DE+PSO+PCX+ES) [150]

variations and individual learning working in symbiosis, as advocated in the symbiotic evolution.

Table 5.7: Results of t-test at 95% confidence level, for SE and other recent evolutionary search algorithms in solving the 30D benchmark problems (Note that s+ or s- indicates that SE is significantly statistically better or not, respectively). $\approx$ is used when both algorithms are able to locate the global optimum at the accuracy level of convergence $1E-8$. N.A indicates that assessment on the function cannot be made, since it was not considered by the authors in their manuscript.

Algorithm	F_{Sphere}	$F_{Schwefel1.2}$	$F_{Elliptic-SR}$	$F_{Schwefel1.2-Noise}$	$F_{Rosenbrock}$	$F_{Griewank-SR}$
APrMF	$\approx$	$\approx$	s-	s+	s+	s+
MA-LSCh-CMA	N.A	N.A	N.A	N.A	s+	s+
DEahcSPX	N.A	N.A	N.A	N.A	$\approx$	s+
LR-CMA-ES	$\approx$	$\approx$	s-	s+	$\approx$	$\approx$
IPOP-CMA-ES	$\approx$	$\approx$	s-	s+	$\approx$	$\approx$
EDA	$\approx$	$\approx$	s-	s-	s+	$\approx$
RCMA	$\approx$	s+	s+	s+	s+	s+
RCGA-FMD	$\approx$	$\approx$	s+	s+	s+	$\approx$
SPC-PNX	$\approx$	s+	s+	s-	s+	s+
DE	$\approx$	s+	s+	s+	s+	s+
CoEVO	s+	s+	s+	s+	s+	s+
PS+CMA-ES	$\approx$	$\approx$	s+	s-	s+	$\approx$
PAP	$\approx$	$\approx$	s-	s+	s+	$\approx$

Algorithm	$F_{Ackley-SR}$	$F_{Rastrigin}$	$F_{Rastrigin-SR}$	$F_{Weierstrass-SR}$	$F_{ExpandedScaffer-SR}$	$F_{Grie+Rosen}$
APrMF	s+	$\approx$	s+	s+	s+	s+
MA-LSCh-CMA	s+	$\approx$	s+	s+	s+	s+
DEahcSPX	s+	$\approx$	s+	s+	s+	s+
LR-CMA-ES	s+	s+	s+	s+	s+	s+
IPOP-CMA-ES	s+	s+	s+	s+	s+	s+
EDA	s+	s+	s+	s+	s+	s+
RCMA	s+	s+	s+	s+	s+	s+
RCGA-FMD	s+	s+	s+	s+	s+	s+
SPC-PNX	s+	s+	s+	s+	s+	s+
DE	s+	s+	s+	s+	s+	s+
CoEVO	s+	s+	s+	s+	s+	s+
PS+CMA-ES	s+	$\approx$	$\approx$	s+	s+	s+
PAP	s+	$\approx$	s+	s+	s+	s-

Table 5.8: Numerical Error to global optimal solution, as attained by SE and the other algorithms considered at $30 * 10^4$ function evaluations on the 30D benchmark functions. N.A indicates that assessment on the function cannot be made, since it was not considered by the authors in their manuscript.

Algorithm	F_{Sphere}	$F_{Schwefel1.2}$	$F_{Elliptic-SR}$	$F_{Schwefel1.2-Noise}$	$F_{Rosenbrock}$	$F_{Griewank-SR}$
SE	1.815E-26	2.901E-10	1.319E+00	6.040E-03	4.186E-09	3.480E-07
APrMF	<1E-8	<1E-8	<1E-8	3.610E+04	8.270E+00	8.760E-04
MA-LSCh-CMA	N.A	N.A	N.A	N.A	1.191E+01	8.871E-04
DEahcSPX	N.A	N.A	N.A	N.A	1.000E-09	1.163E-03
LR-CMA-ES	5.280E-09	6.930E-09	5.180E-09	9.260E+07	6.310E-09	6.480E-09
IPOP-CMA-ES	5.420E-09	6.220E-09	5.550E-09	1.110E+04	5.900E-09	5.310E-09
EDA	9.095E-15	9.095E-15	1.776E-12	2.260E-12	2.110E+01	3.865E-14
RCMA	9.364E-09	8.717E-06	8.775E+05	3.966E+01	4.955E+01	1.329E-02
RCGA-FMD	8.878E-09	9.835E-09	3.112E+03	1.684E+01	2.598E-07	9.073E-09
SPC-PNX	9.352E-09	6.948E-07	1.102E+06	8.132E-07	1.520E+01	1.460E-02
DE	0.000E+00	3.330E-02	6.920E+05	1.520E+01	2.510E+01	2.960E-03
CoEVO	7.965E-01	4.402E-01	3.671E+02	4.798E+03	1.211E+03	1.412E-01
PS+CMA-ES	8.790E-09	9.260E-09	8.000E+04	8.470E-04	1.350E+01	9.330E-09
PAP	1.000E-13	1.000E-13	1.000E-13	2.000E-02	1.060E-07	1.000E-13
Algorithm	$F_{Ackley-SR}$	$F_{Rastrigin}$	$F_{Rastrigin-SR}$	$F_{Weierstrass-SR}$	$F_{ExpandedScaffer-SR}$	$F_{Grie+Rosen}$
SE	7.770E-02	0	8.292E-10	4.700E-01	2.990E-01	1.488E+00
APrMF	2.060E+01	<1E-8	2.170E+02	2.930E+01	5.410E-01	7.720E+00
MA-LSCh-CMA	2.027E+01	7.828E-09	1.839E+01	4.351E+00	1.268E+01	2.345E+00
DEahcSPX	2.095E+01	1.000E-09	9.450E+01	2.922E+01	1.279E+01	2.366E+00
LR-CMA-ES	2.000E+01	2.910E+02	5.630E+02	1.520E+01	1.400E+01	2.320E+00
IPOP-CMA-ES	2.010E+01	9.380E-01	1.650E+00	5.480E+00	1.290E+01	2.490E+00
EDA	2.095E+01	1.787E+02	1.886E+02	3.945E+01	1.332E+01	1.531E+01
RCMA	2.071E+01	6.806E-01	9.058E+01	3.114E+01	1.256E+01	3.963E+00
RCGA-FMD	2.095E+01	1.511E+01	3.520E+01	2.474E+01	1.212E+01	5.152E+00
SPC-PNX	2.093E+01	2.393E+01	6.030E+01	1.126E+01	1.313E+01	3.588E+01
DE	2.100E+01	1.850E+01	9.690E+01	3.420E+01	1.340E+01	3.230E+00
CoEVO	2.090E+01	1.314E+02	2.324E+02	3.770E+01	1.325E+01	9.020E+00
PS+CMA-ES	2.100E+01	8.850E-09	8.980E-09	3.910E+00	1.29E+01	2.110E+00
PAP	2.000E+01	1.000E-13	4.180E+01	8.640E+00	1.24E+01	1.370E+00

5.4 Conclusion

The general practice on manual crafting of dedicated search solvers in computational intelligence community has evolved into the emerging field of self-configurable search algorithms. Adaptation of parameters and operators in search now represents one of the most important, promising and challenging areas of research in the field computational intelligence.

To demonstrate how the study on *symbiosis* of stochastic variation and individual learning in memetic search introduced in Chapter 4 can be put into practice for solving complex problems, a self-configurable Symbiotic Evolution (SE) is proposed in this chapter. SE performed a statistical learning of *evolvability* at runtime to infer the respective productive symbiotic search profile for each individual, leading to the self-configuration of solver that searches effectively on a given problem in hand. Numerical study of the SE with assessment made against several recent state-of-the-art modern evolutionary methods, adaptive and hybrid approaches on representative benchmark problems confirmed the motivation for formalizing the symbiosis of search mechanisms in self-configurable memetic search.

Chapter 6

Evolvability of Surrogates

Engineering reliable and high quality products is now becoming an important practice of many industries to stay competitive in today's increasingly global economy, which is constantly exposed to high commercial pressures. Strong engineering design know-how results in lower time to market and better quality at lower cost. In recent years, advancement in science, engineering and the availability of massive computational power have led to the increasing high-fidelity approaches introduced for precise studies of complex systems *in silico*. Modern Computational Structural Mechanics, Computational Electro-Magnetics, Computational Fluid Dynamics and Quantum mechanical calculations represent some of the approaches that have been shown to be highly accurate [73, 228, 63]. These techniques play a central role in the modelling, simulation and design process since they serve as efficient and convenient alternatives for conducting trials on the original real-world complex system that are otherwise deemed to be too costly or hazardous to construct.

Typically, when high-fidelity analysis codes are used, it is not uncommon for the single simulation process to take minutes, hours to days of supercomputer time to compute. A motivating example at Honda Research Institute is aerodynamic car rear design, where one function evaluation involving a Computational Fluid Dynamics (CFD) simulation to calculate the fitness performance of a potential design can take many hours of wall clock time. Since the design cycle time of a product is directly proportional to the number of calls made to the costly analysis

solvers, researchers are now seeking for novel stochastic optimization approaches, including evolutionary frameworks, that handle these forms of problems elegantly. Besides parallelism, which is an obvious choice to achieving near linear order improvement in evolutionary search, researchers are gearing towards surrogate-assisted or meta-model assisted evolutionary frameworks when handling optimization problems imbued with costly non-linear objective and constraint functions [106, 77, 181, 139, 75, 110, 109, 191, 173, 202, 165].

The general consensus on surrogate-assisted evolutionary frameworks is that the efficiency of the search process can be improved by replacing as often as possible, calls to the costly high-fidelity analysis solvers with surrogates that are deemed to be less costly to build and compute. In this manner, the overall computational burden of the evolutionary search can be greatly reduced since the efforts required to build the surrogates and to use them are much lower than those in the traditional approach that directly couples the evolutionary algorithm (EA) with the costly solvers [171, 16, 76, 83, 42, 78, 196, 18, 226]. Among many data-centric approximation methodologies used to construct surrogates to date, polynomial regression or response surface methodology [100], support vector machine [30, 199], artificial neural networks [229], radial basis function [154], Gaussian process referred to as Kriging or design and analysis of computer experiment models [114, 20] and ensembles of surrogates [225, 48, 164, 1] are among the most prominently investigated [75, 226, 173]. Early proposed approaches have considered using surrogates that target to model the entire solution space or fitness landscape of the costly exact objective or fitness function [107, 108]. However, due to the sparseness of data points collected along the evolutionary search, the construction of accurate global surrogates [196, 20] that mimics the entire problem landscape well is impeded by the effect of "curse of dimensionality" [37]. To enhance the accuracies of the surrogates used, researchers have turned to localized models [47, 42, 139, 160] as opposed to globalized models, or their synergies [226, 109]. Others have also considered the use of gradient information [140] to enhance the prediction accuracy of the constructed surrogate models or physics-based models that are deemed to be more trustworthy than pure data-centric counterparts [81, 111].

In the context of surrogate-assisted optimization [74, 173], present performance or assessment metrics and schemes used for surrogate model selection and validation involve many prominent approaches that have taken root in the field of statistical and machine learning [43, 173]. Particularly, the focus have been placed on attaining surrogate model that has minimal apparent error or training error on some optimization data collected during the evolutionary search, as an estimation of the true error when used to replace the original costly high-fidelity analysis solver. Maximum/Mean Absolute Error, Root Mean Square Error (RMSE) and Correlation Measure denote some of the performance metrics that are commonly used [74]. Typical model selection schemes that stem from the field of statistical and machine learning, including the split sample (holdout) approach, cross-validation and bootstrapping, are subsequently used to choose surrogate models that have low estimation of apparent and true errors [159, 193]. [192] used the multiple cross-validation schemes for the selection of low-error surrogates that replace the original costly high-fidelity analysis solver to avoid convergence at false optima of poor accuracy models.

In spite of the significant research effort spent on optimizing computationally expensive problems more efficiently, existing surrogate-assisted evolutionary frameworks may be fundamentally bounded by the use of apparent or training error as the performance metric used to assess and select surrogates. In contrast, it would be more worthwhile to select surrogates that enhance search improvement in the context of optimization, as opposed to the usual practice of choosing surrogate model with minimal estimated true error. Further, the influence of the data-centric approximation methodology employed is deemed to have a major impact on surrogate-assisted evolutionary search performance. The varied suitability of approximation methodology to different fitness landscapes, state of the search, and characteristics of the search algorithm suggests for the varieties of surrogate-assisted evolutionary frameworks in the literature that have emerged with ad-hoc approximation methodology selection. To the best of our knowledge, little work has been done to mitigate this issue since only limited knowledge of the "black-box" optimization problem is available before one starts.

Falling back on the basics of Darwin's grand idea of "Natural Selection" as the criterion for the choice of surrogates that brings about fitness improvement to the search, this paper describes a novel evolutionary search process that evolves along with *fitness improving* surrogates. Here, the study focuses on the *Evolvability Learning* of Surrogates (EvoLS), particularly the adaptive choice of data-centric approximation methodologies that build fitness improving surrogates in place of the original computationally expensive "black-box" problem, during the evolutionary search. In the spirit of reward-based or improvement-based adaptation for the choice of variation operators [49, 194, 32], EvoLS infers the fitness improvement contribution of each approximation methodology towards the search, which is here referred to as *evolvability* measure. Hence for each individual or design solution in the evolutionary search, the *evolvability* of each approximation methodology is determined statistically, according to the current state of the search, properties of the search operators and characteristics of the fitness landscape, while the search progresses online. Using the *evolvability* measures derived, the search adapts by using the most productive approximation methodology inferred for the construction of surrogates embedded within a trust-region enabled individual learning strategy (often referred to as local search), leading to the self-configuration of surrogate-assisted memetic algorithm that deals with complex optimization of computationally expensive problems more effectively. It is worth noting that to the best of our knowledge, our current effort has not been previously studied in the literature of surrogate-assisted EA.

The chapter is outlined as follows: Section 6.1 introduces the notion of *evolvability* as a performance or assessment measure that expresses the suitability of an approximation methodology in guiding towards improved evolutionary search and subsequently the essential ingredients of our proposed EvoLS. Section 6.2 presents a numerical study of EvoLS on commonly used benchmark problems. Detailed analyses on the suitability and cooperation of surrogates in search, as well as the correlation between the estimated *fitness prediction error* and *evolvability* in EvoLS of the surrogate models, are also presented in the section. Finally, Section 6.3 summarizes the present study with a brief discussion and concludes the chapter.

6.1 Self-configurable Memetic Search: Evolvability Learning of Surrogates

This section presents the essential ingredients of the proposed *Evolvability Learning* of Surrogates (EvoLS) for handling computationally expensive optimization problems. In particular, the framework concentrates on the general nonlinear programming problem of the following form:

$$\textbf{Minimize}: \quad f(\mathbf{x})$$

$$\textbf{Subject to}: \quad \mathbf{x}_l \leq \mathbf{x} \leq \mathbf{x}_u$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of design variables, and $\mathbf{x}_l$, $\mathbf{x}_u$ are vectors of lower and upper bounds, respectively. In this paper, we are interested in cases where the evaluation of $f(\mathbf{x})$ is computationally expensive, and it is desired to obtain a near-optimal solution on a limited computational budget, using a novel evolutionary process that adapts fitness improving surrogates, $\widehat{f}_M(\mathbf{x})$ as a replacement of $f(\mathbf{x})$, in the search.

Data-centric surrogates are (statistical) models that are built to approximate computationally expensive simulation codes or the exact fitness evaluations. They are orders of magnitude cheaper to run and can be used in lieu of the exact analysis during evolutionary search. Let $\{(\mathbf{x}_i, t_i)\}_{i=1}^m$ where $t_i = f(\mathbf{x}_i)$ denote the training dataset, where $\mathbf{x} \in \mathbb{R}^n$ is the input vector of scalars or design parameters, and $f(\mathbf{x}) \in \mathbb{R}$ is the output or exact fitness value. Based on the approximation methodology M, the surrogate $\widehat{f}_M(\mathbf{x})$ is constructed as an approximation model of the function $f(\mathbf{x})$. Further, the surrogate model can also yield insights into the functional relationship between the input $\mathbf{x}$ and the objective function $f(\mathbf{x})$. For a detailed description of some representative approximation methodologies used in the literature, the readers is referred to the Appendix C of the thesis.

In what follows, the section begins with a formal introduction on the notion of *evolvability* as a performance or assessment measure to indicate the productivity and suitability of an approximation methodology for constructing surrogate that brings about fitness improvement to

the evolutionary search (see Section 6.1.1). The essential backbone of the proposed *Evolvability Learning* of surrogates framework is a evolutionary algorithm coupled with a trust-region enabled individual learning strategy with adaptive surrogates, in the spirit of Lamarckian learning. In contrast to existing works, the choice of approximation methodology is adapted for the construction of fitness improving data-centric surrogates in place of the original computationally expensive "black-box" problem when conducting the computationally intensive individual learning (local search) in the context of memetic optimization [56, 89, 137] (see Section 6.1.2).

6.1.1 Evolvability of Surrogate

Conventionally, surrogate models are assessed and chosen according to their estimated true error, $|f(\mathbf{x}) - \widehat{f}_M(\mathbf{x})|$, where $\widehat{f}_M(\mathbf{x})$ denotes the predicted fitness value of input vector $\mathbf{x}$ by a surrogate constructed using approximation method M. In contrast to existing surrogate-assisted evolutionary search, the surrogate model employed for each individual design solution in the present study favors fitness improvement as the choice of merit to assess the usefulness of surrogates in enhancing search improvement, as opposed to the estimated true error.

In this subsection, the concept of "*Evolvability*" of an approximation methodology is introduced as the basis for adaptation. Since the term "*Evolvability*" has been used in different contexts[1], it is worth highlighting that our concept of *evolvability* generalizes from that of learnability in machine learning [197]. Here the evolutionary process is regarded as "*evolvable*" on a given optimization problem if the progress in search performance is observed for some moderate number of generations. Hence *evolvability* of an approximation methodology here is referred to the propensity of the method in constructing a surrogate model that guides towards viable, or "potentially favorable" individuals with improved fitness quality.

In particular, the *evolvability* measure of an approximation methodology M for the construction of a fitness improving data-centric surrogate on individual solution $\mathbf{x}$ at generation t,

[1] In [203], "evolvability" is defined as the genome's ability to produce adaptive variants when acted upon by the genetic system. Others have generally referred the term to the ability of stochastic or random variations to produce improvement for adaptation to happen [138].

assuming a minimization problem, is denoted here as $Ev_M(\mathbf{x}, t)$ and derived in the form of

$$\begin{aligned} Ev_M(\mathbf{x}, t) &= Exp[f(\mathbf{x}) - f(\mathbf{z})|\mathbf{P}^t, \mathbf{x}] \\ &= f(\mathbf{x}) - \int_y f(\varphi_M(\mathbf{y})) \times P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) d\mathbf{y} \end{aligned} \quad \text{(Eq. 6.1)}$$

Here $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ denotes the conditional density function of the stochastic variation operators applied on parent $\mathbf{x}$ to arrive at solution $\mathbf{y}$ at generation t, i.e., $\mathbf{y} \sim P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$, where $\mathbf{P}^t$ is the current reproduction pool consisting of individual solutions after undergoing natural selection. $\varphi_M(\mathbf{y})$ denotes the resultant solution derived by the individual learning strategy that operates on the surrogate constructed based on approximation method M. The *evolvability* measure of an approximation methodology indicates the *expectation* of fitness improvement which the refined offspring, denoted here as $\mathbf{z} = \varphi_M(\mathbf{y})$, has gained over its parent, upon undergoing individual learning improvement on the respective constructed surrogate[2]. A high *evolvability* measure encapsulates two core essences of the surrogate-assisted evolutionary search: 1) When a surrogate exhibits low true error estimates, fitness improvement on the refined offspring $\mathbf{z}$ over initial parent $\mathbf{x}$ can be expected and 2) When a surrogate exhibits high true error estimates, the discovery of offspring solutions with improved fitness $\mathbf{z}$ of $\mathbf{x}$ can be still attained due to the "*blessing of uncertainty*" phenomenon [142] (see Fig. 6.1 for an example illustration).

Taking into account the current state of the evolutionary search, properties of the search operators, and characteristics of the fitness landscape, a statistical learning approach to estimate the *evolvability* measure $Ev_M(\mathbf{x}, t)$ of each approximation methodology (as defined in Eq. 6.1) for use on a given individual solution $\mathbf{x}$ at generation t, is proposed. Let $\Phi_M = \{(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))\}_{i=1}^K$ denote the database of distinct samples archived along the search, which represents the historical contribution of the approximation methodology on the problem considered. Through a weighted sampling approach, the weight $w_i(\mathbf{x})$ that defines the probability of choosing a sample $(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))$ for the estimation of expected improvement $Ev_M(\mathbf{x}, t)$, or

[2]The individual learning strategy serves as a function that takes the starting solution $\mathbf{x}$ as input and returns $\mathbf{z}$ as the refined solution. As such, the function is represented as $\mathbf{z} = \varphi_M(\mathbf{y})$.

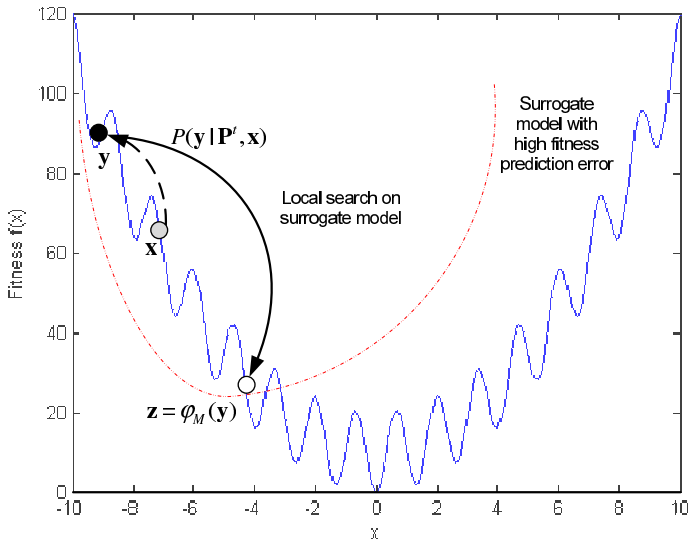


Figure 6.1: Illustration of *evolvability* under the effect of "bless of uncertainty"

the integral $\int_y f(\varphi_M(\mathbf{y})) \times P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})d\mathbf{y}$ in particular, is first derived and efficiently estimated here as follows:

$$w_i(\mathbf{x}) = \frac{P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})}{\sum_{j=1}^{K} P(\mathbf{y}_j|\mathbf{P}^t, \mathbf{x})} \qquad \text{(Eq. 6.2)}$$

The weight $w_i(\mathbf{x})$ essentially reflects the current probability of jumping from solution $\mathbf{x}$, via stochastic variation $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$, to offspring $\mathbf{y}_i$ which was archived in the database. In other words, the weight $w_i(\mathbf{x})$ measures the relevancy of the samples $(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))$ for the *evolvability* learning process on solution $\mathbf{x}$. Considering $\{(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))\}_{i=1}^{K}$ as distinct samples from current distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$, the weights $w_i(\mathbf{x})$ associated with samples $(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))$ satisfy the equations: $\sum_{i=1}^{K} w_i(\mathbf{x}) = 1$ and $w_i(\mathbf{x})$ is proportional to $P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$. Note that the conditional density function $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ is modeled probabilistically based on the properties of the evolutionary variation operators used to reflect the current state of the search. Subsequently, using the archived samples in $\Phi_M = \{(\mathbf{y}_i, \varphi_M(\mathbf{y}_i))\}_{i=1}^{K}$ and weights w_i obtained using Eq. 6.2, $Ev_M(\mathbf{x}, t)$ is estimated as follows:

$$Ev_M(\mathbf{x}, t) = f(\mathbf{x}) - \sum_{i=1}^{K} f(\varphi_M(\mathbf{y}_i)) \times w_i(\mathbf{x}) \qquad \text{(Eq. 6.3)}$$

6.1.2 Evolution with Adapting Surrogates

The proposed *Evolvability Learning* of Surrogates (EvoLS) for solving computationally expensive optimization problems is presented and outlined in Algorithm 7. The essential ingredients of our proposed EvoLS framework, are composed of *multiple* data-centric approximation methodologies having diverse characteristics[3], denoted here as $\{M_{id}\}_{id=1}^{ID}$. In the first step, a population of N individuals is initialized either randomly or using design of experiment techniques such as Latin hypercube sampling. The cost or fitness value of each individuals in the population is then determined using $f(\mathbf{x})$. The evaluated population then undergoes natural selection, for instance, via fitness-proportional or tournament selection. Each individual $\mathbf{x}$ in the reproduction pool $\mathbf{P}^t$ is evolved to arrive at the offspring $\mathbf{y}$ using stochastic variation operators including crossover and mutation. Subsequently, with ample design points in the database Ψ or after some predefined database building phase of exact evaluations G_{db} (line 4), the trust-region enabled individual learning with adaptive surrogate kicks in for each non-duplicated design point or individuals in the population. For a given individual solution $\mathbf{x}$ at generation t, the *evolvability* $Ev_{M_{id}}(\mathbf{x}, t)$ of each data-centric approximation methodology is estimated statistically by taking into account the current state of the search, properties of search operators and characteristics of the fitness landscape via the historical contribution by the respective constructed surrogates, while the search progresses online[4]. Without loss of generality, in the event of a minimization problem, the most productive data-centric approximation methodology, which is deemed as one that has the highest estimated *evolvability* measure $\arg\max Ev_{M_{id}}(\mathbf{x}, t)$, is then chosen to construct a surrogate that will be used by the individual learning to bring about fitness improvement on individual $\mathbf{x}$.

[3]Typical approximation techniques are Radial Basic Function (RBF), Kriging or Gaussian process (GP) and Polynomial Regression (PR).

[4]Note that a uniform selection of approximation methodologies is employed in EvoLS instead of Algorithm 8 in the first generation right after the database building phase. In this way, all approximation methodologies are ensured to be given equal opportunities to work on each individual in the population. Thus sufficient data points are expected to be made available for the evolvability learning of each approximation methodology in the subsequent generations.

Algorithm 7 Evolvability Learning of Surrogates (EvoLS) Framework

1: *Generate* and *evaluate* an initial population
2: **while** computational budget is not exhausted **do**
3: *Select* individuals for the reproduction pool $\mathbf{P}^t$
4: **if** evaluation count $<$ database building phase (G_{db}) **then**
5: *Evolve* population by evolutionary operators (crossover, mutation)
6: **else**
7: *Evolve* population by evolutionary operators (crossover, mutation)
8: **for** each individual $\mathbf{x}$ in the population **do**
9: *Perform* approximation methodology selection on $\mathbf{x}$ to arrive at M_{id}
10: /*** Individual Learning Phase on Surrogate Model***/
11: *Find* m nearest points to $\mathbf{y}$ in the database Ψ as training points for surrogate model
12: *Build* surrogate model $\widehat{f}_{M_{id}}(\mathbf{x})$ based on training points
13: *Apply* individual learning strategy $\varphi_{M_{id}}(\mathbf{y})$ to arrive at $\mathbf{z}$
14: *Replace* $\mathbf{y}$ with $\mathbf{z}$ (Lamarckian learning)
15: *Archive* sample $(\mathbf{y}, \varphi_{M_{id}}(\mathbf{y}))$ into the database $\Phi_{M_{id}}$
16: **end for**
17: **end if**
18: *Evaluate* new population using exact fitness function $f(\mathbf{x})$
19: *Archive* all exact evaluations $(\mathbf{x}, f(\mathbf{x}))$ into the database Ψ
20: **end while**

The outline of the approximation methodology selection process is detailed in Algorithm 8. It is worth noting on the generality of the proposed framework in the use of density function $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ to represent and reflect the unique characteristics of the stochastic search operators (line 1), thus not restricting to any specific type of operator. By formulating the search operator with a density function, the framework allows the incorporation of different stochastic variation operators, which depends on suitability to the given problem of interest. It is worth highlighting that the condition $\sum_{i=1}^{K} w_i(\mathbf{x}) < \epsilon$ (line 5) caters for the scenario when all samples $(\mathbf{y}_i, \varphi_{M_{id}}(\mathbf{y}_i))$ are irrelevant for evolvability estimation on solution $\mathbf{x}$, i.e., $P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$ is too small. The role of ϵ thus specifies the threshold level of irrelevance for archived samples in the evolvability learning. In particular, ϵ is configured to a precision of $1E-9$ in our present study.

Rather than constructing a global surrogate based on the entire archived samples, nearest sampled points to $\mathbf{y}$ in the database Ψ are selected as training dataset $T = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{m}$

for building local fitness improving surrogate $\widehat{f}_{M_{id}}$. The improved solution found using the respective constructed surrogate, denoted here as $\mathbf{z} = \varphi_{M_{id}}(\mathbf{y})$, is subsequently evaluated using the original computationally expensive fitness function $f(\mathbf{x})$ and replaces the parent individual in the population, in the spirit of Lamarckian learning. Exact evaluations of all newly found individuals $\{(\mathbf{z}, f(\mathbf{z}))\}$, together with $\{(\mathbf{y}, \varphi_{M_{id}}(\mathbf{y}))\}$ are then archived into the database Ψ and $\Phi_{M_{id}}$, respectively, for subsequent use in the search. The entire process repeats until the specified stopping criteria are satisfied.

Algorithm 8 Approximation Methodology Selection Process

Construct density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ of variation operators
for each approximation methodology M_{id} **do**
  Query archived data $\Phi_{M_{id}} = \{(\mathbf{y}_j, \varphi_{M_{id}}(\mathbf{y}_j))\}$ for M_{id}
  Calculate weight $w_i(\mathbf{x}) = P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$ for each sample $\mathbf{y}_i$
  if $\sum_{i=1}^{K} w_i(\mathbf{x}) < \epsilon$ **then**
    $w_i(\mathbf{x}) = 0$ {No relevant data available}
    $Ev_{M_{id}}(\mathbf{x}, t) = -\infty$
  else
    $w_i(\mathbf{x}) = w_i(\mathbf{x}) / \sum_{j=1}^{K} w_j(\mathbf{x})$ {Normalize w_i}
    $Ev_{M_{id}}(\mathbf{x}, t) = f(\mathbf{x}) - \sum_{i=1}^{K} f(\varphi_{M_{id}}(\mathbf{y}_i)) \times w_i(\mathbf{x})$ {Eq. 6.3}
  end if
end for
if $Ev_{M_{id}}(\mathbf{x}, t) < 0 \; \forall \; M_{id}$ **then**
  Select approximation methodology randomly
else
  Select approximation methodology with highest $Ev_{M_{id}}(\mathbf{x}, t)$ for $\mathbf{x}$
end if

6.1.3 Complexity Analysis and Parameters of EvoLS Framework

Here the computational complexity of present conventional surrogate selection schemes that take roots in the fields of statistical and machine learning [43, 159, 192] is first discussed. In conventional surrogate selection schemes, multiple sets of sample data are generally segregated, typically into training and test sets. For each approximation methodology, the respective

surrogate model is commonly constructed based on the training set and the true error is estimated using the test set in the prediction process. This procedure of computation cost C_M is typically repeated for k times on different training and test sets to arrive at a statistically sound estimation of the approximation error that is then used in the selection scheme. Although many error estimation approaches are in abundance, the major differences lie mainly on how the training and test sets are generated, which vary from random subsampling (holdout), k-fold cross-validation and bootstrapping as described in [84]. For ID number of approximation methodologies considered, the overall computational complexity of the conventional selection scheme in estimating the error of the surrogates can thus be derived as $O(ID \times k \times C_M)$.

Next, a complexity analysis of the EvoLS framework is detailed. Apart from the standard parameters of a typical surrogate-assisted evolutionary algorithm [109], EvoLS has two additional parameters: database $\Phi_{M_{id}}$ and density function $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$. Typically, the form of $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ can be explicitly defined according to the stochastic operator used (as illustrated in Section 6.2), while databases $\Phi_{M_{id}}$ naturally follows a first-in-first-out queue structure to favor more recently archived optimization data $\{(\mathbf{y}, \varphi_{M_{id}}(\mathbf{y}))\}$. The complexity for *evolvability* learning of surrogates can be derived as $O(ID \times |\Phi_{M_{id}}| \times C_E)$ where $|\Phi_{M_{id}}|$ denotes the database size, ID denotes the number of approximation methodologies used, and C_E is the computational effort incurred to determine $P(\mathbf{y}_i|\mathbf{P}^t, \mathbf{x})$ for each $\mathbf{y}_i$. For each individual, since only the most productive approximation methodology inferred is used to construct a new surrogate at a computational requirement of C_M, the complexity of EvoLS can be derived as $O(ID \times |\Phi_{M_{id}}| \times C_E + C_M)$. Nevertheless, as $(ID \times |\Phi_{M_{id}}| \times C_E) << C_M$ in practice, the computational complexity of EvoLS becomes $O(C_M)$. Thus, EvoLS offers an alternative to the conventional selection scheme with a significantly lower complexity of $O(C_M)$ that is independent of the number of approximation methodologies considered in the framework.

6.2 Empirical Study

In this section, the numerical results obtained by the proposed EvoLS using three commonly used approximation methodologies, namely: 1) interpolating linear spline Radial Basic Function (RBF), 2) 2^{nd} order Polynomial Regression (PR) and 3) interpolating Kriging/ Gaussian Process (GP), are presented. For the details on GP, PR and RBF, the reader is referred to Appendix C. Representative *30* dimensional benchmark functions considered in the present study are summarized in Table 6.1 while the algorithmic parameters of EvoLS are summarized in Table 6.2.

Table 6.1: Benchmark problems considered in the empirical study. On shifted rotated problems, note that $\mathbf{z} = \mathbf{M} \times (\mathbf{x} - \mathbf{o})$ where $\mathbf{M}$ is the rotation matrix and $\mathbf{o}$ is the shifted global optimum. Otherwise, $\mathbf{z} = \mathbf{x}$.

Function	**Benchmark test functions**	**Range of x**	**Multi***	**Non-sep***
Ackley (F1)	$F_{Ackley} = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)}$	$[-32, 32]^n$	Yes	Yes
Griewank (F2)	$F_{Griewank} = 1 + \sum_{i=1}^{n} z_i^2/4000 - \prod_{i=1}^{n} \cos(z_i/\sqrt{i})$	$[-600, 600]^n$	Yes	Yes
Rosenbrock (F3)	$F_{Rosenbrock} = \sum_{i=1}^{n-1}(100 \times (z_{i+1} - z_i^2)^2 + (1 - z_i)^2)$	$[-2.048, 2.048]^n$	Yes	Yes
Shifted Rotated Rastrigin (F4)	$F_{Rastrigin-SR} = 10n + \sum_{i=1}^{n}(z_i^2 - 10\cos(2\pi z_i))$	$[-5, 5]^n$	Yes	Yes
Shifted Rotated Weierstrass (F5)	$F_{Weierstrass-SR} = \sum_{i=1}^{n}(\sum_{k=0}^{k_{max}}(a^k \cos(2\pi b^k(z_i + 0.5)))) - n\sum_{k=0}^{k_{max}}(a^k \cos(\pi b^k))$ $a = 0.5, b = 3, k_{max} = 20$	$[-0.5, 0.5]^n$	Yes	Yes
Expanded Griewank plus Rosenbrock (F6)	$F_{Grie+Rosen} = \sum_{i=1}^{D} F_{Griewank}(F_{Rosenbrock}(z_i, z_{i+1}))$ $z_{D+1} = z_1$	$[-3, 1]^n$	Yes	Yes

The stochastic variation operators considered in the present study are uniform crossover and mutation, which have been widely used in real-coded genetic evolution. For the sake of

brevity, the use of simple variation operators is considered in the illustrating study to showcase the mechanism and generality of the proposed EvoLS. It is worth noting that other advanced real-parameter search operators, such as that discussed in [60], can also be considered in the framework through density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$.

In particular, the crossover procedure considered here is outlined as follows: a) randomly select two solutions, $\mathbf{x}$ and $\mathbf{x}'$ from the population; b) perform uniform crossover on $\mathbf{x}$ and $\mathbf{x}'$ to create two offspring $\mathbf{y}_1$ and $\mathbf{y}_2$ where the locus i of offspring $\mathbf{y}_1^{(i)}$/$\mathbf{y}_2^{(i)}$ has value $\mathbf{x}_1^{(i)}$/$\mathbf{x}_2^{(i)}$ with crossover probability of p_{cross}, and c) select $\mathbf{y} = \mathbf{y}_1$ or $\mathbf{y}_2$ as the offspring of $\mathbf{x}$. Uniform mutation is conducted on $\mathbf{y}$ such that each locus $\mathbf{y}^{(i)}$ is assigned to a random value bounded by $[\min_{j=1...N}\{\mathbf{x}_j^{(i)}\}, \max_{j=1...N}\{\mathbf{x}_j^{(i)}\}]$ with a mutation probability of p_{mut}. Note that N denotes the population size.

The task of deriving exact density distribution function is non-trivial due to the complexity of the search dynamics. Thus a simplified assumption of uniformity in the offspring distribution is considered for practical purposes in our empirical study. The density distribution $P(\mathbf{y}|\mathbf{P}^t, \mathbf{x})$ is then modeled as a uniform distribution with

$$P(\mathbf{y}|\mathbf{P}^t, \mathbf{x}) = UniformDist(\mathbf{R}) = \begin{cases} \frac{1}{Vol(\mathbf{R})} & \textit{if } \mathbf{y} \in \mathbf{R} \\ 0 & \textit{otherwise} \end{cases} \quad \text{(Eq. 6.4)}$$

where $Vol(\mathbf{R})$ denotes the hyper-volume of hyper-rectangle with bounds $\mathbf{R}$ defined as

$$\mathbf{R} = [\min_{j=1...N}\{\mathbf{x}_j^{(i)}\}, \max_{j=1...N}\{\mathbf{x}_j^{(i)}\}]_{i=1...n}$$

Note that the stochastic variations considered impose the resultant offspring $\mathbf{y}$ to be bounded by $\min_{j=1...N}\{\mathbf{x}_j^{(i)}\}$ and $\max_{j=1...N}\{\mathbf{x}_j^{(i)}\}$ for each dimension[5], i.e., $\forall i = 1 \ldots n$. Since the hyper-rectangle $\mathbf{R}$ reduces as the search progresses, the probabilistic model of the variation operators reflects well on the refinement of the search space throughout the evolution.

[5]If $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{y}$ denote the parents and offspring, then each locus of the offspring $\mathbf{y}$ satisfies the inequality

$$\min\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\} \leq \mathbf{y}^{(i)} \leq \max\{\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}\}, \forall i = 1 \ldots n \quad \text{(Eq. 6.5)}$$

On the other hand, the individual learning strategy of EvoLS involves a trust-region framework [142] with Broyden-Fletcher-Goldfarb-Shanno (L-BFGS-B) method [227]. Note that under mild assumptions, trust-region framework ensures theoretical convergence to the local optimum or stationary point of the exact objective function, despite the use of surrogate models [3, 139]. For each individual $\mathbf{y}_i$ in the population, the individual learning method L-BFGS-B proceeds on the inferred most productive surrogate model $\widehat{f}_M(\mathbf{x})$ to perform a sequence of trust-region sub-problems of the form:

$$\begin{aligned} \textbf{Minimize}: \quad & \widehat{f}_M(\mathbf{y} + \mathbf{y}_i^k) \\ \textbf{Subject to}: \quad & ||\mathbf{y}|| \leq \Omega^k \end{aligned}$$

where $k = 0, 1, 2, \ldots k_{max}$, $\widehat{f}_M(\mathbf{y})$ denotes the approximate function corresponding to the original fitness function $f(\mathbf{y})$, $\mathbf{y}_i^k$ and Ω^k denote the starting point and trust-region radius at iteration k, respectively. For each individual $\mathbf{y}_i$, the surrogate model $\widehat{f}_M(\mathbf{y})$ of the original fitness function is created dynamically using training data from archived database $\Psi = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{Q}$ to estimate the fitness during individual learning. Note that $\mathbf{y}_i^{k+1} = \arg\min \widehat{f}_M(\mathbf{y} + \mathbf{y}_i^k)$ denotes the local optimum of the trust-region sub-problem at iteration k. At each kth iteration, $\mathbf{y}_i^{k+1}$ and the trust-region radius Ω^k are updated accordingly. In the present study, the resultant individual denoted here as $\mathbf{y}_i^{k_{max}} = \varphi_M(\mathbf{y}_i)$, is the improved solution attained by L-BFGS-B over surrogate model $\widehat{f}_M(\mathbf{x})$.

6.2.1 Search Quality and Efficiency of EvoLS

To see how adapting the choice of most productive approximation methodology affects the performance and efficiency of the search as compared to the use of single approximation method, in this subsection, the performance of Symbiotic Evolution is compared with the canonical surrogate-assisted evolutionary algorithms (SAEAs) with single approximation methodology, i.e., EA-RBF, EA-PR and EA-GP. Note that the procedure of canonical SAEA is consistent to

Table 6.2: Algorithm Configuration

Parameters	
Population size	100
Selection scheme	Roulette wheel
Stopping criteria	8000 evaluations
Individual learning method	L-BFGS-B
Number of trust region iteration	3
Crossover probability (p_{cross})	1
Mutation probability (p_{mut})	0.01
Variation operator	Uniform crossover and mutation
Database building phase (G_{db})	2000 evaluations
Precision Accuracy	1E-8

the EvoLS outlined in Algorithm 7, except that the former lacks any approximation methodology selection mechanism (line 11). In addition, EA-Perfect, which refers to a canonical SAEA that employs an imaginary approximation method that generates error-free surrogates[6] is also considered here to assess the benefits of using *evolvability* measure versus approximation errors as the criterion for model selection.

The averaged convergence trends obtained by EvoLS on the benchmark problems as a function of the total number of exact fitness function evaluations are summarized in Figs. 6.2.a-6.2.f. The results presented here are the average of 20 independent runs for each test problem. Also shown in the figures are the averaged convergence trends obtained using the canonical surrogate-assisted evolutionary algorithms (SAEAs), i.e., EA-RBF, EA-PR, EA-GP and EA-Perfect. To facilitate a fair comparison study, the parametric configurations of EvoLS and SAEAs are maintained consistent, as defined in Table 6.2. In addition, the proposed EvoLS is further assessed against other recent state-of-the-art evolutionary approaches reported in the literature. In particular, GS-SOMA [109] and IPOP-CMA-ES[7] [6], which are well established

[6] An error-free surrogate model is realized by using exact fitness function for evaluation inside the individual learning strategy where a surrogate model should be used. Note that the incurred fitness evaluation here is not counted as part of the computational budget.

[7] MATLAB implementation, version 3.55.beta January 2011, available at http://www.lri.fr/ hansen/cmaes.m

Table 6.3: Results of Wilcoxon test at 95% confidence level, for EvoLS and other SAEAs in solving the 30D benchmark problems. Note that s+, s- or $\approx$ indicates that EvoLS is significantly statistically better, worse or indifferent, respectively.

Algorithm	EA-GP	EA-PR	EA-RBF	EA-Perfect	GS-SOMA	IPOP-CMA-ES
F_{Ackley}	s+	$\approx$	s+	s+	s+	s+
$F_{Griewank}$	s+	s-	s+	s-	$\approx$	s+
$F_{Rosenbrock}$	s-	s+	s+	s-	s+	s+
$F_{Rastrigin-SR}$	s+	s+	s+	s+	s+	s+
$F_{Weiertrass-SR}$	s+	$\approx$	s+	s+	s+	s+
$F_{Grie+Rosen}$	s+	s+	$\approx$	s+	s+	s+

evolutionary algorithms designed for numerical optimization under the scenarios of limited computational budget, are included here as the state-of-the-art algorithms for comparison. Note that all algorithms considered are configured to their default parametric settings in [109, 6] to facilitate a fair comparison in our study[8]. Using a statistical Wilcoxon test at 95% confidence level, the search performances of each algorithm are pitted against the EvoLS on solving the set of benchmark functions described in Table 6.1, where the results are tabulated in Table 6.3. The detailed statistics of the algorithms from 20 independent runs on the numerical errors with respect to the global optimum are provided separately in Table 6.10.

Statistical analysis shows that EvoLS fares competitively or significantly outperforms most of the methods considered, at 95% confidence level on the 30-dimensional benchmark problems. In the ideal case of error-free surrogate model, denoted by EA-Perfect, the results indicated that the search on surrogate model with low approximation error does not always lead to better performance over the others. As one would expect, the EA-Perfect exhibits superior performance on the low-modality Rosenbrock (with 2 local optima [172]) and Griewank (which has a relatively smooth fitness landscape at high dimensionalities [94]), as observed in Figs. 6.2.b-6.2.c, due to the high efficacy of the individual learning strategy used on a *perfect* or

[8]Note that the population size was configured to 100 for GS-SOMA and $(50, 100)$ for IPOP-CMA-ES in the study.

error-free model. However, on problems with highly multi-modal and rugged landscapes including Ackley, Rastrigin-SR, Weierstrass-SR and expanded Griewank plus Rosenbrock (see Figs. 6.2.a, 6.2.d-6.2.f), EvoLS which operates on inferring the most suitable approximation method according to their *evolvability* measure, is observed to outperform EA-Perfect significantly.

Focusing on the Griewank function, it is worth noting that EvoLS exhibits more robust search than EA-PR, as indicated by the relatively lower standard deviation in solution quality (as shown in Table 6.10). Similar robustness in the EvoLS over EA-GP can also be observed on the Rosenbrock function. On the other hand, the competitive performance displayed by EA-PR and EvoLS on 3 out of the 6 benchmark problems suggests both are benefitting well from the landscape smoothing effect of low-order polynomial regression. Such a phenomenon is established as the "*blessing of uncertainty*" in SAEA [109]. Overall, the results from the statistical tests confirmed the robustness and superiority of the EvoLS over those that assumed a single fixed approximation methodology throughout the search (EA-GP, EA-PR, EA-RBF and EA-Perfect). Last but not least, from the statistical results in Table 6.3, EvoLS is also observed to fare competitively or significantly outperforms the state-of-the-art methods GS-SOMA and IPOP-CMA-ES, at 95% confidence level on the 30-dimensional benchmark problems, thus highlighting the robustness and superiority of EvoLS.

In Table 6.4, the percentage of savings in computational budget by EvoLS (in terms of the number of calls made to the original computational expensive function) to arrive at equivalent solution quality to the respective algorithms on the 30D benchmark problems, are tabulated. Note that only those algorithms that are significantly outperformed by EvoLS on the respective benchmark problem are discussed here. The superior efficiency of the EvoLS over the counterpart algorithms can be observed from the table, where significant savings of 25.32% to 88.41% are noted.

Table 6.4: Percentage of savings in computational budget by EvoLS on the benchmark problems, in terms of the number of calls made to the original computational expensive function to arrive at equivalent converged solution quality of the respective algorithms.

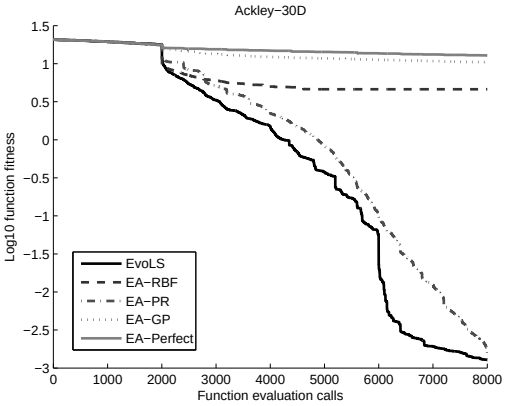
Algorithm	EA-GP	EA-PR	EA-RBF	EA-Perfect	GS-SOMA	IPOP-CMA-ES
F_{Ackley}	74.60	-	66.91	74.91	62.01	47.83
$F_{Griewank}$	74.76	-	74.76	-	-	74.76
$F_{Rosenbrock}$	-	50.27	55.20	-	52.88	40.25
$F_{Rastrigin-SR}$	49.64	55.31	26.33	25.32	53.85	59.45
$F_{Weiertrass-SR}$	58.57	-	42.28	63.32	41.72	88.41
$F_{Grie+Rosen}$	47.67	45.05	-	44.13	44.99	74.76

Table 6.5: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

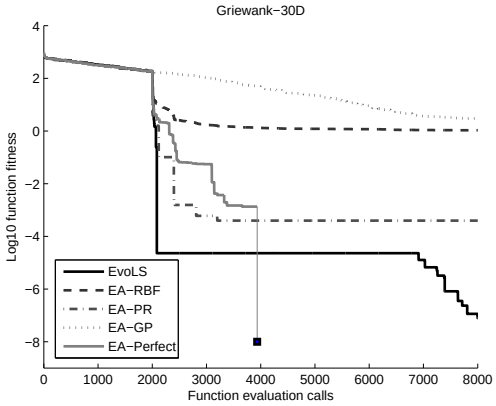
Optimization	**Ackley (F1)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	1.05E+01	4.42E+00	1.23E+01	1.17E-03	1.53E+01
EA-PR	1.54E-03	8.34E-04	1.30E-03	4.93E-04	3.39E-03
EA-RBF	4.62E+00	1.67E+00	4.47E+00	2.66E+00	6.40E+00
EA-Perfect	1.28E+01	1.17E+00	1.30E+01	9.73E+00	1.42E+01
GS-SOMA	3.58E+00	5.09E-01	3.67E+00	2.87E+00	4.28E+00
IPOP-CMA-ES	1.08E+00	1.34E+00	6.01E-01	1.50E-01	5.67E+00
EvoLS	1.28E-03	9.84E-04	1.13E-03	1.32E-04	3.40E-03

Table 6.6: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

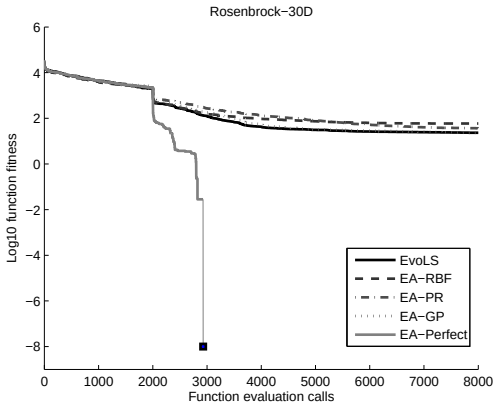
Optimization	**Griewank (F2)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	2.67E+00	1.12E+01	1.88E-02	4.02E-05	5.02E+01
EA-PR	3.96E-04	1.27E-03	3.77E-09	4.34E-10	5.04E-03
EA-RBF	1.07E+00	3.03E-01	1.10E+00	2.43E-01	1.47E+00
EA-Perfect	6.02E-019	1.51E-019	5.96E-019	3.25E-019	8.67E-019
GS-SOMA	2.20E-03	4.60E-03	8.95E-09	1.40E-10	1.54E-02
IPOP-CMA-ES	6.14E-01	2.72E-01	6.06E-01	2.23E-01	1.05E+00
EvoLS	7.89E-08	2.80E-07	1.12E-08	1.71E-09	1.17E-06



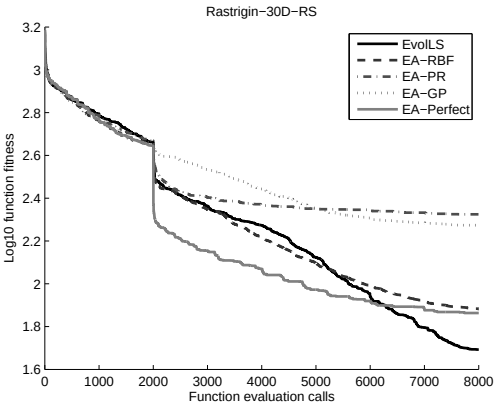
6.2.a: **Ackley-30D**



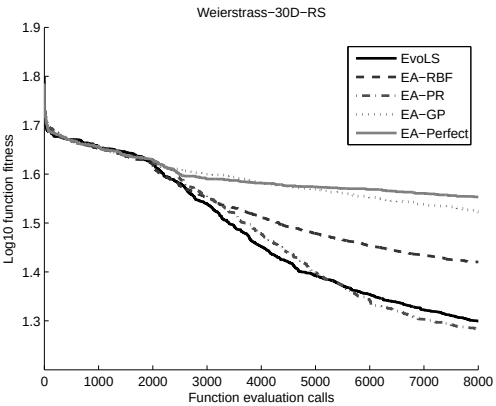
6.2.b: **Griewank-30D**



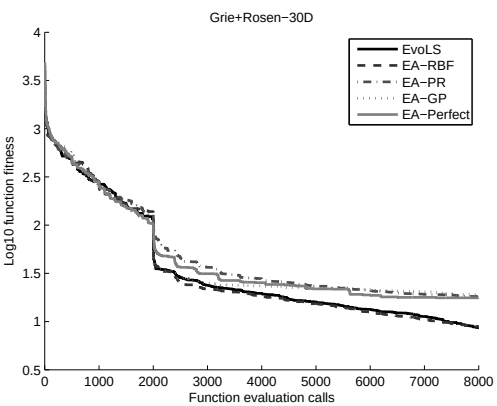
6.2.c: **Rosenbrock-30D**



6.2.d: **Rastrigin-30D-RS**



6.2.e: **Weierstrass-30D-RS**



6.2.f: **Grie+Rosen-30D**

Figure 6.2: Performance of EvoLS and SAEAs on the benchmark problems

Table 6.7: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

Optimization	**Rosenbrock (F3)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	2.29E+01	1.76E+01	1.92E+01	1.47E+01	9.70E+01
EA-PR	3.63E+01	2.29E+01	2.81E+01	2.66E+01	1.18E+02
EA-RBF	5.90E+01	2.15E+01	5.57E+01	3.07E+01	1.02E+02
EA-Perfect	5.65E-18	2.04E-17	1.05E-18	2.96E-019	9.22E-17
GS-SOMA	4.63e+01	2.92e+01	3.02e+01	2.83e+01	1.26e+02
IPOP-CMA-ES	2.84E+01	5.12E-01	2.85E+01	2.72E+01	2.92E+01
EvoLS	2.32E+01	1.66E+00	2.29E+01	2.11E+01	2.66E+01

Table 6.8: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

Optimization	**Shifted Rotated Rastrigin (F4)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	1.88E+02	7.42E+01	1.79E+02	6.67E+01	3.44E+02
EA-PR	2.11E+02	1.36E+01	2.12E+02	1.86E+02	2.39E+02
EA-RBF	7.63E+01	2.86E+01	8.02E+01	3.43E+01	1.38E+02
EA-Perfect	7.30E+01	1.54E+01	6.99E+01	4.97E+01	1.09E+02
GS-SOMA	2.04E+02	1.60E+01	2.07E+02	1.66E+02	2.30E+02
IPOP-CMA-ES	2.28E+02	1.76E+01	2.35E+02	1.91E+02	2.51E+02
EvoLS	4.93E+01	1.66E+01	4.73E+01	2.36E+01	8.16E+01

Table 6.9: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

Optimization	**Shifted Rotated Weierstrass (F5)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	3.33E+01	4.42E+00	3.42E+01	2.44E+01	3.99E+01
EA-PR	1.91E+01	2.61E+00	1.93E+01	1.38E+01	2.37E+01
EA-RBF	2.63E+01	2.97E+00	2.63E+01	2.08E+01	3.29E+01
EA-Perfect	3.57E+01	2.64E+00	3.58E+01	3.15E+01	4.12E+01
GS-SOMA	2.60E+01	3.05E+00	2.90E+01	2.40E+01	3.40E+01
IPOP-CMA-ES	4.62E+01	6.47E+00	4.87E+01	2.97E+01	5.26E+01
EvoLS	1.99E+01	2.69E+00	1.97E+01	1.52E+01	2.52E+01

Table 6.10: Statistics of the converged solution quality at the end of 8000 exact function evaluations for SAEAs, GS-SOMA, IPOP-CMA-ES and EvoLS on benchmark problems.

Optimization	**Expanded Griewank plus Rosenbrock (F6)**				
Algorithm	Mean	Std. Dev.	Median	Best	Worst
EA-GP	1.90E+01	4.58E+00	1.87E+01	1.20E+01	2.81E+01
EA-PR	1.81E+01	1.04E+00	1.81E+01	1.61E+01	2.04E+01
EA-RBF	8.85E+00	2.03E+00	9.25E+00	5.84E+00	1.17E+01
EA-Perfect	1.76E+01	6.26E+00	1.66E+01	9.27E+00	3.67E+01
GS-SOMA	1.80E+01	1.05E+00	1.77E+01	1.70E+01	1.90E+01
IPOP-CMA-ES	1.93E+01	1.17E+00	1.93E+01	1.66E+01	2.15E+01
EvoLS	8.60E+00	1.78E+00	8.77E+00	6.27E+00	1.15E+01

6.2.2 Suitability of Surrogates

In this subsection, the suitability of surrogates in evolutionary search with respect to the benchmark problem of interest, prediction quality and states of the evolution are investigated and discussed.

6.2.2.1 Fitness Landscapes

The summarized search performances of the algorithms with single approximation methodology (i.e., RBF, PR or GP), as tabulated in Table 6.10, confirm our hypothesis that the suitability of surrogates in an evolutionary search largely depends on problem fitness landscape. For the purpose of discussion here, we focus the attention on the results for EA-PR and EA-RBF. EA-PR is observed to outperform other SAEAs with single approximation method (GP or RBF) on Ackley and Griewank, while EA-RBF emerging as superior on Rastrigin-SR and expanded Griewank plus Rosenbrock. Through adapting the choice of suitable approximation methodology by means of *evolvability* learning while the search progresses online, EvoLS is noted to attain search performances that are better or at least competitive to the best performing SAEA (with single approximation method) for the respective problems considered.

6.2.2.2 Prediction Quality and Fitness Improving Surrogate

The correlations between prediction quality and surrogate suitability in bringing about the search improvement in EvoLS are further discussed in this subsection. The correlations can be assessed by analyzing the average frequency of usage and the normalized root mean square fitness prediction errors (N-RMSE) of the surrogates (or the associated approximation methodology used to construct the surrogate), across the search[9], such as that depicted in Figs. 6.3 and 6.4, respectively. From Figs. 6.3 and 6.4, it is notable that the low-error fitness prediction surrogates constructed using RBF have been beneficial in enhancing the search towards the refinement of solutions with improved fitness. The presence of fitness improving surrogates thus naturally led to the high frequency of usage of RBF in EvoLS . Conversely, negative correlations between fitness prediction quality and surrogate suitability may also occur to benefit search. Here, one such scenario is observed in the study for illustration purpose. In spite of the high-error fitness predictions exhibited by the 2nd-order polynomial regression (PR) on Griewank (as can be observed in Figs. 6.3 and 6.4), the persistently high frequency of usage for PR in EvoLS clearly depicts the inference towards productive fitness improving over fitness error-free surrogates.

6.2.2.3 State of Evolution

To assess the suitability of the surrogates at different stages of the evolution, the fitness prediction errors of the surrogates and their frequencies of usage in the EvoLS search are further next analyzed and summarized in Figs. 6.5.a-6.5.f. The upper subplot of each figure depicts how often an approximation methodology is inferred as most productive and hence chosen for use within the individual learning strategy of the EvoLS. The lower subplot, on the other hand, depicts the root mean square fitness prediction errors of the respective surrogates. The

[9]For each generation t, the root mean square (RMS) prediction errors of all methodologies (GP, PR and RBF) are first computed across the independent runs and subsequently normalized, thus bringing the prediction errors to the common scale of [0, 1] throughout the different stages of the search. Subsequently, for each problem, the normalized RMSEs of GP, PR and RBF are averaged across all generations of the entire search.

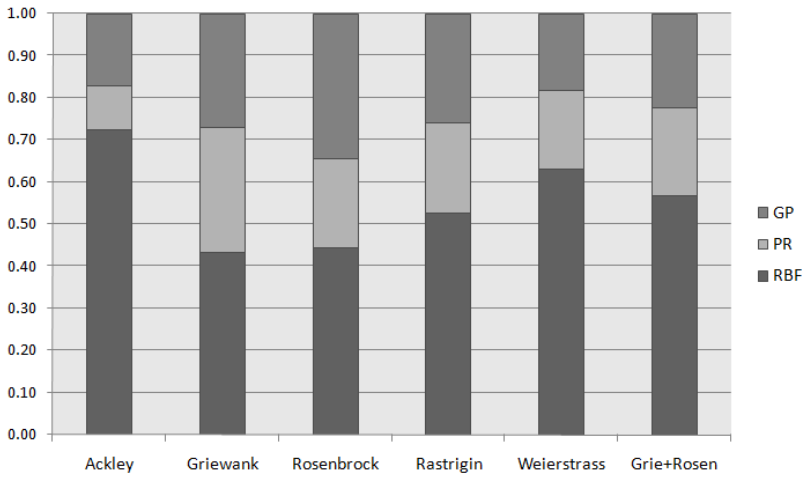


Figure 6.3: **Frequency** of usage of surrogate models on benchmark problems.

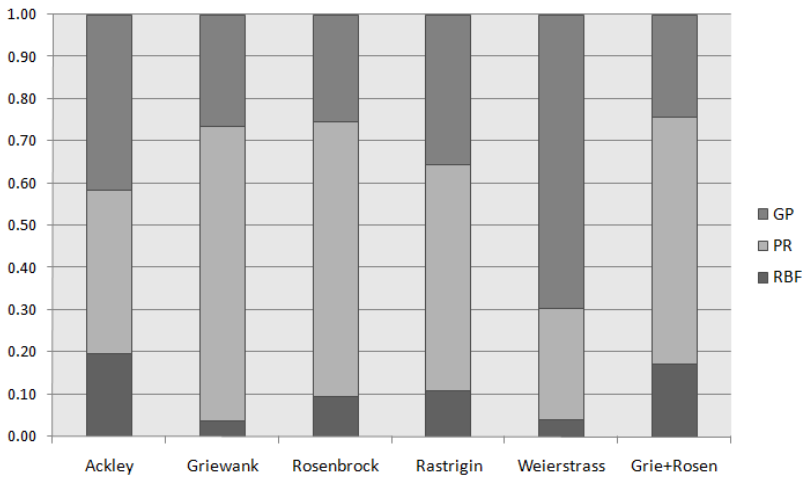
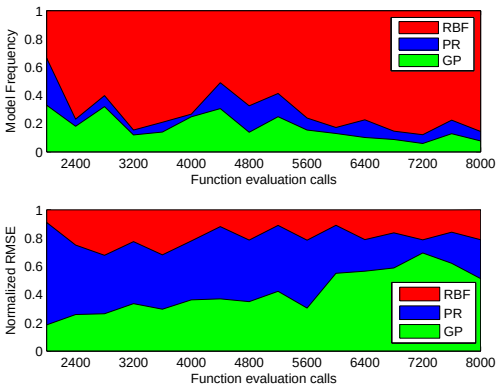
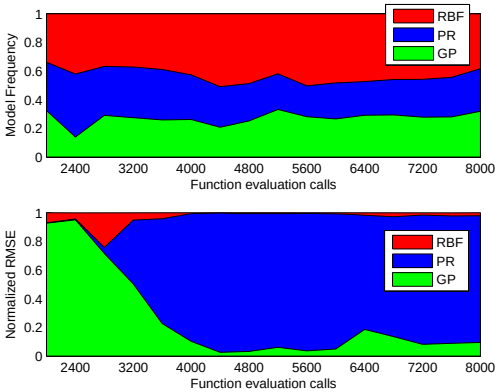


Figure 6.4: **Fitness prediction error** of surrogate models on benchmark problems.

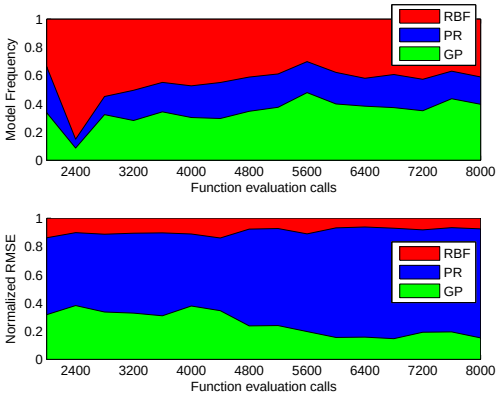
results from Fig. 6.5 suggest that no single approximation methodology has served as most suitable throughout the different stages of the search. For instance, RBF is noted to be used more prominently than the other approximation methodologies at the initial stages of the search but then exhibits a decreasing trend in frequency of usage at the later stage of the search on both the Rosenbrock and rotated shifted Rastrigin (see Figs. 6.5.c, 6.5.d). Likewise, the fitness prediction qualities of GP is noted to be significantly low at the initial stage of the search on Griewank (see Fig. 6.5.b) but the error eases as the search evolves. Note that such variations in fitness prediction qualities of the PR model can also be observed on Ackley (see Fig. 6.5.a). These results in a way strengthens our aspiration towards the notion of evolvability and hence the significance of EvoLS in facilitating productive cooperation among diverse approximation methodologies, working together to accomplish the shared goal of effective and efficient global optimization in the evolutionary search.



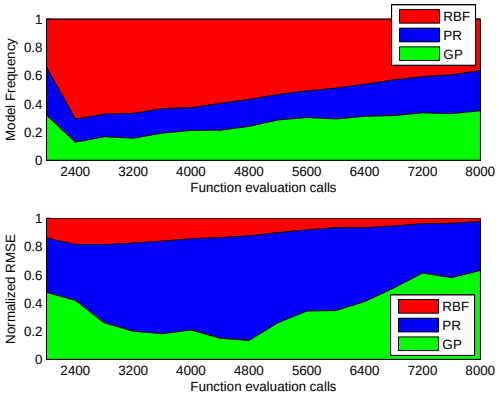
6.5.a: **Ackley-30D**



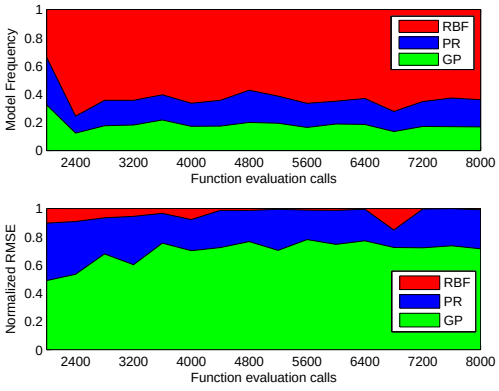
6.5.b: **Griewank-30D**



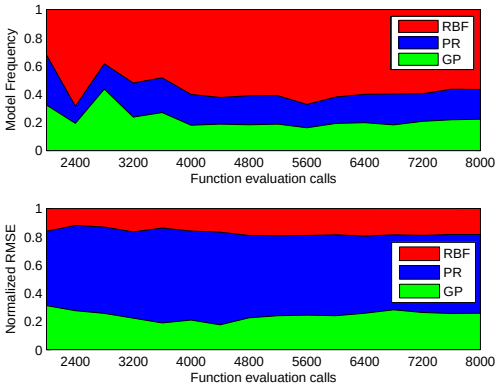
6.5.c: **Rosenbrock-30D**



6.5.d: **Rastrigin-30D-RS**



6.5.e: **Weierstrass-30D-RS**



6.5.f: **Grie+Rosen-30D**

Figure 6.5: Surrogates in EvoLS

6.3 Conclusions

In this chapter, a novel *Evolvability Learning* of Surrogates (EvoLS) framework that operates on *multiple* approximation methodologies of diverse characteristics has been presented. The suitability of an approximation method in the construction of surrogates for guiding the search is assessed by the *evolvability* metric instead of solely focusing on the fitness prediction error. By constructing the respective surrogate using the most productive approximation method inferred for each individual solution in the population, EvoLS serves as a self-configurable surrogate-assisted memetic algorithm for optimizing computationally expensive problems at improved search performance.

Numerical study of EvoLS with assessment made against the use of either single approximation methodology or the imaginary perfect surrogates as well as other state-of-the-art algorithms on representative benchmark problems demonstrated the robustness and efficiency of EvoLS. Further analysis on the suitability of surrogates operating within EvoLS also confirmed the motivations to introduce the evolvability measure that takes into account the current state of the search, properties of the search operators and characteristics of the fitness landscape, while the search progress online. EvoLS thus serves as an initial effort to design a framework that promotes competition and cooperation among diverse approximation methodologies, working together to accomplish the shared goal of global optimization in the evolutionary search.

Chapter 7

Real-World Applications

Chapter 7 presents two case studies of the proposed self-configurable memetic framework in solving real-world optimization problems. First, Section 7.1 illustrates Symbiotic Evolution for *optimizing the OSS2 potential model* that describes $(H_2O)_n$ water clusters when no prior knowledge of suitable search profile on the problem of interest is available. Section 7.2 then showcases the real world application of EvoLS on *an aerodynamic car rear design* that involves highly computationally expensive CFD simulations. Finally, Section 7.3 concludes the present study with a brief discussion.

7.1 OSS2 Potential Modelling for $(H_2O)_n$ Water Clusters

Water clusters play a fundamental role for understanding the enigmatic properties of water. In science, water clusters are heavily studied to characterize the basic molecular interactions, elucidate water's role in biochemical processes and the hydrophobic/ hydrophilic interactions [204]. As opposed to conducting experimental study which is extremely difficult, some scientists have turned to computer simulation studies on the structures, properties and spectra of water, including the potential energy and electrostatic properties. Among the models currently in use are first-principles (*ab initio*) quantum chemical computations and semi-empirical methods. The former possess the benefits of high fidelity but can be extremely computationally

expensive, limiting their use to simulations involving only a small numbers of atoms. Advanced empirical water models, which are fitted to experimental data, on the other hand, are computationally more efficient but give a trade-off in terms of model fidelity or accuracy.

Today, the design of computationally cheap advanced empirical water models remains an important and unsolved problem. Here, the parametric design of a potential model, known as OSS2, for describing neutral water clusters $(H_2O)_n$ [135, 128, 179] by means of symbiotic evolution is considered. The benefits of OSS2 potential model include dissociability, polarizability and flexibility, allowing the simulation of different forms of water clusters. It is worth noting that the potential model was originally designed to describe protonated water clusters, $H^+(H_2O)_n$, by taking into consideration the interactions between H^+ and O^{2-} ions, with additional three-body H-O-H interactions terms and self-consistent treatment of the polarizability of the oxygen ions, as described in [135]. In order to adopt the OSS2 potential model for neutral water clusters, $(H_2O)_n$, 40 variables of the model are generated by fitting to the high level *ab initio* MP2/cc-pVTZ potential energy surface of the H_2O and $(H_2O)_2$ clusters. The objective is then to minimize the root mean square differences between the cluster potential energies of OSS2 model and those based on high-fidelity *ab initio* MP2 calculations, i.e., $(E^k_{OSS2}(\mathbf{p}) - E^k_{MP2})$, which takes the form of

$$f_{RMS}(\mathbf{p}) = \sqrt{\frac{1}{M}\sum_{k=1}^{M}(E^k_{OSS2}(\mathbf{p}) - E^k_{MP2})^2} \qquad \text{(Eq. 7.1)}$$

where $\mathbf{p}$ is the candidate set of parameters, *M* is total number of cluster configurations in the fitting data, $E^k_{OSS2}(\mathbf{p})$ and E^k_{MP2} are the energies of water cluster configuration k obtained by OSS2 and *ab initio* MP2 calculations, respectively.

The parametric design of a potential model OSS2, described by Eq. 7.1, has been known in molecular chemistry to possess a multimodal fitness landscape [135]. As such, a conventional gradient descent algorithm may easily get stuck in a local optimum. Here, symbiotic evolution (SE) and three other canonical memetic algorithms using a gradient-based local search

DFP (MA-DFP), direct search DSCG (MA-DSCG) and stochastic local search (MA-ES) are considered on the problem. Under the computational budget of $300,000$ evaluations, the resultant optimized f_{RMS} on the 40-dimensional problem from the best run (out of five runs) for each algorithm are MA-DFP (0.298 kJ.mol^{-1}), MA-DSCG (0.436 kJ.mol^{-1}) and MA-ES (2.36 kJ.mol^{-1}). On this potential model design problem, the worst and best canonical MA with single search profile were found to be MA-ES and MA-DFP, respectively. Symbiotic evolution with six search profiles (as described in Section 5.2.3), on the other hand, generated a low numerical error of **0.222** kJ.mol^{-1}, which outperforms the best canonical MA on this real world problem. It also worth noting that to the best of our knowledge, the f_{RMS} solution attained by SE is much lower than that of the original OSS2 parameters reported in the literature [135], which stands at 2.58 kJ.mol^{-1}.

Aside from the ability of the OSS2 potential model in reproducing *ab initio* MP2 energy calculation, it is also desirable for the potential model to be capable of accurately describing the stable geometry structures of $(H_2O)_n$ clusters. Figure 7.1 illustrates the geometry structures of water clusters [$(H_2O)_n$, *n*=1-3] in which red balls represent the oxygen atoms whereas the yellow ones represent hydrogen atoms. Due to the symmetry of the clusters' structures, only representative bond information of the distances (D), angles (A) and torsion angles (T) between atoms are shown in the figure.

The stable geometry structures of water clusters with lowest potential energy are first obtained by conducting a structural optimization procedure on the OSS2 potential energy surface with the original and optimized parameters obtained by the algorithms. The bonding information of the optimized [$(H_2O)_n$, *n*=1-3] clusters is then assessed by taking the *ab initio* MP2 calculations as baseline for investigating the quality of the freshly designed OSS2 potential model. The numerical results as reported in Table 7.2 include, from left to right, the MP2 bonding information and those calculated using the original OSS2 model [135] and optimized OSS2 models obtained by the MAs and symbiotic evolution (SE). Note that the *ab initio* MP2

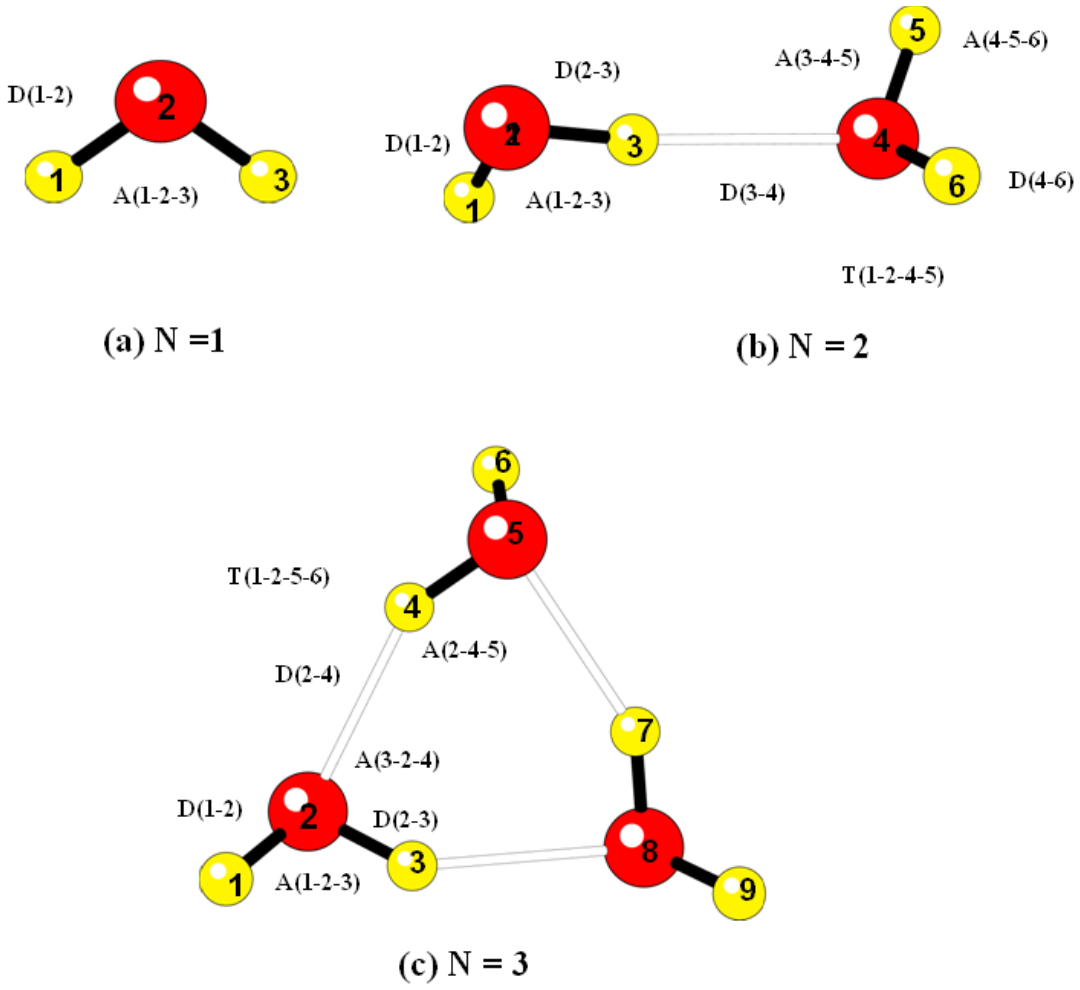


Figure 7.1: Structures of water clusters [$(H_2O)_n$, n=1-3].

bonding information is considered as the reference geometric structures for comparison. The bond information errors of the optimized structures with respect to the *ab initio* MP2 calculations are normalized and reported in Table 7.1. Summation of the normalized errors for each algorithm on the $(H_2O)_n$ clusters is subsequently used as an error index for comparison. From the table, SE exhibits the lowest error index, thus indicating that the potential model derived by symbiotic evolution reproduces well not only the *ab initio* potential energies but also the geometrical structures of water clusters that agree favorably with those obtained by means of *ab initio* calculations. Such results confirmed our motivation for formalizing the symbiosis of gene and meme in adaptive search, thus encouraging the use of symbiotic evolution model, rather than relying simply on one fixed, and possibly poor choice of search profile that may not lead to truly optimized designs. Symbiotic evolution clearly offers a high quality and robust approach for continuous parametric design problems, regardless of whether *a priori* knowledge of the most suitable search profile on the problem is available.

7.2 Aerodynamic Optimization of the Rear of a Car Model

Our ultimate motivation of the present work lies in the difficulties and challenges posed by computationally expensive real-world applications. In this section, the proposed EvoLS is considered for the design of a quasi-realistic aerodynamic car rear using a simplified model version of the Honda Civic. The objective is to minimize the aerodynamic performance calculation of the design, i.e., the total drag of the car rear. The design model of the Honda Civic used in the present study is labelled here as *Baseline-C-Model*. Aerodynamic car rear design is an extremely complex task that is normally undertaken over an extended time period and at different levels of complexity. In this application, an aerodynamic performance calculation of the design requires a Computational Fluid Dynamics (CFD) simulation that generally takes around 60 minutes wall-clock time on a Quad-Core machine. For the calculation, the *Open-*

Table 7.1: Normalized errors on the optimized structures of water clusters with respect to the *ab initio* MP2 calculations. From left to right: normalized errors from the original OSS2 model [135] and optimized OSS2 models obtained by the MAs and symbiotic evolution (SE).

$[(H_2O)_n]$ Clusters	**Original OSS2 Model**	**MA-DFP**	**MA-DSCG**	**MA-ES**	**SE**
n=1					
D(1-2)	0.33	0.04	0.25	0.33	0.04
A(1-2-3)	0.47	0.00	0.16	0.37	0.00
n=2					
D(1-2)	0.24	0.31	0.20	0.20	0.04
D(2-3)	0.03	0.03	0.81	0.03	0.10
A(1-2-3)	0.15	0.19	0.49	0.11	0.06
D(3-4)	0.02	0.18	0.72	0.02	0.06
A(3-4-5)	0.13	0.29	0.32	0.13	0.14
D(4-6)	0.04	0.20	0.40	0.00	0.36
A(4-5-6)	0.44	0.20	0.00	0.32	0.04
T(1-2-4-5)	0.03	0.92	0.01	0.03	0.02
n=3					
D(1-2)	0.23	0.14	0.45	0.09	0.09
D(2-3)	0.07	0.12	0.57	0.07	0.17
A(1-2-3)	0.15	0.03	0.69	0.10	0.02
A(3-2-4)	0.07	0.11	0.62	0.07	0.13
D(2-4)	0.03	0.10	0.71	0.02	0.14
A(2-4-5)	0.13	0.22	0.38	0.09	0.18
T(1-2-5-6)	0.06	0.01	0.54	0.38	0.00
Total	2.65	3.06	7.34	2.38	**1.58**

Table 7.2: Optimized structures of water clusters. From left to right: the *ab initio* MP2 bonding information and those calculated using the original OSS2 model [135] and optimized OSS2 models obtained by the MAs and symbiotic evolution (SE). Note that the *ab initio* MP2 bonding information is considered as the reference geometric structures for comparison.

$[(H_2O)_n]$ Clusters	**MP2 bonding information**	**Original OSS2 Model**	**MA-DFP**	**MA-DSCG**	**MA-ES**	**SE**	**Unit**
n=1							
D(1-2)	0.966	0.958	0.965	0.972	0.958	0.967	Å
A(1-2-3)	103.9	109.2	103.9	102.1	108	103.9	°
n=2							
D(1-2)	0.965	0.953	0.98	0.955	0.955	0.963	Å
D(2-3)	0.973	0.982	0.98	1.194	0.982	1.000	Å
A(1-2-3)	104.3	109.7	97.8	87.1	108.1	102.3	°
D(3-4)	1.952	1.932	2.109	1.309	1.935	1.899	Å
A(3-4-5)	104.2	113.7	126.2	80.2	114.3	114.6	°
D(4-6)	0.967	0.966	0.972	0.957	0.967	0.976	Å
A(4-5-6)	104.2	109.2	101.9	104.2	107.9	104.6	°
T(1-2-4-5)	123.3	116.7	-90	124.9	116.8	118.8	°
n=3							
D(1-2)	0.965	0.96	0.968	0.975	0.963	0.967	Å
D(2-3)	0.979	1.008	1.027	1.209	1.007	1.046	Å
A(1-2-3)	105.2	111.8	104.1	75.7	109.6	106.2	°
A(3-2-4)	89.3	90.0	90.4	83.0	90.0	88.0	°
D(2-4)	1.925	1.893	1.836	1.264	1.906	1.798	Å
A(2-4-5)	148.4	146.2	144.7	154.9	150	151.4	°
T(1-2-5-6)	-21.2	-2.8	-24.5	-176	-129.5	-22.2	°

FOAM CFD flow solver [71, 72] used allows a tight integration into the optimization process, providing an automatic meshing procedure as well as parallelization.

The choice of an adequate geometrical representation of the car simulation model that can be reasonably coupled to the optimization algorithm is also crucial. In the presented experiments, the state-of-the-art *Free Form Deformation* (FFD) [169, 28] has been chosen as the geometrical representations since it provides a fair trade-off between design flexibility and scalable number of optimization parameters. FFD is a shape morphing technique that allows smooth design changes using an arbitrary number of parameters which are intuitively adjustable to the problem at hand. The benefits of FFD can be found in [116].

For the technical realization, the application of FFD requires a control volume, i.e. a lattice of control points, in which the model geometry is embedded. In the next step the geometry is transferred to the spline parameter space of the control volume, a numerical process which is called "freezing". After the object is frozen it is possible to select and move single or several control points to generate deformed shape variations. To amplify the surface deformations, it is important to position the control points close to the car body. Fig. 7.2 illustrates the *Baseline-C-Model* as well as the implemented FFD control volume. Based on this set-up, ten optimization parameters p_i have been chosen. Each parameter comprises a group of 22 control points within one layer of the control volume as marked by the dashed box in the lower left image of Fig. 7.2. Since the model is symmetric to the center plane in y-direction, the parameters affect the left and right side of the car in the same way.

During the evaluation step of the optimization process, the aerodynamic performance of each design solution, i.e. the total drag of the car, is calculated. Therefore, each of the ten parameters stored in the design solution or individual are extracted and added as an offset on the corresponding layer of control points, either in x-direction $(p_1, p_3, p_5, p_7, p_9)$ or in z-direction $(p_2, p_4, p_6, p_8, p_{10})$. Second, based on the modified control points, the car shape is updated using FFD and stored as a triangulated mesh, i.e. STL file format. Third, the external air flow

around the car is computed by the *OpenFOAM* CFD solver. Therefore, a CFD calculation grid around the updated car shape has to be generated and is automatically carried out using the *snappyHexMesh* tool of *OpenFOAM*. Based on this grid, the external air flow is calculated, resulting in a drag value that is monitored every 10^{th} time step of the simulation. After the calculation has sufficiently converged, the total drag value is extracted and assigned to the individual.

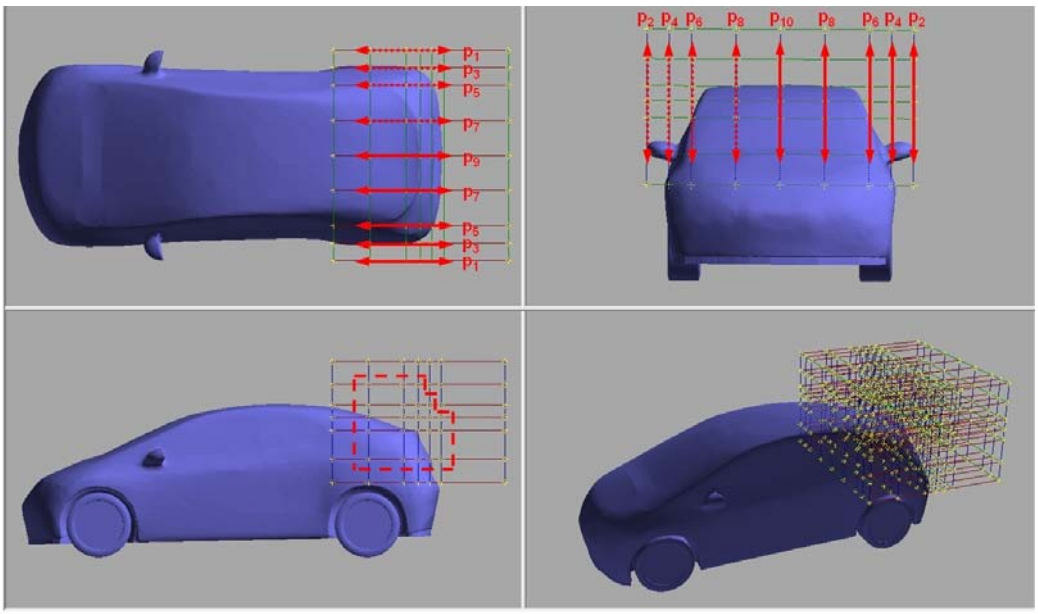


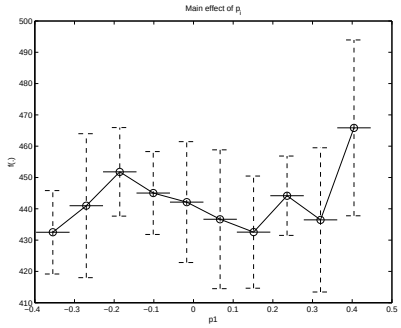
Figure 7.2: Experiment Set-up

7.2.1 Fitness Landscape Analysis

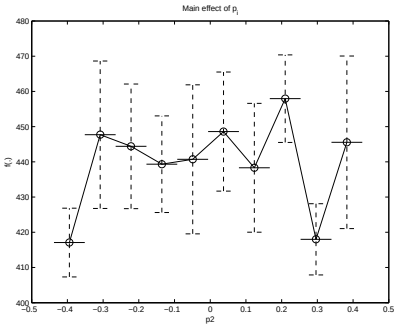
To conduct a preliminary analysis on the conditions of the problem fitness landscape, the effect of design variables p_i on drag at the car rear is investigated in this section. In particular, the design solutions generated randomly using the Latin hypercube sampling technique and the corresponding aerodynamic performance (i.e., total drag) from the SimpleBase-C-Model landscape are first grouped into different bins based on each control variable p_i. The average fitness and standard deviation of each bin is plotted against the control variable for the sensi-

tivity analysis. Here the plots of 700 samples drawn from the search range of $[-0.4, 0.4]^{10}$ for each dimension p_i which is divided into 10 bins are presented in Fig. 7.3.

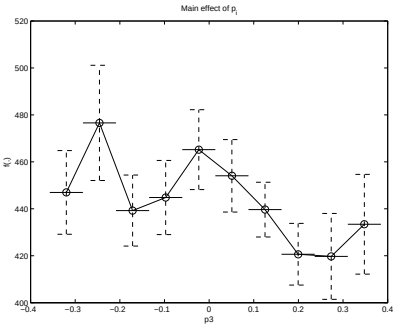
The sensitivity analysis of the design variables presented in Fig. 7.3 depicts the relationships between each input variable and the objective function. In this case, the non-monotonic sensitivity of the problem landscape with respect to each design variable was observed, especially in Figs. 7.3.a, 7.3.b, 7.3.c, 7.3.f, 7.3.h and 7.3.j. The dependencies of the objective function to the interactions of multiple design variables as implied by the non-monotonic trends in Fig. 7.3 thus highlighted the landscape ruggedness of the quasi-realistic aerodynamic car rear design problem.



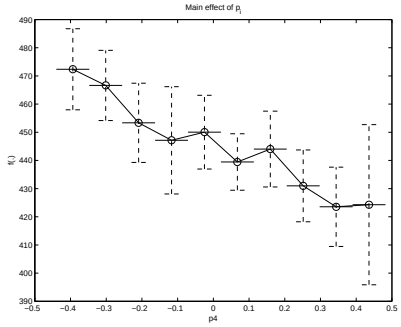
7.3.a: **Main effect of** p_1



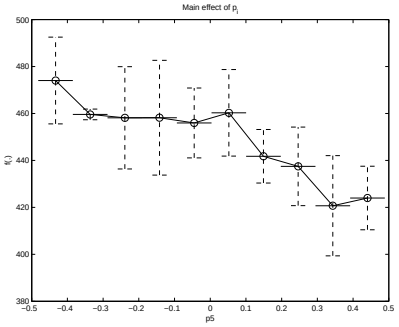
7.3.b: **Main effect of** p_2



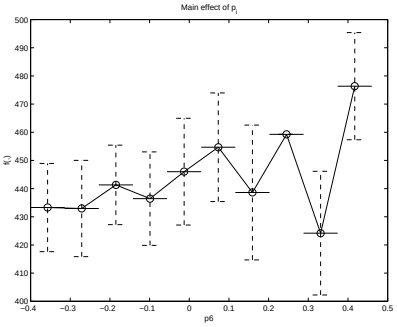
7.3.c: **Main effect of** p_3



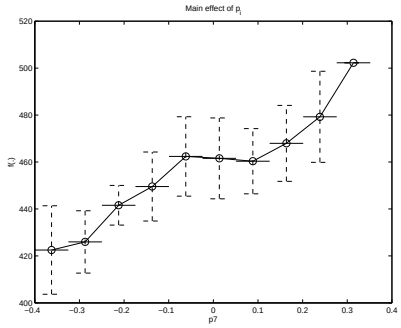
7.3.d: **Main effect of** p_4



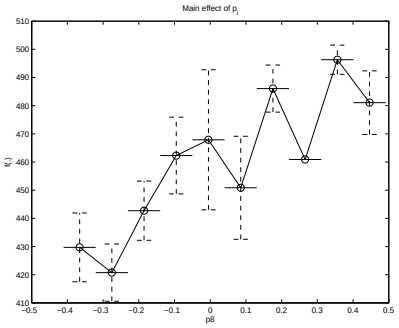
7.3.e: **Main effect of** p_5



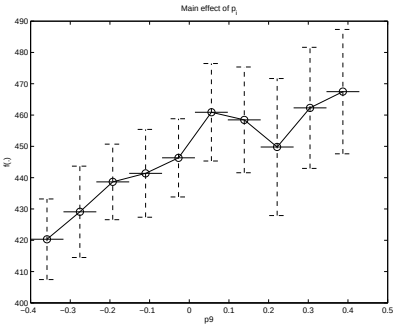
7.3.f: **Main effect of** p_6



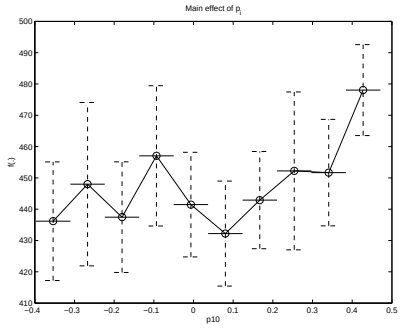
7.3.g: **Main effect of** p_7



7.3.h: **Main effect of** p_8



7.3.i: **Main effect of** p_9



7.3.j: **Main effect of** p_{10}

Figure 7.3: Landscape Analysis of the Aerodynamic Car Rear Design Problem

7.2.2 Optimization Results

Due to the highly computationally expensive CFD simulations, a computational budget of 200 exact evaluations (200 hours) is used for one optimization run in our study. A small population size of 10 was considered in EvoLS. Note that no database building phase (i.e., $G_{db} = 0$) is required by EvoLS in this case since sufficient sample data of the search space was available at hand. The convergence trend of the best run (out of five runs) for the aerodynamic car rear design problem obtained by EvoLS (as described in Section 6.2) after 200 evaluations on the aerodynamic car rear design problem is shown in Fig. 7.4. The optimization results obtained previously based on the Covariance Matrix Adaptation Evolution Strategy [51], including CMA-ES(5, 10) and CMA-ES(1, 10), are also reported in the figure for comparison. As shown in the figure, due to the complexity of the problem landscape, CMA-ES(1, 10) as an individual-based search strategy performed worst as compared to the other population-based approaches. Among the algorithms considered, EvoLS exhibited the best performances by locating the car rear design with the lowest drag value of $\mathbf{403.573}$. The search trends also show that the proposed EvoLS arrived at the best design solution discovered by CMA-ES(5, 10) using only $1/2$ of the computational budget incurred by the latter.

Subsequently, the actual designs of the car model constructed from the control volume (i.e., 22 control points) optimized using EvoLS and CMA-ES(5,10) are presented in Figure 7.5. The grid in *red* color reflects the change from the modified control points, compared to the shape of the default design in *white* color. From Figures 7.5.c-7.5.d, it is suggested that the better aerodynamic performance of the design from EvoLS is achieved with a higher curvature and extended car rear as compared to that obtained by CMA-ES(5,10) in Figures 7.5.a-7.5.b. Thus, the design principle to reduce the total drag force drawn from the optimization results comes with a trade-off in higher lift force at the car rear.

In conclusion, the computational saving of more than 50% and improved solution quality attained thus show that in a practical setting, the usage of the proposed EvoLS can be highly

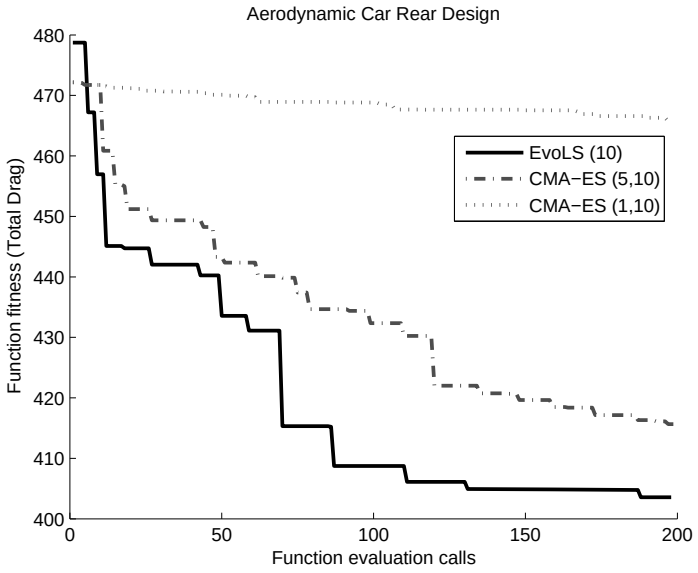
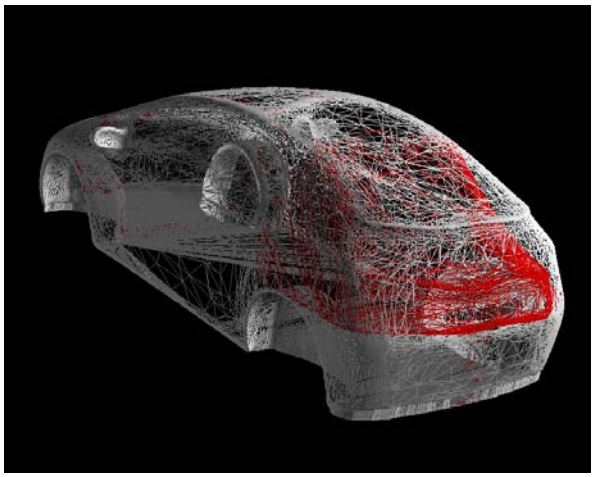
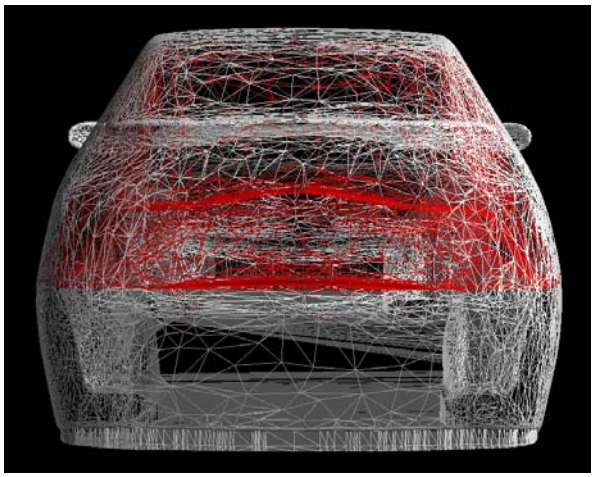


Figure 7.4: Convergence Trace on Aerodynamic Car Rear Design

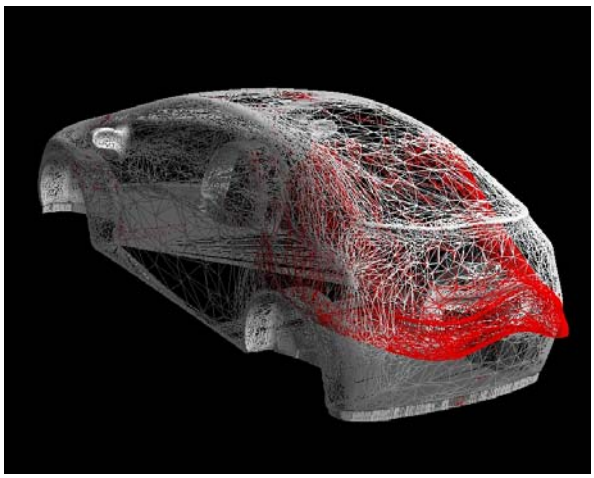
beneficial as compared to standard techniques for solving challenging computationally expensive design optimization.



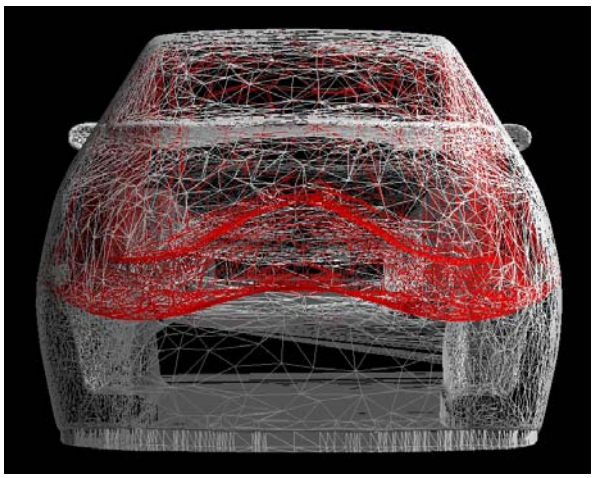
7.5.a: **Design from CMA-ES(5, 10) - Side view**



7.5.b: **Design from CMA-ES(5, 10) - Rear view**



7.5.c: **Design from EvoLS - Side view**



7.5.d: **Design from EvoLS - Rear view**

Figure 7.5: Actual design models from the optimized parameters.

7.3 Conclusions

The general practice on manual crafting through trials of dedicated search solvers in computational intelligence community has evolved into the self-configurable memetic framework proposed and presented in this thesis. Two main difficulties that are often encountered when solving real-world optimization problems have been identified as: 1) a lack of prior knowledge for suitable search configurations or search landscape and 2) a burden of computationally expensive simulation to evaluate candidate (design) solutions. By formalizing the symbiosis of search mechanisms in memetic search and further incorporating approximation models, the present work arrives at a practical self-configurable memetic search paradigm that can handle these difficulties effectively and elegantly. The case studies presented in this chapter have highlighted the competitiveness of the proposed self-configurable memetic framework in attaining improved search performance on complex real-world applications under limited computational budget.

Chapter 8

Concluding Remarks

This chapter concludes the work done so far in the research topic of "Self-configurable Memetic Algorithm" with a brief summary of the core contributions and outlines several potential areas of future research.

8.1 Research Contributions

The focus of this research is to develop intelligent and practical self-configurable memetic framework that leverages from hybridizing available high performance population-based and individual-based search algorithms to achieve new advancements in solving complex optimization problems. By encouraging productive symbiosis among unique search components and incorporating fitness-improving approximation models, the proposed framework has effectively handled two common difficulties found in solving real-world complex optimization problems, identified as: 1) a lack of prior knowledge for suitable search configurations or search landscape and 2) a burden of computationally expensive simulation to evaluate candidate (design) solutions. This was achieved by the effective use of statistical learning techniques on optimization data archived online while the search progresses.

The primary contribution of this dissertation work is summarized as follows:

- The core contributions of the present research include a comprehensive survey on existing conventional mathematical programming and evolutionary algorithms for single-

objective continuous optimization in Chapter 2. The survey, which establishes a background for the reader before proceeding to the main highlights of the thesis in the subsequent chapters, serves to be a helpful reference for researchers or practitioners on related works and technology.

- For the understanding of MA search mechanisms, the influence of individual learning on the selection pressure is first analyzed in Chapter 3 to illustrate how memetic algorithm benefits from individual learning in advancing the search towards the global optimum, due to the presence of *constructive* local optimum structure. Subsequently, the symbiosis of stochastic variation and individual learning operators in MA search, labelled in this thesis as symbiosis search profile, is formalized in the form of the local optimum connectivity in Chapter 4. To quantify the suitability of search components in creating viable, or potentially favorable solutions, the *evolvability* of symbiotic search profile is introduced as the basis for assessment and subsequently used as the metric for adaptation in the present work. Using the proposed concepts, experimental study and analyses on benchmark problems are then performed to provide insights into the success of MAs reported recently in the literature.

- To support the online self-configuration of productive search on complex continuous optimization problems, a novel statistically learning scheme on the *evolvability* of search profiles using historical data archived along the search is introduced in Chapter 5. From here, the main contribution of the current research is the proposal of a self-configurable memetic framework, established as Symbiotic Evolution, that facilitates the emergence of productive search profiles from multiple stochastic variation and individual learning strategies, based on their inferred *evolvability* measure.

- To deal with real-world complex optimization problems plagued with computationally expensive fitness functions that would take many hours of wall clock time (per evaluation) to compute, the proposed memetic framework is thus extended in Chapter 6 to

incorporate approximation models that mitigate these forms of problems elegantly. In particular, the notion of *evolvability* is further studied as an effective metric to assess the appropriateness of approximation models, known also as surrogates, in bringing about search improvement efficiently. This has led to the proposal of a novel *Evolvability Learning* of Surrogates framework that operates autonomously on multiple *fitness improving* surrogates working in synergy to enhance memetic search.

- Last but not least, empirical analyses on representative benchmark problems and case studies on real-world problems including the optimization of *OSS2 potential energy model* and *aerodynamic car rear design* indicate noteworthy results in our effort to provide a formal modeling on the combined behaviors of search components, leading to an unified self-configurable approach to many population-based and numerical optimization techniques available.

8.2 Future Work

The methodologies proposed in this dissertation have provided some of the groundwork towards an intelligent self-configurable optimization system for solving complex optimization problems, by leveraging from available high performance population-based and individual-based search algorithms and the effective use of statistical learning techniques on optimization data archived online while the search progresses. Future research will continue to focus on the challenges posed by optimization problems in real-world scenarios and further extends the practicality of the self-configurable framework. A summary of several potential future works is outlined below.

8.2.1 Generalization of Variation Operators

Future research to demonstrate the generality of the framework on other advanced search profiles is warranted to promote the proposed self-configurable memetic framework to practitioner

and researchers. The aim is to provide users a possibility to integrate the solvers in their specified domain to the self-configurable framework straightforwardly to achieve advancement in search. The main challenges lie in developing the density distribution functions of the search operators while addressing the complexity of the framework in spite of the increasing number of search components.

8.2.2 Multi-objective/Constrained Optimization

Optimization problems in real-world applications often come in more complex scenario than single-objective optimization. For example, a successful development of car designs with multiple interrelating disciplines or scenarios requires advanced computational methods for the complexity of engineering tasks in the automotive area. Increasing computational power as well as simulation capabilities allows the evaluation and optimization of a manifold of design proposals in various environmental scenarios, such as aerodynamic optimization, structure optimization or noise, vibration and harshness (NVH) optimization. Even within a single scenario, usually several objectives have to be treated simultaneously, such as drag reduction, design constraints, internal car air flow, etc., in the aerodynamic scenario. Therefore, it is important and yet challenging to place the future research efforts on the topics of multi-objective and/or constraints optimization problems [113, 222, 200], in which the notion of *evolvability* could be extended to encapsulate the goals of the self-configurable solver.

8.2.3 Combinatorial Optimization

Last but not least, combinatorial optimizations which exist in many real-world optimization problems and application, such as job scheduling [69, 58] or large scale integrated-circuit (VLSI) design automation [189], pose as another area of notable future search for self-configurable memetic framework. Here, the main challenge would be on modelling the probability functions for various stochastic variations and individual learning procedures on discrete variables.

References

[1] E. Acar and M. Rais-Rohani. Ensemble of metamodels with optimized weight factors. *Structural and Multidisciplinary Optimization*, 37(3):279–294, 2009.

[2] U. Aickelin, E. K. Burke, and J. Li. An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering. *Journal of the Operational Research Society*, 58(12):1574–1585, 2006.

[3] N. M. Alexandrov, J. E. Dennis, R. M. Lewis, and V. Torczon. A Trust-region Framework for Managing The use of Approximation Models in Optimization. *Structural and Multidisciplinary Optimization*, 15(1):16–23, 1998.

[4] L. Altenberg. The Schema Theorem and Prices Theorem. *Foundations of Genetic Algorithms*, 3:23–49, 1995.

[5] A. Auger and N. Hansen. A Restart CMA evolution strategy with increasing population size. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, 2:1769–1776, 2005.

[6] A. Auger and N. Hansen. A Restart CMA evolution strategy with increasing population size. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, 2:1769–1776, 2005.

[7] A. Auger and N. Hansen. Performance Evaluation of an Advanced Local Search Evolutionary Algorithm. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, 2:1777–1784, 2005.

[8] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. Taylor & Francis, 1997.

[9] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms*, pages 14–21, 1987.

[10] P. J. Ballester, J. Stephenson, J. Carter, and K. Gallagher. Real-Parameter Optimization Performance Study on the CEC-2005 Benchmark with SPC-PNX. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 498–505, 2005.

[11] S. Baluja. Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical report, Pittsburgh, PA, USA, 1994.

[12] H. G. Beyer. Toward a Theory of Evolution Strategies: Some Asymptotical Results from the (1,+ λ)-Theory. *Evolutionary Computation*, 1(2):165–188, 1993.

[13] H. G. Beyer. *The Theory of Evolution Strategies*. Springer, 2001.

[14] H.-G. Beyer and H.-P. Schwefel. Evolution strategies–A comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002.

[15] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1996.

[16] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization*, 17(1):1–13, 1999.

[17] E. Borenstein, I. Meilijson, and E. Ruppin. The effect of phenotypic plasticity on evolution in multipeaked fitness landscapes. *Journal of Evolutionary Biology*, 19(5):1555–1570, 2006.

[18] J. Branke and C. Schmidt. Faster convergence by means of fitness estimation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1):13–20, 2005.

[19] A. Brindle. *Genetic algorithms for function optimization*. (doctoral dissertation and technical report tr81-2), Edmonton: University of Alberta, Department of Computer Science, 1981.

[20] D. Buche, N. N. Schraudolph, and P. Koumoutsakos. Accelerating evolutionary algorithms with Gaussian process fitness function models. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 35(2):183–194, 2005.

[21] B. D. Bunday. *Basic Optimization Methods*. Edward Arnold (Publishers) Ltd, 1985.

[22] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner. A Fast Adaptive Memetic Algorithm for Online and Offline Control Design of PMSM Drives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1):28–41, 2007.

[23] U. K. Chakraborty. *Advances in Differential Evolution*. Springer Verlag, 2008.

[24] R. Chelouah and P. Siarry. Tabu search applied to global optimization. *European Journal of Operational Research*, 123(2):256–270, 2000.

[25] R. Chelouah and P. Siarry. A hybrid method combining continuous tabu search and Nelder-Mead simplex algorithms for the global optimization of multiminima functions. *European Journal of Operational Research*, 161(3):636–654, 2005.

[26] T. Chen, K. Tang, G. Chen, and X. Yao. On the analysis of average time complexity of estimation of distribution algorithms. *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pages 453–460, 2007.

[27] D. C. Chin. Comparative study of stochastic algorithms for system optimization based on gradient approximations. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 27(2):244–249, 1997.

[28] S. Coquillart. Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling. *Computer Graphics*, 24(4):187–196, 1990.

[29] A. Corana, M. Marchesi, C. Martini, and S. Ridella. Minimizing multimodal functions of continuous variables with the simulated annealing algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 13(3):262–280, 1987.

[30] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[31] P. Cowling, G. Kendall, and E. Soubeiga. A hyperheuristic approach to scheduling a sales summit. pages 176–190. Springer, 2001.

[32] L. DaCosta, A. Fialho, M. Schoenauer, and M. Sebag. Adaptive operator selection with dynamic multi-armed bandits. *Proceedings of the 10th annual Conference on Genetic and Evolutionary Computation*, pages 913–920, 2008.

[33] W. Daneshyari and G. G. Yen. Cultural MOPSO: A cultural framework to adapt parameters of multiobjective particle swarm optimization. *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pages 1325–1332, 2008.

[34] L. Davis. Bit Climbing, Representational Bias, and Test Suite Design. *Fourth International Conference on Genetic Algorithm*, pages 18–23, 1991.

[35] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold Company, 1991.

[36] K. De Jong. *Evolutionary Computation: A Unified Approach*. MIT Press, 2006.

[37] D. L. Donoho. High-dimensional data analysis: the curses and blessings of dimensionality. *American Mathematical Society Conference on Math Challenges of the 21st Century*, August 2000.

[38] A. E. Douglas. *Symbiotic Interactions*. Oxford University Press, USA, 1994.

[39] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[40] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS '95.*, pages 39–43, 1995.

[41] R. Eberhart, P. Simpson, and R. Dobbins. *Computational intelligence PC tools*. Academic Press Professional, Inc., San Diego, CA, USA, 1996.

[42] M. Emmerich, A. Giotis, M. Özdemir, T. Bäck, and K. Giannakoglou. Metamodel-Assisted Evolution Strategies. *Parallel Problem Solving from NaturePPSN VII*, 2439:361–370, 2002.

[43] A. H. Fielding and J. F. Bell. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, 24(01):38–49, 1997.

[44] R. Fletcher. A New Approach to Variable Metric Algorithm. *Computer Journal*, 7(4):303–307, 1964.

[45] C. A. Floudas. *Deterministic Global Optimization: Theory, Methods, and Applications*. Springer-Verlag New York, 2005.

[46] C. C. García-Martínez and M. Lozano. Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 896–903, 2005.

[47] K. C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1):43–76, 2002.

[48] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.

[49] D. E. Goldberg. Probability Matching, the Magnitude of Reinforcement, and Classifier System Bidding. *Machine Learning*, 5(4):407–425, 1990.

[50] D. E. Goldberg and K. Deb. A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. *Foundations of Genetic Algorithms*, 1:69–93, 1991.

[51] N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 282–291. Springer, 2004.

[52] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.

[53] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[54] G. H. Hardy, J. E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 2nd edition, 1988.

[55] W. E. Hart. *Adaptive Global Optimization with Local Search*. PhD thesis, University of California, San Diego, 1994.

[56] W. E. Hart, N. Krasnogor, and J. E. Smith. Editorial Introduction, Special Issue on Memetic Algorithms. *Evolutionary Computation*, 12(3):v–vi, 2004.

[57] W. E. Hart, N. Krasnogor, and J. E. Smith. *Recent Advances in Memetic Algorithms*, volume 166. Springer, 2004.

[58] S. M. K. Hasan, R. Sarker, D. Essam, and D. Cornforth. Memetic algorithms for solving job-shop scheduling problems. *Memetic Computing*, 1(1):69–83, 2009.

[59] F. Herrera, M. Lozano, and A. M. Sánchez. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems*, 18(3):309–338, 2003.

[60] F. Herrera, M. Lozano, and J. L. Verdegay. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. *Artificial Intelligence Review*, 12(4):265–319, 1998.

[61] R. Hinterding, Z. Michalewicz, and A. E. Eiben. Adaptation in evolutionary computation: A survey. In *Proceedings of the Fourth International Conference on Evolutionary Computation (ICEC 97)*, pages 65–69, 1997.

[62] G. E. Hinton and S. J. Nowlan. How Learning Can Guide Evolution. *Complex Systems*, 1(1):495–502, 1987.

[63] C. Hirsch. *Numerical computation of internal and external flows: fundamentals of computational fluid dynamics*. Butterworth-Heinemann, 2nd edition, 2007.

[64] M. W. Ho. Why Lamarck won't go away. *Annals of Human Genetics*, 60(1):81–84, 1996.

[65] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press Cambridge, MA, USA, 1992.

[66] R. Hooke and T. A. Jeeves. "Direct Search" Solution of Numerical and Statistical Problems. *Journal of ACM*, 8(2):212–229, 1961.

[67] H. H. Hoos and T. Stützle. *Stochastic Local Search: Foundations and Applications*. Morgan Kaufmann, 2005.

[68] C. Houck, J. Joines, and M. Kay. Utilizing Lamarckian evolution and the Baldwin effect in hybrid genetic algorithms. Technical report, 1996.

[69] H. Ishibuchi, T. Yoshida, and T. Murata. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Transactions on Evolutionary Computation*, 7(2):204–223, 2003.

[70] E. Jablonka and M. J. Lamb. *Epigenetic Inheritance and Evolution: The Lamarckian Dimension*. Oxford University Press, 1995.

[71] H. Jasak, A. Jemcov, and Z. Tukovic. OpenFOAM: A c++ library for complex physics simulations. *International Workshop on Coupled Methods in Numerical Dynamics*, 2007.

[72] H. Jasak and H. Rusche. Dynamic Mesh Handling in OpenFOAM. *American Institute of Aeronautics and Astronautics Proceedings*, January 2009.

[73] J. Jin. *The Finite Element Method in Electromagnetics*. Wiley-IEEE Press, 2nd edition, 2002.

[74] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001.

[75] Y. Jin. A Comprehensive Survey of Fitness Approximation in Evolutionary Computation. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9(1):3–12, 2005.

[76] Y. Jin, M. Olhofer, and B. Sendhoff. On evolutionary optimization with approximate fitness functions. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Las Vegas, Nevada*, pages 786–793, 2000.

[77] Y. Jin, M. Olhofer, and B. Sendhoff. A Framework for Evolutionary Optimization with Approximate Fitness Functions. *IEEE Transactions on Evolutionary Computation*, 6(5):481–494, Oct 2002.

[78] Y. Jin and B. Sendhoff. Reducing fitness evaluations using clustering techniques and neural networks ensembles. *Genetic and Evolutionary Computation Conference (GECCO)*, LNCS 3102:688–699, 2004.

[79] T. Jones and S. Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms. *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 184–192, 1995.

[80] A. J. Keane. The OPTIONS Design Exploration System, 1995.

[81] A. J. Keane and N. Petruzzelli. Aircraft wing design using GA-based multi-level strategies. *AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, USA 06 - 08 Sep 2000.*, 2000.

[82] C. T. Kelley. *Solving Nonlinear Equations with Newton's Method*. in 'Fundamentals of Algorithms' series, SIAM Press., 2003.

[83] H.-S. Kim and S.-B. Cho. An efficient genetic algorithm with less fitness evaluation by clustering. *Proceedings of the 2001 Congress on Evolutionary Computation*, 2:887–894, 2001.

[84] R. Kohavi. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence (IJCAI)*, 14:1137–1145, 1995.

[85] N. Krasnogor. *Studies on the Theory and Design Space of Memetic Algorithms*. PhD thesis, Doctoral dissertation, University of the West of England, Bristol, England, 2002.

[86] N. Krasnogor, B. P. Blackburne, E. K. Burke, and J. D. Hirst. Multimeme algorithms for protein structure prediction. *Proceedings of the Parallel Problem Solving from Nature VII. Lecture Notes in Computer Science.*, 2439/2002:769–778, 2002.

[87] N. Krasnogor and J. Smith. A memetic algorithm with self-adaptive local search: TSP as a case study. *Proceeding of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 987–994, 2000.

[88] N. Krasnogor and J. Smith. Emergence of profitable search strategies based on a simple inheritance mechanism. *Proceedings of the 2001 Genetic and Evolutionary Computation Conference (GECCO)*, pages 432–439, 2001.

[89] N. Krasnogor and J. Smith. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Transactions on Evolutionary Computation*, 9(5):474–488, 2005.

[90] K. W. C. Ku, M. W. Mak, and W. C. Siu. A study of the Lamarckian evolution of recurrent neural networks. *IEEE Transactions on Evolutionary Computation*, 4(1):31–42, 2000.

[91] M. W. S. Land. *Evolutionary Algorithms with Local Search for Combinatorial Optimization*. PhD thesis, Citeseer, 1998.

[92] P. Larranaga. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, 2001.

[93] P. Larranaga, R. Etxeberria, J. A. Lozano, and J. M. Pena. Optimization in continuous domains by learning and simulation of Gaussian networks. pages 201–204, 2000.

[94] M. N. Le, Y. S. Ong, Y. Jin, and B. Sendhoff. Lamarckian memetic algorithms: local optimum and connectivity structure analysis. *Memetic Computing*, 1(3):175–190, 2009.

[95] M. N. Le, Y. S. Ong, Y. Jin, and B. Sendhoff. A Unified Framework for Symbiosis of Evolutionary Mechanisms with Application to Water Clusters Potential Model Design. *IEEE Computational Intelligence Magazine*, 7(1):20–35, 2012.

[96] M. N. Le, Y. S. Ong, and Q. H. Nguyen. Optinformatics for Schema Analysis of Binary Genetic Algorithms. *Proceedings of the 10th annual Genetic and Evolutionary Computation Conference (GECCO)*, (2):1121–1122, 2008.

[97] C.-Y. Lee and X. Yao. Evolutionary programming using mutations based on the Lévy probability distribution. *IEEE Transactions on Evolutionary Computation*, 8(1):1–13, 2004.

[98] W. F. Leong and G. G. Yen. Dynamic population size in PSO-based multiobjective optimization. *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1718–1725, 2006.

[99] W. F. Leong and G. G. Yen. PSO-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(5):1270–1293, 2008.

[100] F. H. Lesh. Multi-dimensional least-squares polynomial curve fitting. *Communications of ACM*, 2(9):29–30, 1959.

[101] Y.-W. Leung and Y. Wang. An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*, 5(1):41–53, 2 2001.

[102] R. Lewis and B. Paechter. Finding feasible timetables using group-based operators. *IEEE Transactions on Evolutionary Computation*, 11(3):397–413, 2007.

[103] R. M. Lewis, V. Torczon, and M. W. Trosset. Direct search methods: then and now. *Journal of Computational and Applied Mathematics*, 124(1-2):191–207, 2000.

[104] B. Li, Y. S. Ong, M. N. Le, and C. K. Goh. Memetic Gradient Search. *IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)*, 1-6:2894 – 2901, June 2008.

[105] J. J. Liang, P. N. Suganthan, and K. Deb. Novel composition test functions for numerical global optimization. *Proceeding of the Swarm Intelligence Symposium (SIS)*, pages 68–75, 2005.

[106] K.-H. Liang, X. Yao, and C. Newton. Combining Landscape Approximation and Local Search in Global Optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1514–1520, 1999.

[107] K.-H. Liang, X. Yao, and C. Newton. Evolutionary search of approximated N-dimensional landscapes. *International Journal of Knowledge-Based Intelligent Engineering Systems*, 4(3):172–183, July 2000.

[108] K.-H. Liang, X. Yao, and C. Newton. Lamarckian evolution in global optimization. *Proceedings of the 26th Annual Confjerence of the IEEE Industrial Electronics Society (IECON) 2000*, 4:2975–2980, October 2000.

[109] D. Lim, Y. Jin, Y. S. Ong, and B. Sendhoff. Generalizing Surrogate-assisted Evolutionary Computation. *IEEE Transactions on Evolutionary Computation*, 14(3):329–355, June 2010.

[110] D. Lim, Y. S. Ong, Y. Jin, and B. Sendhoff. A Study on Metamodeling Techniques, Ensembles, and Muti-Surrogates in Evolutionary Computation. *GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 1288–1295, 2007.

[111] D. Lim, Y. S. Ong, Y. Jin, and B. Sendhoff. Evolutionary Optimization with Dynamic Fidelity Computational Models. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 235–242, 2008.

[112] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded Memetic Algorithms with Crossover Hill-climbing. *Evolutionary Computation*, 12(3):273–302, 2004.

[113] H. Lu and G. G. Yen. Rank-density-based multiobjective genetic algorithm and benchmark test function study. *IEEE Transactions on Evolutionary Computation*, 7(4):325–343, 2003.

[114] D. J. Mackay. Introduction to Gaussian processes. *Neural Networks and Machine Learning*, 168:133–165, 1998.

[115] B. Manderick, M. K. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 143–150, 1991.

[116] S. Menzel and B. Sendhoff. Representing the Change - Free Form Deformation for Evolutionary Design Optimisation. *Evolutionary Computation in Practice*, pages 63–86, 2008.

[117] P. Merz. *Memetic algorithms for combinatorial optimization problems: fitness landscapes and effective search strategies*. PhD thesis, University of Siegen, Germany, 2000.

[118] P. Merz. Advanced fitness landscape analysis and the performance of memetic algorithms. *Evolutionary Computation*, 12(3):303–325, 2004.

[119] Z. Michalewicz. *Genetic Algorithms+ Data Structures= Evolution Programs*. Springer-Verlag London, UK, 1996.

[120] H. Miihlenbein and G. Paafi. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature-PPSN IV*, pages 178–187, 1996.

[121] D. Molina, F. Herrera, and M. Lozano. Adaptive Local Search Parameters for Real-Coded Memetic Algorithms. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 888–895, 2005.

[122] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera. Memetic Algorithm for Intense Local Search Methods Using Local Search Chains. *Hybrid Metaheuristics*, 5296:58–71, 2008.

[123] D. Molina, M. Lozano, C. García-Martínez, and F. Herrera. Memetic Algorithms for Continuous Optimisation Based on Local Search Chains. *Evolutionary Computation*, 18(1):27–63, 2010.

[124] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826:1989, 1989.

[125] P. Moscato. Memetic algorithms: a short introduction. *Mcgraw-Hill's Advanced Topics In Computer Science Series*, pages 219–234, 1999.

[126] C. L. Muller, B. Baumgartner, and I. F. Sbalzarini. Particle Swarm CMA Evolution Strategy for the Optimization of Multi-Funnel Landscapes. *Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC' 2009)*, pages 2685–2692, 2009.

[127] J. A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.

[128] Q. C. Nguyen, Y. S. Ong, H. Soh, and J. L. Kuo. Multiscale approach to explore the potential energy surface of water clusters $(H_2O)_n$, $n \leq 8$. *The Journal of Physical Chemistry A*, 112(28):6257–6261, 2008.

[129] Q. H. Nguyen. *Adaptive and Formal Memetic Frameworks*. PhD thesis, School of Computer Engineering, Nanyang Technological University, Singapore, 2010.

[130] Q. H. Nguyen, Y. S. Ong, and N. Krasnogor. A study on the design issues of Memetic Algorithm. *IEEE Congress on Evolutionary Computation (CEC), 25-28 September 2007*, pages 2390–2397, 2007.

[131] Q. H. Nguyen, Y. S. Ong, and M. H. Lim. Non-genetic transmission of memes by diffusion. pages 1017–1024, 2008.

[132] Q. H. Nguyen, Y. S. Ong, and M. H. Lim. A Probabilistic Memetic Framework. *IEEE Transactions on Evolutionary Computation*, 13:604–623, 2009.

[133] Q. H. Nguyen, Y. S. Ong, M. H. Lim, and N. Krasnogor. Adaptive Cellular Memetic Algorithms. *Evolutionary Computation*, 17(2):231–256, 2009.

[134] N. Noman and H. Iba. Accelerating Differential Evolution using an Adaptive Local Search. *IEEE Transactions on Evolutionary Computation*, 12(1):107–125, 2008.

[135] L. Ojamäe, I. Shavitt, and S. J. Singer. Potential models for simulations of the solvated proton in water. *The Journal of Chemical Physics*, 109(13):5547–5564, 1998.

[136] Y. S. Ong and A. J. Keane. Meta-Lamarckian learning in memetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):99–110, 2004.

[137] Y. S. Ong, M. H. Lim, and X. S. Chen. Memetic Computation-Past, Present & Future [Research Frontier]. *IEEE Computational Intelligence Magazine*, 5(2):24–31, 2010.

[138] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong. Classification of Adaptive Memetic Algorithms: A Comparative Study. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 36(1):141–152, 2006.

[139] Y. S. Ong, P. B. Nair, and A. J. Keane. Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modelling. *American Institute of Aeronautics and Astronautics Journal*, 41(4):687–696, 2003.

[140] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong. Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems. *Knowledge Incorporation in Evolutionary Computation, Studies in Fuzziness and Soft Computing Series. Springer Verlag*, 2004.

[141] Y. S. Ong, P. B. Nair, and K. Lum. Max-min surrogate-assisted evolutionary algorithm for robust design. *IEEE Transactions on Evolutionary Computation*, 10(4):392–404, 2006.

[142] Y. S. Ong, Z. Zhou, and D. Lim. Curse and Blessing of Uncertainty in Evolutionary Algorithm Using Approximation. *IEEE Congress on Evolutionary Computation (CEC)*, pages 2928–2935, 2006.

[143] I. Paenke, Y. Jin, and J. Branke. Balancing population and individual level adaptation in changing environments. *Adaptive Behavior*, 17(2):153–174, 2009.

[144] I. Paenke, T. J. Kawecki, and B. Sendhoff. The influence of learning on evolution: A mathematical framework. *Artificial Life*, 15(2):227–245, 2009.

[145] I. Paenke, B. Sendhoff, J. Rowe, and C. Fernando. On the adaptive disadvantage of Lamarckianism in rapidly changing environments. *Advances in Artificial Life*, pages 355–364, 2007.

[146] I. Paenke, B. Sendhoff, J. Rowe, and C. Fernando. On the Adaptive Disadvantage of Lamarckianism in Rapidly Changing Environments. *European Conference on Artificial Life (ECAL), 10-14 September 2007, Lecture Notes in Computer Science*, 4648:355–364, 2007.

[147] K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing*, 1(2):235–306, 2002.

[148] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz. BOA: The Bayesian optimization algorithm. 1:525–532, 1999.

[149] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A Survey of Optimization by Building and Using Probabilistic Models. *Computational optimization and applications*, 21(1):5–20, 2002.

[150] F. Peng, K. Tang, G. Chen, and X. Yao. Population-based algorithm portfolios for numerical optimization. *IEEE Transactions on Evolutionary Computation*, 14:782 – 800, 2010.

[151] R. Poli, J. Kennedy, and T. Blackwell. Particle Swarm Optimization: An overview. *Swarm Intelligence*, 1(1):33–57, 2007.

[152] P. Posik. Real Parameter Optimisation Using Mutation Step Co-evolution. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 872–879, 2005.

[153] M. J. D. Powell. An Efficient Method for Finding the Minimum of A Function of Several Variables without calculating derivatives. *Computer Journal*, 7(2):155–162, 1964.

[154] M. J. D. Powell. Radial basis functions for multivariable interpolation: a review. pages 143–167, 1987.

[155] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, 2002.

[156] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Verlag, 2005.

[157] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part I: Basic properties of selection and mutation. *IEEE Transactions on Neural Networks*, 5(1):102–119, 1994.

[158] X. Qi and F. Palmieri. Theoretical analysis of evolutionary algorithms with an infinite population size in continuous space. Part II: Analysis of the diversification role of crossover. *IEEE Transactions on Neural Networks*, 5(1):120–129, 1994.

[159] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.

[160] R. G. Regis and C. A. Shoemaker. Local function approximation in evolutionary algorithms for the optimization of costly functions. *IEEE Transactions on Evolutionary Computation*, 8(5):490–505, 2004.

[161] J. M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence.*, 1:312–317, 1994.

[162] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, 3rd edition, 2007.

[163] J. Ronkkonen, S. Kukkonen, and K. Price. Real-Parameter Optimization with Differential Evolution. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 506–513, 2005.

[164] E. Sanchez, S. Pintos, and N. V. Queipo. Toward an optimal ensemble of kernel-based approximations with engineering applications. *Structural and multidisciplinary optimization*, 36(3):247–261, 2008.

[165] L. V. Santana-Quintero, A. A. Montao, and C. A. C. Coello. A Review of Techniques for Handling Expensive Functions in Evolutionary Multi-Objective Optimization. *Computational Intelligence in Expensive Optimization Problems*, pages 29–59, 2010.

[166] T. Sasaki and M. Tokoro. Adaptation toward changing environments: Why Darwinian in nature. *European conference on Artificial Life*, pages 145–153, 1997.

[167] J. D. Schaffer and A. Morishima. An adaptive crossover distribution mechanism for genetic algorithms. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 36–40, Hillsdale, NJ, USA, 1987. L. Erlbaum Associates Inc.

[168] H.-P. Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[169] T. W. Sederberg and S. R. Parry. Free-Form Deformation of Solid Geometric Models. *Computer Graphics*, 20(4):151–160, 1986.

[170] B. Sendhoff and M. Kreutz. A model for the dynamic interaction between evolution and learning. *Neural Process. Lett.*, 10:181–193, December 1999.

[171] D. B. Serafini. *A framework for managing models in nonlinear optimization of computationally expensive functions*. PhD thesis, Rice University, 1998.

[172] Y.-W. Shang and Y.-H. Qiu. A note on the extended rosenbrock function. *Evolutionary Computation*, 14:119–126, March 2006.

[173] L. Shi and K. Rasheed. A Survey of Fitness Approximation Methods Applied in Evolutionary Algorithms. *Computational Intelligence in Expensive Optimization Problems*, pages 3–28, 2010.

[174] P. Siarry and G. Berthiau. Fitting of tabu search to optimize functions of continuous variables. *International Journal for Numerical Methods in Engineering*, 40(13):2449–2457, 1997.

[175] P. Siarry, G. Berthiau, F. Durdin, and J. Haussy. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. *ACM Transactions on Mathematical Software (TOMS)*, 23(2):209–228, 1997.

[176] J. Smith, Y. S. Ong, S. Gustafson, M. H. Lim, and N. Krasnogor. Workshop on Memetic Algorithms. *IEEE Symposium Series on Computational Intelligence*, 2009.

[177] J. E. Smith. Co-evolving memetic algorithms: a learning approach to robust scalable optimisation. *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, 1, 2003.

[178] J. E. Smith. Coevolving Memetic Algorithms: A Review and Progress Report. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(1):6–17, 2007.

[179] H. Soh, Y. S. Ong, Q. C. Nguyen, Q. H. Nguyen, M. S. Habibullah, T. Hung, and J.-L. Kuo. Discovering Unique, Low-Energy Pure Water Isomers: Memetic Exploration, Optimization, and Landscape Analysis. *IEEE Transactions on Evolutionary Computation*, 14(3):419 –437, 2010.

[180] F. J. Solis and R. J.-B. Wets. Minimization by Random Search Techniques. *Mathematics of Operations Research*, 6(1):pp. 19–30, 1981.

[181] W. Song. *Shape Optimisation of Turbine Blade Firtrees*. PhD thesis, School of Engineering Sciences, University of Southampton, Southampton, UK, 2002.

[182] J. C. Spall. A stochastic approximation algorithm for large-dimensional systems in the Kiefer-Wolfowitz setting. *Proceedings of the 27th IEEE Conference on Decision and Control*, 2:1544–1548, 1988.

[183] J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

[184] J. C. Spall. *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*. John Wiley & Sons, April 2003.

[185] R. Storn and K. Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[186] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL Report*, 2005.

[187] W. H. Swann. A report on the development of a new direct searching method of optimization. *ICI CENTER Instrument Laboratory, Middlesborough*, 1964.

[188] K. Tang, Y. Mei, and X. Yao. Memetic Algorithm with Extended Neighborhood Search for Capacitated Arc Routing Problems. *IEEE Transactions on Evolutionary Computation*, 13(5):1151–1166, 2009.

[189] M. Tang and X. Yao. A memetic algorithm for VLSI floorplanning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(1):62–69, 2007.

[190] J. Tavares, F. B. Pereira, and E. Costa. Multidimensional Knapsack Problem: A Fitness Landscape Analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(3):604–616, June 2008.

[191] Y. Tenne. A Model-Assisted Memetic Algorithm for Expensive Optimization Problems. *Nature-Inspired Algorithms for Optimisation*, pages 133–169, 2009.

[192] Y. Tenne and S. W. Armfield. A Versatile Surrogate-Assisted Memetic Algorithm for Optimization of Computationally Expensive Functions and its Engineering Applications. *Studies in Computational Intelligence*, 92:43–72, 2008.

[193] Y. Tenne and S. W. Armfield. Metamodel accuracy assessment in evolutionary optimization,. *IEEE Congress on Evolutionary Computation*, pages 1505–1512, 2008.

[194] D. Thierens. An adaptive pursuit strategy for allocating operator probabilities. *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*, pages 1539–1546, 2005.

[195] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, and T. Rossi. An enhanced memetic differential evolution in filter design for defect detection in paper production. *Evolutionary Computation*, 16(4):529–555, 2008.

[196] H. Ulmer, F. Streichert, and A. Zell. Evolution strategies assisted by Gaussian processes with improved preselection criterion. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, 1:692–699, 2004.

[197] L. G. Valiant. Evolvability. *J. ACM*, 56(1):1–21, 2009.

[198] D. Vanderbilt and S. G. Louie. A Monte Carlo simulated annealing approach to optimization over continuous variables. *Journal of Computational Physics*, 56(2):259–271, 1984.

[199] V. N. Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

[200] S. Venkatraman and G. G. Yen. A generic framework for constrained optimization using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 9(4):424–435, 2005.

[201] A. Vicini and D. Quagliarella. Airfoil and wing design through hybrid optimization strategies. *American Institute of Aeronautics and Astronautics Journal*, 37(5):634–641, 1999.

[202] I. Voutchkov and A. Keane. Multi-objective optimization using surrogates. *Computational Intelligence in Optimization*, pages 155–175, 2010.

[203] G. P. Wagner and L. Altenberg. Perspective: Complex adaptations and the evolution of evolvability. *Evolution*, pages 967–976, 1996.

[204] D. J. Wales and I. Ohmine. Structure, dynamics, and thermodynamics of model $(H_2O)_8$ and $(H_2O)_{20}$ clusters. *J. Chem. Phys.*, 98(9):7245–7256, 1993.

[205] Z. Wang, K. Tang, and X. Yao. A Memetic Algorithm for Multi-Level Redundancy Allocation. *IEEE Transactions on Reliability*, 59(4):754–765, 2010.

[206] E. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63(5):325–336, 1990.

[207] T. White and F. Oppacher. Adaptive Crossover Using Automata. *Lecture Notes in Computer Science*, pages 229–229, 1994.

[208] D. Whitley, V. S. Gordon, and K. Mathias. Lamarckian evolution, the Baldwin effect and function optimization. *Parallel Problem Solving from Nature–PPSN III: International Conference on Evolutionary Computation, The Third Conference on Parallel Problem Solving from Nature*, pages 6–15, 1994.

[209] D. M. Wilkinson. At cross purposes. *Nature*, 412(6846):485–485, Aug. 2001.

[210] J. Wojtusiak and R. S. Michalski. The LEM3 implementation of learnable evolution model and its testing on complex function optimization problems. pages 1281–1288, 2006.

[211] A. H. Wright. Genetic algorithms for real parameter optimization. *Foundations of Genetic Algorithms*, 1:205–218, 1991.

[212] H. Xiaohui, S. Yuhui, and R. Eberhart. Recent advances in particle swarm. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2004)*, pages 90–97, 2004.

[213] Z. Yang, K. Tang, and X. Yao. Differential Evolution for High-dimensional Function Optimization. *IEEE Congress on Evolutionary Computation, 2007. CEC 2007*, pages 3523–3530, 2007.

[214] Z. Yang, K. Tang, and X. Yao. Scalability of generalized adaptive differential evolution for large-scale continuous optimization. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 2010.

[215] X. Yao. Optimization by Genetic Annealing. *Proc. of the 2nd Australian Conf. on Neural Networks (ACNN'SI), ed. M. Jabri, Sydney*, pages 94–97, 1991.

[216] X. Yao. Simulated annealing with extended neighborhood. *International Journal of Computer Mathematics*, 40:169–189, 1991.

[217] X. Yao and Y. Liu. Fast Evolutionary Programming. *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 451–460, 1996.

[218] X. Yao and Y. Liu. Fast Evolution Strategies. *Control and Cybernetics*, 26:467–496, 1997.

[219] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 3:82–102, 1999.

[220] G. G. Yen and M. Daneshyari. Diversity-based information exchange among multiple swarms in particle swarm optimization. *2006 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1686–1693, 2006.

[221] G. G. Yen and W. F. Leong. Dynamic multiple swarms in multiobjective particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans*, 39(4):890–911, 2009.

[222] G. G. Yen and H. Lu. Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation*, 7(3):253–274, 2003.

[223] B. Yuan and M. Gallagher. Experimental Results for the Special Session on Real-Parameter Optimization at CEC 2005: A Simple, Continuous EDA. *Proceedings of the 2005 IEEE Congress on Evolutionary Computation (CEC' 2005)*, pages 1792–1799, 2005.

[224] C. Yunpeng, S. Xiaomin, and J. Peifa. Probabilistic modeling for continuous eda with boltzmann selection and kullback-leibeler divergence. *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation (GECCO '06)*, pages 389–396, 2006.

[225] L. E. Zerpa, N. V. Queipo, S. Pintos, and J.-L. Salager. An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. *Journal of Petroleum Science and Engineering*, 47(3-4):197–208, 2005.

[226] Z. Zhou, Y. S. Ong, P. B. Nair, A. J. Keane, and K. Y. Lum. Combining Global and Local Surrogate Models to Accelerate Evolutionary Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37:66–76, 2007.

[227] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23:550–560, December 1997.

[228] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method for Solid and Structural mechanics*. Butterworth-Heinemann, 6th edition, 2006.

[229] J. Zurada. *Introduction to Artificial Neural Systems*. West Publishing Co., St. Paul, MN, USA, 1992.

Appendix A: Benchmark Test Problems

Some commonly used benchmark test problems already extensively discussed in the literature are used here in this research work. They represent classes of general constrained, unimodal and multimodal continuous parametric test problems. They represent classes of continuous parametric benchmark functions with diverse fitness landscapes for studying evolutionary optimization [105, 150].

In the equations, n is the dimensionality, $\mathbf{x}$ is the vector of design variables to be optimized. Note that most problems have their global optima located at the origin. In order to avoid any biased of the search algorithms on exploiting the symmetric property of the benchmark functions, the design variables are shifted. Vector $\mathbf{o}$ is the shifted global optimum and $\mathbf{z} = \mathbf{x} - \mathbf{o}$ is the vector of shifted variables. For rotated problems, including of Problem 5, 7, 8, 10, 11 and 12, $\mathbf{M}$ is the rotation matrix and $\mathbf{z} = \mathbf{M} \times (\mathbf{x} - \mathbf{o})$ is the vector of shifted rotated design variables.

Table A.1: Benchmark Functions for Real-Parameter Optimization

Function	**Benchmark test functions**	**Range of x**	**Multi***	**Non-sep***
Shifted Sphere function	$F_{Sphere} = \sum_{i=1}^{n}(z_i^2)$	$[-100, 100]^n$	No	No
Step function	$F_{Step} = 6n + \sum_{i=1}^{n}\lfloor z_i \rfloor$	$[-5.12, 5.12]^n$	No	No
Shifted Schwefel's 1.2	$F_{Schwefel1.2} = \sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2$	$[-100, 100]^n$	No	Yes
Shifted Schwefel's 1.2 with Noise	$F_{Schwefel1.2-Noise} = (\sum_{i=1}^{n}(\sum_{j=1}^{i} z_j)^2) \times (1 + 0.4\lvert N(0,1)\rvert)$	$[-100, 100]^n$	No	Yes
Shifted Rotated Elliptic	$F_{Elliptic-SR} = \sum_{i=1}^{n}(10^6)^{\frac{i-1}{n-1}} z_i^2$	$[-100, 100]^n$	No	Yes
Shifted Rosenbrock	$F_{Rosenbrock} = \sum_{i=1}^{n-1}(100 \times (z_{i+1} - z_i^2)^2 + (1 - z_i)^2)$	$[-2.048, 2.048]^n$	Yes	Yes
Shifted Rotated Ackley	$F_{Ackley-SR} = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} z_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi z_i)}$	$[-32, 32]^n$	Yes	Yes
Shifted Rotated Griewank	$F_{Griewank-SR} = 1 + \sum_{i=1}^{n} z_i^2/4000 - \prod_{i=1}^{n}\cos(z_i/\sqrt{i})$	$[-600, 600]^n$	Yes	Yes
Shifted Rastrigin	$F_{Rastrigin} = 10n + \sum_{i=1}^{n}(z_i^2 - 10\cos(2\pi z_i))$	$[-5, 5]^n$	Yes	No
Shifted Rotated Rastrigin	$F_{Rastrigin-SR} = 10n + \sum_{i=1}^{n}(z_i^2 - 10\cos(2\pi z_i))$	$[-5, 5]^n$	Yes	Yes
Shifted Rotated Weierstrass	$F_{Weierstrass-SR} = \sum_{i=1}^{n}(\sum_{k=0}^{k_{max}}(a^k \cos(2\pi b^k(z_i + 0.5))))$ $-n\sum_{k=0}^{k_{max}}(a^k \cos(\pi b^k)), a = 0.5, b = 3, k_{max} = 20$	$[-0.5, 0.5]^n$	Yes	Yes
Shift Rotated Expanded Scaffer F6	$F_{ExpandedScaffer-SR} = \sum_{i=1}^{n} F(z_i, z_{i+1}), \quad z_{n+1} = z_1$ $F(x,y) = 0.5 + \frac{\sin^2(\sqrt{x^2+y^2}) - 0.5}{(1+0.001(x^2+y^2))^2}$	$[-100, 100]^n$	Yes	Yes
Expanded Griewank plus Rosenbrock	$F_{Grie+Rosen} = \sum_{i=1}^{D} F_{Griewank}(F_{Rosenbrock}(z_i, z_{i+1})), \; z_{D+1} = z_1$	$[-3, 1]^n$	Yes	Yes

Appendix B: Individual-based Search Methods

B.1 Davies, Swann, and Campey with Gram-Schmidt orthogonalization (DSCG) method

In the DSCG procedure, the search begins with starting point $\mathbf{x}^{(k,i)} = \mathbf{x}^{(1,0)}$, where i and k are the direction and iteration counters, respectively. A line search is conducted on each independent n dimension along direction set $\{\mathbf{v}^{(k)}\}$. At the end of each iteration k, a new set of orthogonal search directions is generated. This process is repeated until convergence to a local optimum or the allowable computational budget has elapsed. For the details, the reader is referred to Figure B.1 which outlines the main steps of the DSCG procedure.

B.1.1 Pseudo-code

Algorithm 9 DSCG [129]

Initialization:
Starting point $\mathbf{x}^{(0,0)}$
Initial step length $s^{(0)}$
Accuracy ε
First set of direction $\{\mathbf{v}_i^{(0)}\} = \{e_i\}, \quad (i = 1, 2, \ldots, n)$
Iteration counter k=0 Direction counter i=1
loop
Run n line search along n direction vectors $\{\mathbf{v}_i^{(k)}\}, \quad (i = 1, 2, \ldots, n)$:
$F(\mathbf{x}^{(k,i)}) = \min_d F(\mathbf{x}^{(k,i-1)} + d\mathbf{v}_i^{(k)})$
Calculate the distance vector from the first point and the last point:
$\mathbf{z} = \mathbf{x}^{(k,n)} - \mathbf{x}^{(k,0)}$
if $(\|\mathbf{z}\| > 0)$ **then**
Do one more line search along the direction of z:
$\mathbf{v}_{n+1}^{(k)} = z/\|z\|$
$F(\mathbf{x}^{(k,n+1)}) = \min_d F(\mathbf{x}^{(k,n)} + d\mathbf{v}_{n+1}^{(k)})$
else
$\mathbf{x}^{(k,n+1)} = \mathbf{x}^{(k,n)}$
end if
if $(\|\mathbf{x}^{(k,n+1)} - \mathbf{x}^{(k,0)}\| \geq s^{(k)})$ **then**
Perform orthogonalization using Gram-Schmidt process to get new set of direction vectors $\{\mathbf{v}_i^{(k+1)}\}$
$s^{(k+1)} = s^{(k)}$
$\mathbf{x}^{(k+1,0)} = \mathbf{x}^{(k,n)}$
$\mathbf{x}^{(k+1,1)} = \mathbf{x}^{(k,n+1)}$
$k = k + 1, \quad i = 2$
else
$s^{(k+1)} = 0.1s^{(k)}$
if $(s^{(k+1)} < \varepsilon)$ **then**
End the search, return the current best point $\mathbf{x}^{(k,n+1)}$
else
$k = k + 1, \quad i = 1$
end if
end if
end loop

B.1.2 Flow chart

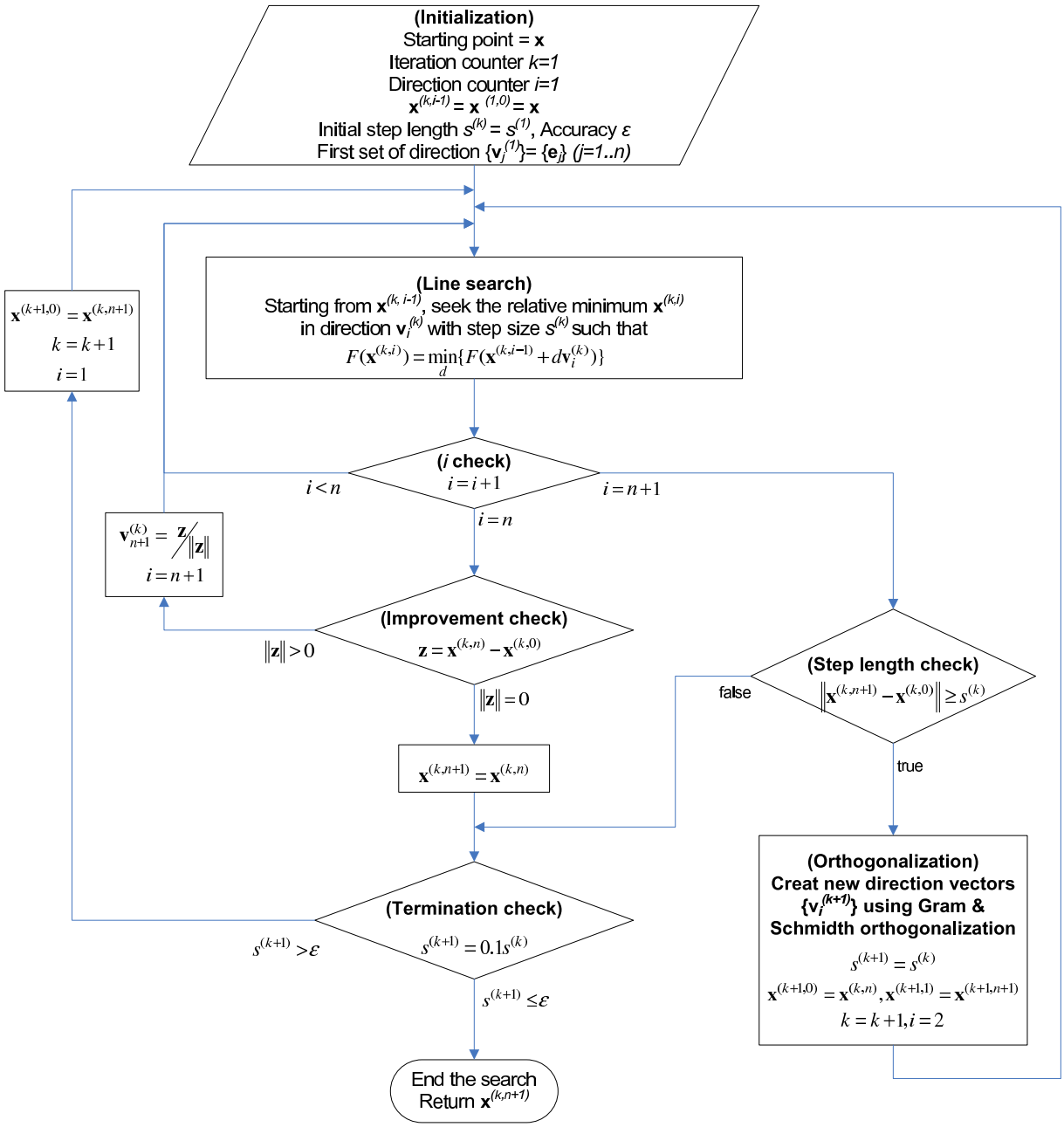


Figure B.1: The strategy of Davies, Swann, and Campey with Gram-Schmidt orthogonalization (DSCG). [129]

B.2 Broyden-Fletcher-Goldfarb-Shanno (BFGS) method

Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is a method to solve an unconstrained nonlinear optimization problem [129].

The BFGS method is derived from the Newton's method in optimization, a class of hill-climbing optimization techniques that seeks the stationary point of a function, where the gradient is 0. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and use the first and second derivatives to find the stationary point.

In Quasi-Newton methods the Hessian matrix of second derivatives of the function to be minimized does not need to be computed at any stage. The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. In multi-dimensions the secant equation is under-determined, and quasi-Newton methods differ in how they constrain the solution. The BFGS method is one of the most successful member of this class.

B.2.1 Formula

The search direction $\mathbf{p}_k$ at stage k is given by the solution of the analogue of the Newton equation

$$B_k \mathbf{p}_k = -\nabla f(\mathbf{x}_k).$$

A line search in the direction $\mathbf{p}_k$ is then used to find the next point $\mathbf{x}_{k+1}$. Instead of requiring the full Hessian matrix at the point $\mathbf{x}_{k+1}$ to be computed as B_{k+1}, the approximate Hessian at stage k is updated by the addition of two matrices.

$$B_{k+1} = B_k + U_k + V_k$$

Both U_k and V_k are rank-one matrices but have different bases. So equivalently, U_k and V_k construct a rank-two update matrix which is robust against the scale problem often suffered in the gradient descent searching. The quasi-Newton condition imposed on this update is

$$B_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = -\left(\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)\right).$$

B.2.2 Algorithm

From an initial guess $\mathbf{x}_0$ and an approximate Hessian matrix B_0 the following steps are repeated until x converges to the solution.

(i) Obtain $\mathbf{s}_k$ by solving: $B_k\mathbf{s}_k = -\nabla f(\mathbf{x}_k)$.

(ii) Perform a line search to find the optimal α_k in the direction found in the first step, then update $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{s}_k$.

(iii) $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$.

(iv) $B_{k+1} = B_k + (\mathbf{y}_k\mathbf{y}_k^T)/(\mathbf{y}_k^T\mathbf{s}_k) - (B_k\mathbf{s}_k\mathbf{s}_k^T B_k^T)/(\mathbf{s}_k^T B_k\mathbf{s}_k)$.

$f(\mathbf{x})$ denotes the objective function to be minimized. Convergence can be checked by observing the norm of the gradient, $|\nabla f(\mathbf{x}_k)|$. Practically, B_0 can be initialized with $B_0 = I$, so that the first step will be equivalent to a gradient descent, but further steps are more and more refined by B_k, the approximation to the Hessian.

The first step of the algorithm is carried out using an approximate inverse of the matrix B_k, which is usually obtained efficiently by applying the ShermanMorrison formula to the fourth line of the algorithm, giving

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(\mathbf{s}_k\mathbf{s}_k^T)(\mathbf{s}_k^T\mathbf{y}_k + \mathbf{y}_k^T B_k^{-1}\mathbf{y}_k)}{(\mathbf{s}_k^T\mathbf{y}_k)^2} - \frac{(B_k^{-1}\mathbf{y}_k\mathbf{s}_k^T + \mathbf{s}_k\mathbf{y}_k^T B_k^{-1})}{(\mathbf{s}_k^T\mathbf{y}_k)}.$$

Credible intervals or confidence intervals for the solution can be obtained from the inverse of the final Hessian matrix.

B.3 Newton-Raphson method

Newton's method gives rise to a wide and important class of algorithms that require computation of the gradient vector [129]

$$\Delta f(\mathbf{x}) = \begin{pmatrix} \partial f/\partial x_1 \\ \partial f/\partial x_2 \\ \dots \\ \partial f/\partial x_n \end{pmatrix}$$

and the Hessian matrix

$$\Delta^2 f(\mathbf{x}) = \left(\partial^2 f/(\partial x_i \partial x_j)\right)$$

Consider at iteration k:

$$f(\mathbf{x}^{(k)} + \mathbf{s}) = f(\mathbf{x}^{(k)} + \mathbf{s}) + \Delta f(\mathbf{x}^{(k)})^T \mathbf{s} + \frac{1}{2}\mathbf{s}^T \Delta^2 f(\mathbf{x}^{(k)})\mathbf{s}$$

for some small $\mathbf{s}$. When the Hessian matrix is positive definite, the above equation has a unique minimizer, obtained by solving:

$$\Delta^2 f(\mathbf{x}^{(k)})\mathbf{s} = -\Delta f(\mathbf{x}^{(k)})$$

And the next search point will be

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{s}$$

The convergence rate is quadratic, that is:

$$\|\mathbf{x}^{(k+2)} - \mathbf{x}^*\| < \|\mathbf{x}^{(k+1)} - \mathbf{x}^*\|$$

where $\mathbf{x}^*$ is the global optimum.

B.4 Evolutionary Strategy Individual Learning

Similar to the adaptation mechanism in other deterministic individual-based or local search methods, $(1+n)$ ES with Gaussian mutation adapted by the 1/5-rule is proposed and described in Algorithm 10. From the starting point $\mathbf{x}$, n candidate solutions are randomly generated using the stochastic Gaussian mutation. If the number of improved solutions is larger than $n/5$, the mutation strength is increased by a factor of *ratio*. Otherwise it is decreased. The initial point $\mathbf{x}$ is then replaced by the best candidate solution found. The process is repeated until convergence to a local optimum or the allowable computational budget has elapsed.

Algorithm 10 Evolutionary Strategy Individual Learning $(1 + n)$ ES (Starting point $\mathbf{x}$)

Initialize $good = 0, bad = 0$
Initialize mutation strength $\sigma = 1.0$
while (stopping conditions are not satisfied) **do**
  Generate the mutation vector $\mathbf{m} = \{m_i\}, m_i = N(0, \sigma)$
  Create a new search point using Gaussian mutation $\mathbf{x}^i = \mathbf{x} + \mathbf{m}$
  if ($f(\mathbf{x}^i) < f(\mathbf{x})$) **then**
    $good = good + 1$
  else
    $bad = bad + 1$
  end if
  $evaluationCount = evaluationCount + 1$
  if ($good + bad = n$) **then**
    if ($good \geq 1/5 \times n$) **then**
      Increase mutation strength $\sigma = \sigma \times ratio$
    else
      Decrease mutation strength $\sigma = \sigma / ratio$
    end if
    $good = 0, bad = 0$
  end if
  if ($evaluationCount \mod n = 0$) **then**
    Replace $\mathbf{x}$ with the best search point $\mathbf{x} = \arg\min_{\mathbf{x}^i \in \{\mathbf{x}, \mathbf{x}^1, \ldots, \mathbf{x}^n\}} f(\mathbf{x}^i)$
  end if
end while

Appendix C: Surrogate Modelling

There exist a variety of approximation methodologies for constructing surrogate models that take its roots from the field of statistical and machine learning field [75, 173]. One popular approach in the design optimization literature is polynomial regression or response surface methodology [100]. Neural networks which have shown to be effective tools for function approximation have also been employed as surrogates extensively in evolutionary optimization. These include support vector machine [30, 199], artificial neural networks [229] and radial basis function [154]. A statistically sound alternative is Gaussian process or Kriging [114], referred to as design and analysis of computer experiment models. Next, a brief overview on three different approximation methods used in the paper, namely, polynomial regression (PR), radial basic function (RBF) and Gaussian process (GP) is provided in the subsections.

C.1 Polynomial Regression

The most widely used polynomial regression model is the quadratic model which takes the form

$$\widehat{f}_M(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i \mathbf{x}^{(i)} + \sum_{1 \le i \le j \le n} \beta_{n-1+i+j} \mathbf{x}^{(i)} \mathbf{x}^{(j)} \tag{C.1}$$

where n is the number of input variables, $\mathbf{x}^{(i)}$ is the i-th component of $\mathbf{x}$, and β_i are the coefficients to be estimated. As the number of terms in the quadratic model is $n_t = (n+1)(n+2)/2$ in total, the number of training sample points should be at least n_t for proper estimation of the unknown coefficients, by means of either least square or gradient-based methods [75].

C.2 Radial Basic Function

The surrogate models in this category are interpolating radial basic function (RBF) networks of the form

$$\widehat{f}_M(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\parallel \mathbf{x} - \mathbf{x}_i \parallel) \tag{C.2}$$

where $K(\parallel \mathbf{x} - \mathbf{x}_i \parallel) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a RBF and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_m]^T \in \mathbb{R}^m$ denotes the vector of weights. The number of hidden nodes used in interpolating RBF is often assumed as equal to the number of training vector points.

Typical choices for the kernel function include linear splines, cubic splines, multiquadrics, thin-plate splines and Gaussian functions [15]. Given a suitable kernel, the weight vector $\boldsymbol{\alpha}$ can be computed by solving the linear algebraic system of equations

$$\mathbf{K}\boldsymbol{\alpha} = \mathbf{t}$$

where $\mathbf{t} = [t_1, t_2, \ldots, t_m]^T \in \mathbb{R}^m$ denotes the vector of outputs and $\mathbf{K} \in \mathbb{R}^{m \times m}$ denotes the Gram matrix formed using the training inputs (i.e., the *ij*-th element of $\mathbf{K}$ is computed as $K(\parallel \mathbf{x}_i - \mathbf{x}_j \parallel)$).

C.3 Kriging/Gaussian Process

The Kriging model or Gaussian Process (GP) assumes the presence of a global model $g(\mathbf{x})$ and an additive noise term $Z(\mathbf{x})$ in the original function.

$$f(\mathbf{x}) = g(\mathbf{x}) + Z(\mathbf{x})$$

where $g(\mathbf{x})$ is a known function of $\mathbf{x}$ as a global model of the original function, and $Z(\mathbf{x})$ is a Gaussian random function with zero mean and non-zero covariance that represents a localized noise or deviation from the global model. Usually, $g(\mathbf{x})$ is a polynomial and in many cases, it is reduced to a constant β. The approximation model of $f(\mathbf{x})$, given the m samples and the current input $\mathbf{x}$, is defined as:

$$\widehat{f}_M(\mathbf{x}) = \widehat{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{t} - \widehat{\beta}\mathbf{I}) \tag{C.3}$$

where $\mathbf{t} = [t_1, t_2, \ldots, t_m]^T$, $\mathbf{I}$ is a unit vector of length m, and $\mathbf{R}$ is the correlation matrix which can be obtained by computing the correlation function between any two samples, i.e., $\mathbf{R}_{i,j} =$

$R(\mathbf{x}_i, \mathbf{x}_j)$. While the correlation function can be specified by the user, Gaussian exponential correlation function, defined by correlation parameters $\{\theta_k\}_{k=1}^n$, has often been used:

$$R(\mathbf{x}_i, \mathbf{x}_j) = \exp\left[-\sum_{k=1}^{n} \theta_k |\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}|^2\right]$$

where $\mathbf{x}_i^{(k)}$ and $\mathbf{x}_j^{(k)}$ are the k-th component of sample points $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. $\mathbf{r}$ is the correlation vector of length m between the given input $\mathbf{x}$ and the samples $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$, i.e., $\mathbf{r} = [R(\mathbf{x}, \mathbf{x}_1), R(\mathbf{x}, \mathbf{x}_2), \ldots, R(\mathbf{x}, \mathbf{x}_m)]^T$.

The estimation of the unknown parameters β and $\{\theta_k\}_{k=1}^n$ can be carried out using the maximum likelihood method [173]. Aside from the approximation values, Kriging model or Gaussian process can also provide a confidence interval without much additional computational cost incurred. However, one main disadvantage of Gaussian process is the significant increasing of computational expense when the dimensionality becomes high, due to the matrix inversions in the estimation of parameters.

List of Related Publications

Journal

- **M. N. Le**, Y. S. Ong, S. Menzel, Y. Jin and B. Sendhoff. Evolution by Adapting Surrogates. *Evolutionary Computation Journal*. Accepted in 2012.

- **M. N. Le**, Y. S. Ong, Y. Jin and B. Sendhoff. A Unified Framework for Symbiosis of Evolutionary Mechanisms with Application to Water Clusters Potential Model Design. *IEEE Computational Intelligence Magazine*, 7(1): 20-35, 2012.

- **M. N. Le**, Y. S. Ong, Y. Jin and B. Sendhoff. Lamarckian Memetic Algorithms: Local Optimum and Connectivity Structure Analysis. *Memetic Computing*, 1(3):175-190, 2009

Conference

- **M. N. Le**, Y. S. Ong, S. Menzel, C. W. Seah and B. Sendhoff. Multi Co-objective Evolutionary Optimization: Cross Surrogate Augmentation for Computationally Expensive Problems. *IEEE Congress on Evolutionary Computation, CEC 2012. IEEE World Congress on Computational Intelligence*. Accepted in 2012.

- **M. N. Le** and Y. S. Ong. A Frequent Pattern Mining Algorithm for Understanding Genetic Algorithms. *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, 5227/2008:131-139.

- **M. N. Le**, Y. S. Ong and Q. H. Nguyen. Optinformatics for Schema Analysis of Binary Genetic Algorithms. *Genetic and Evolutionary Computation Conference, GECCO 2008*, Atlanta, Georgia, US, ACM Press, 1121-1122, 12-16 July 2008.

- B. Li, Y. S. Ong, **M. N. Le** and C. K. Goh. Memetic Gradient Search. *IEEE Congress on Evolutionary Computation, CEC 2008. IEEE World Congress on Computational Intelligence*, 2894 - 2901, 1-6 June 2008.