

Diversity Control in Evolutionary Computation using Asynchronous Dual-Populations with Search Space Partitioning



H. A. Bashir*

Department of Electrical Engineering, Bayero University, Kano, Nigeria.



ABSTRACT: Diversity control is vital for effective global optimization using evolutionary computation (EC) techniques. This paper classifies the various diversity control policies in the EC literature. Many research works have attributed the high risk of premature convergence to sub-optimal solutions to the poor exploration capabilities resulting from diversity collapse. Also, excessive cost of convergence to optimal solution has been linked to the poor exploitation capabilities necessary to focus the search. To address this exploration-exploitation trade-off, this paper deploys diversity control policies that ensure sustained exploration of the search space without compromising effective exploitation of its promising regions. First, a dual-pool EC algorithm that facilitates a temporal evolution-diversification strategy is proposed. Then a quasi-random heuristic initialisation based on search space partitioning (SSP) is introduced to ensure uniform sampling of the initial search space. Second, for the diversity measurement, a robust convergence detection mechanism that combines a spatial diversity measure; and a population evolvability measure is utilised. It was found that the proposed algorithm needed a pool size of only 50 samples to converge to optimal solutions of a variety of global optimization benchmarks. Overall, the proposed algorithm yields a 33.34% reduction in the cost incurred by a standard EC algorithm. The outcome justifies the efficacy of effective diversity control on solving complex global optimization landscapes.

KEYWORDS: Diversity, exploration-exploitation tradeoff, evolutionary algorithms, heuristic initialisation, taxonomy.

[Received December 31, 2019, Revised May 21, 2020, Accepted August 21, 2020]

Print ISSN: 0189-9546 | Online ISSN: 2437-2110

I. INTRODUCTION

The notion of diversity in evolutionary computation is synonymous with that in other population-based search techniques. Diversity may be defined as the degree of entropy among all the sample solution points in a given pool (Squillero and Tonda, 2016; Cheng et al., 2015). Population diversity reflects the extent to which the solution pool is heterogeneous or homogeneous. When diversity is assessed based on the spread (i.e., the genotypic distance) of the sample points within the feasible search space, it is referred to as *spatial/genotypic diversity* (Corriveau et al., 2012). Otherwise, when measured in the phenotype space, diversity reflects the fitness distribution in the solution space and is called *phenotypic diversity*. When diversity is mentioned in the EC literature it often refers to spatial diversity.

From the perspective of evolutionary optimization, the availability of diverse samples at any instance in a search pool serves as a driving force for continuous evolution. Diversity allows the evolutionary operators to generate newer and possibly higher quality solutions. By their nature, evolutionary algorithms (EAs) eventually converge to a region of high quality solutions in the search space; thus, they inevitably lose

the crucial diversity in their sample pool. The convergence rate depends, partly on the mutation rate and crossover probability, and to a large extent on the *selection pressure*¹ employed.

In a global search, premature convergence refers to the collapse of evolutionary process resulting from dwindled exploratory capabilities. Over the last decades, a plethora of research works (Segura et al., 2017; Squillero and Tonda, 2016; Sharma et al., 2013) have linked ineffective diversity control to high risk of premature convergence of the evolutionary search to sub-optimal solutions. This signifies the contributions of diversity in sustaining exploration and consequently mitigating premature convergence. In essence, while exploration is maintained via diversity control, effective exploitation is sustained through controlled selection pressure over successive generations. Thus, this paper views exploration from the perspective of diversification or spread of the search over the entire search space, whereas exploitation is viewed as intensification or steering of the search to a given neighbourhood. This view is consistent with the definitions of exploration and exploitation reported in (Yi et al., 2008; Blum et al., 2008; Thierens, 1998). In a slightly more general view, Molina et al. (2010) state that exploration plays the crucial role

¹ Selection pressure describes the convergence rate and it is often defined as the ratio of the probability of selecting the currently best sample solution to that of an average sample solution (Liu et al., 2019).

*Corresponding author: habashir.ele@buk.edu.ng

of guiding population-based algorithms towards achieving global-optimality; whereas exploitation leads the algorithms to search around and within the neighbourhood of good solutions so as to produce better ones.

As in many other population-based (Meta) heuristics (Reeves, 1993; Blum et al., 2008), the ability of EAs to solve stochastic optimization tasks is aligned to the efficacy of their exploration and exploitation strategies. Although exploration and exploitation play complementary roles for achieving a robust global search mechanism, establishing a suitable balance between the two can be a difficult and challenging task in the design of EAs. This might be attributed to the facts that:

1) Simultaneous optimization of both exploration and exploitation needs a careful treatment, and could require a tradeoff, since they are sometimes-conflicting goals; and

2) EAs are mainly applied to large scale, complex problems which are usually characterised with poorly understood objective landscapes.

In other reviews, Corriveau et al. (2012) argued that, viewing exploration and exploitation as opposing forces is rather naive because this may only be true in some special cases (such as when optimizing a unimodal landscape). Thus, it would be fair to view them as orthogonal forces, making it possible to improve both simultaneously.

Clear manifestations of qualitative features that seek to strike a balance between exploration and exploitation are inherent in the core processes of the majority of evolutionary paradigms. For instance, variation operators, such as crossover and especially mutation, are believed to enhance exploration by ensuring reachability of the entire search space. The reproduction operator (i.e., selection) mainly favours exploitation of the promising region(s) of the explored search space (Potter et al., 2003; Greenhalgh and Marshall, 2000). And, as demonstrated by Wen et al. (2010), the crossover operator has the innate tendency to simultaneously enhance both exploration and exploitation. Nevertheless, it seems that such nature inspired population-based metaheuristics are mainly good for exploration of the search space and identification (but not exploitation) of the areas with high quality solutions (Blum et al., 2011).

Consequently, researchers in the EC community resort to formulating various techniques that can potentially reinvigorate the balance between exploration and exploitation. The aim has been to ensure optimum exploitation while maintaining useful level of diversity (Lozano et al., 2008)² throughout the search process. In this vein and to validate the key objective in this work, this paper seeks to verify the following hypothesis:

Diversity management and control, in a global search, enhance the exploration-exploitation balance and improve maintenance of useful diversity. As a caveat to this hypothesis, diversity control is achieved via the use of subpopulations, with a separate pool for evolution and another for diversity.

In order to validate the above hypothesis, as a key contribution, this paper proposes a dual population-based diversity control technique. Section II presents an overview of the fundamental aspects of EC methodologies, a survey of literature on analytic description of diversity measures and a classification of the various approaches to improving diversity management. The proposed dual-pool EC model is presented in Section III. Section IV presents the results and discussions of evaluations of the proposed method. Finally, the paper concludes in Section V.

II. LITERATURE REVIEW

Evolution is a process that originated from the biologically inspired neo-Darwinian paradigm, i.e., the principle of survival of the fittest (Fogel, 1997). Evolutionary algorithms mimic the intrinsic mechanisms of natural evolution to progressively yield improved solutions to a wide range of optimization problems. Therefore, the standard EA model has been successfully applied on various problem types (Fleming and Purshouse, 2002) without any incorporation of domain specific information.

While the focus in this section is to review related existing strategies utilised for diversity control in EC theory, we begin with an overview of the general concept of the standard evolutionary algorithm.

A. The Standard Evolutionary Algorithms (Single-Pool)

One of the most widely used EAs is the genetic algorithm (GA)³. As originally inspired by (Holland, 1975), GA is an iterative procedure (Algorithm 1) that evolves a pool of candidate solutions across generations t . It starts with an initial fixed sized set of candidate solutions called the population, $P(t) : |P(t)| = N$ (lines 2-3). A candidate solution point x_i is called an *individual*, and represents a single possible solution to the problem under consideration, i.e., in the phenotype space $x_i \in \mathcal{P}$. A candidate solution, $x'_i \in \mathcal{G}$, is a representation of an individual by a computational data structure called a *chromosome* in the genotype space \mathcal{G} . Usually, a chromosome is encoded as a string of symbols of finite-length called *genes*. An encoded chromosome may be in the form of binary bit string, real-valued or any otherwise representation (Goldberg and Holland, 1988; Pengfei et al., 2010). Typically, the chromosomes in the initial population (line 3) are created randomly or via a simple heuristic construction.

Following an initial evaluation that is based on some measure of fitness (lines 4), in every evolutionary cycle t , called a generation (lines 5-12), a stochastic selection process (line 6) is applied on the initial population to choose better solutions. The selected solutions $Q_s(t)$, called parent, undergo the evolutionary variation processes – crossover and mutation (lines 7-8). The evolutionary cycle (lines 5-12) repeats and the average fitness of the population is expected to grow with successive generations. The process stops when a termination

² Useful diversity refers to the population diversity that in some way helps produce good solutions.

³ Genetic algorithm will be used in this work and unless otherwise stated, any subsequent mentions of EC or EA in this paper will be referring to the genetic algorithm.

Algorithm 1: A Standard Model of Genetic Algorithm (Holland, 1975).

```

1: begin
2:   t ← 0;
3:   initialise P(t) : P(t) = {xi | xi ∈ P};
4:   evaluate the fitness of P(t);
5:   while not termination do
6:     Qs(t) ← select from P(t);
7:     Qr(t) ← recombine Qs(t);
8:     Qm(t) ← mutate Qr(t);
9:     evaluate the fitness of Qm(t);
10:    P(t + 1) ← select from {Qm(t) ∪ P(t)};
11:    t ← t + 1;
12:  end while
13: end

```

condition is satisfied. Typical termination conditions enforce a user-defined limit on function evaluations, execution time, or when the solution pool, $P(t)$, sufficiently converges to the optimum – or at least a suboptimal – solution. The following section introduces diversity measurement and analysis in EC.

B. Analytic Description of Evolutionary Diversity Measures

Diversity reflects the degree of entropy among all the sample solution points in a given pool. It is arguably the natural source of power for sustainable progress in evolutionary search. Diversity crucially contributes to the inherent adaptive capabilities of EAs (Cobb and Grefenstette, 1993) which made them suitable for a wide range of global optimization problems. In the following, some parametric mathematical models for the diversity measures are presented.

1.) Evaluating the Coefficient of Diversity

At every generation t , the instantaneous diversity among the sample solutions $\{x_i\}$ in any search pool $P(t)$ of size $|P(t)| = N$ is measured using a Euclidean distance measure. The diversity is then expressed in terms of a coefficient of diversity C_{Div} . However, the approach to determining a suitable reference sample point from which the distance of every sample solution is measured varies. On one hand, the locus of the current best sample solution has been used as the reference point (Herrera and Lozano, 1996). On the other hand, a *centroid* point (i.e., a hypothetical average sample point position), re-evaluated at every generation is often used (McGinley, 2011; Eshelman and Schaffer, 1991; Taejin and Kwang, 2010). The latter approach tends to yield an unbiased estimate of the true spread of the solution points; it is therefore adopted in this paper. Thus, in the following, C_{Div} is derived by evaluating the distance of every sample solution from a chosen reference point.

Suppose that a search pool $P(t)$ of size N consists of a set of sample solutions $\{X_i \in P(t) : X_i \in \mathbb{R}^n\}$, where n is the dimensionality of the problem, then at any given dimension j the vertex c_j of a centroid sample point is

$$c_j = \frac{1}{N} \sum_{i=1}^N x_{ij}. \quad (1)$$

Therefore, for an n -dimensional problem, the position of the centroid sample C at any instance is

$$C = \frac{1}{N} \sum_{i=1}^N [x_{i1}, x_{i2}, \dots, x_{in}] = c_1, c_2, \dots, c_n, \quad (2)$$

where the instance is aligned to the temporal evolution of the pool across generations t , such that as time evolves, the position of the centroid is tracked through the search space.

Consequently, the Euclidean distance σ_j of all the sample points N across any dimension j is

$$\sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{ij} - c_j)^2}. \quad (3)$$

Hence, the instantaneous spatial diversity across all n dimensions for any sample pool $P(t)$ is expressed as a coefficient of diversity C_{Div} , such that

$$C_{Div} = \frac{1}{n} \sum_{j=1}^n \sigma_j. \quad (4)$$

And the vector of the overall temporal spatial diversity for all generations $t = 1, 2, \dots, k$ is

$$C_{Div} = C_{Div}^1 C_{Div}^2 \dots C_{Div}^k. \quad (5)$$

It is noteworthy that majority of the conventional and problem dependent diversity measures suffer from either: (i) sensitivity to distribution of outlier samples, (ii) sensitivity to changes in pool sizes or (iii) changes in problem dimension. However, experimental findings, comparing various spatial diversity measures by Corriveau et al. (2012), have justified the suitability of the above measure (4).

2.) Normalisation of the Coefficient of Diversity

Normalisation helps filter out the effects of varying pool sizes or problem types from the true diversity dynamics of a given EC model. It therefore aids effective evaluation of diversity across populations and/or generations.

To normalise the coefficient of diversity C_{Div} (5), a running normalisation with the maximum coefficient of diversity C_{Div}^{\max} is utilised. The technique comes from the intuition that, since the initial pool is generally created from a random uniform distribution, unless a more diverse pool is found over the course of the evolution, the coefficient of diversity in the first generation C_{Div}^1 comes from the most diverse population and is used as the initial normalisation factor C_{Div}^{\max} . Subsequently, if a more diverse pool is found at any generation t , then the newly found $C_{Div}^t | C_{Div}^t = \max_i(C_{Div}^i)$ is used to update the normalised \tilde{C}_{Div} vector as in (6):

$$\tilde{C}_{Div} = \frac{C_{Div}}{C_{Div}^{\max}} = \frac{C_{Div}^1 C_{Div}^2 \dots C_{Div}^k}{C_{Div}^{\max}}, \quad (6)$$

where $C_{Div}^{\max} = \max_i(C_{Div}^i)$. This approach, referred to as normalisation with maximum diversity thus far (NMDF) (Corriveau et al., 2012), is immune to variations in problem dimensions or pool sizes. In the context of measuring convergence, a new technique for evaluation of a population's evolvability is developed.

3.) Evaluating Population Evolvability

In order to examine the independent effect of EC operators while in interaction, Price (Frank, 1997) formulated a theorem that permits decomposition of the evolutionary process to separate the genetic effect (or contribution) of the selection operator from that of other variation operators (i.e. crossover and mutation).

The population evolvability measure was derived from an extension to the Price's equation originally proposed by Bassette et al. (2004). By using the small changes in the crossover's contribution, the population evolvability measures the ability of the pool to generate new solutions at any instance during the course of the evolution. Consequently, it dynamically assesses convergence of a search pool by monitoring the contribution of the crossover operator to the fitness growth and diversity profile of the search pool.

The extension to the Price's equation proposed in (Bashir and Neville, 2012) separates the individual contributions of variation operators to fitness growth during the evolution. Thus, (7) has a term for selection, crossover (\mathcal{X}) and mutation (\mathcal{M}) operators:

$$\begin{aligned} \Delta Q &= f(\text{selection}) + f(\text{crossover}) + \\ &\quad f(\text{mutation}) \\ &= \frac{\text{Cov}(z, q)}{\bar{z}} + \frac{\sum_{i=1}^N z_i \Delta q_{i\mathcal{X}}}{N\bar{z}} + \frac{\sum_{i=1}^N z_i \Delta q_{i\mathcal{M}}}{N\bar{z}}, \end{aligned} \quad (7)$$

where $\Delta Q = Q_2 - Q_1$ is the change in the fitness Q , N is the population size, z_i is the number of offspring of parent i , and $\bar{z} = (\sum_i z_i)/N$ is the average number of the offspring produced. The first term in (7) modelled the effect of selection operator in terms of the covariance between the individuals z and their fitness q . For the crossover and mutation, $\Delta q_i = q'_i - q_i$ is the difference between the average fitness q of the offspring of parent i measured before and after the application of operator \mathcal{X} or \mathcal{M} . Thus, each term in (7) estimates the changes in the average change in population's fitness (Δq) due to one of the three genetic operators.

Suppose that the change in fitness due to crossover operator is:

$$\Delta Q_{\mathcal{X}} = \frac{\sum_{i=1}^N z_i \Delta q_{i\mathcal{X}}}{N\bar{z}}, \quad (8)$$

then, the width of one standard deviation envelope ($\pm\sigma$) for the effect of the crossover operator on fitness growth at every t th generation lies within the interval:

$$[\Delta Q_{\mathcal{X}t} - \sigma_{\mathcal{X}t}, \Delta Q_{\mathcal{X}t} + \sigma_{\mathcal{X}t}], \quad (9)$$

where $\Delta Q_{\mathcal{X}t}$ is the change in the average fitness of the population at iteration t due to the crossover operator; $\sigma_{\mathcal{X}t}$ is the corresponding standard deviation. Therefore, the population evolvability measure σ_{Xover} is defined as:

$$\sigma_{\text{Xover}} = (\Delta Q_{\mathcal{X}t} + \sigma_{\mathcal{X}t}) - (\Delta Q_{\mathcal{X}t} - \sigma_{\mathcal{X}t}) = 2\sigma_{\mathcal{X}t}. \quad (10)$$

The following section classifies some diversity control approaches in EC.

C. Taxonomy of Diversity Control Policies in EC

The presentation herein is underpinned by a classification of the commonly used diversity control policies in EC as demonstrated by the research relevance tree in Fig. 1. This classification permits separation of the fundamental research domain into a number of possible approaches. It aids in the design and development phases and translates into

mathematical model parameters and data structures for the dual-pool EC model proposed in Section III.

Generally, the diversity control approaches presented in Fig. 1 are based on the following frequently used methods:

- 1) heuristic population initialisation strategies,
- 2) multipopulations models, and
- 3) hybrids and portfolios of algorithms.

Note, however, that the classification in Fig. 1 is by no means exhaustive of the multitude of approaches that could be used in diversity control.

1.) Heuristic Population Initialisation

Traditionally, the initial pool in evolutionary algorithms is generated in a random manner, by means of a uniform distribution (De Jong, 1975). For any given n -dimensional search space $\mathcal{D} \in \mathbb{R}^n$, the sample solution points x , are randomly created within the feasible boundaries such that the initial pool is:

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid a_i \leq x_i \leq b_i\}; \quad i = 1, 2, \dots, n. \quad (11)$$

where a_i and b_i are the lower and upper bounds of the i th dimension.

The random sampling described by Equation (11) yields a problem independent means of starting any population-based stochastic search process. However, from the last decade, a number of researchers (Maaranen et al., 2004; Tometzki and Engell, 2011; Rahnamayan et al., 2007) have suggested that using *quasi-random* heuristics for population initialisation can have a profound impact on not only the search efficiency (i.e., convergence speed), but also the overall quality of the resulting final solution. This intuition comes from the fact that, even with no *a priori* information on the nature of the final solution, heuristic initialisation can ease the generation of more diverse and probably fitter samples (Xu et al., 2019).

In an attempt to examine the benefits of a uniformly distributed sample over a mere randomly generated one, Maaranen et al. (2004) used quasi-random sequences to generate initial pool. Although the distribution property of their quasi-random sequence seems to degrade with increase in dimension, they found that the pools generated using quasi-random sequences, which try to imitate points with a perfect uniform distribution, tend to cover the entire feasible search space more optimally. Similarly, Rahnamayan et al. (2007) proposed a novel approach that utilised opposition-based learning⁴ to generate the initial pool for a genetic algorithm. The authors (Rahnamayan et al., 2007) reported acceleration in the algorithm's overall convergence speed. In the same vein, Melo and Botazzo (2012) reported improvements in the evolutionary search when a smart sampling technique is used for creation of the initial pool.

⁴ Opposition-based learning works based on the theory of opposite numbers, see Melo and Botazzo (2012) for details.

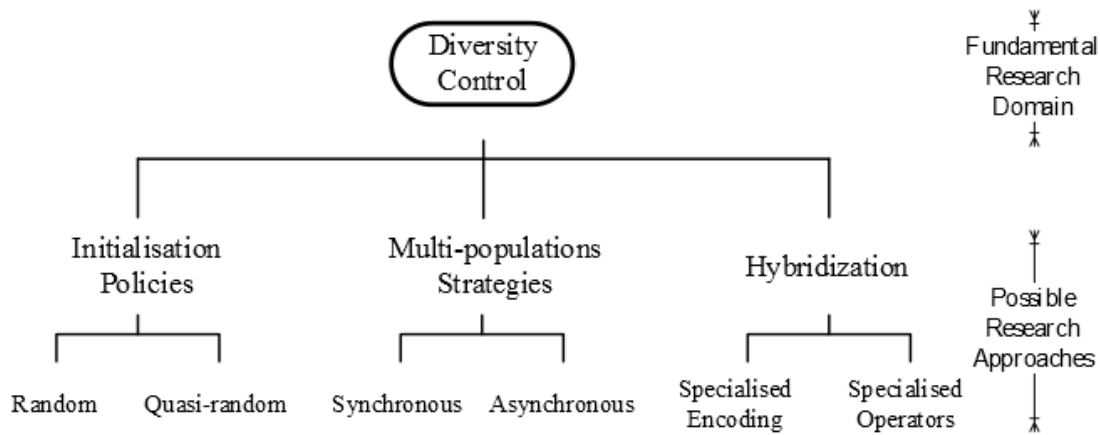


Fig. 1: A Research Relevance Tree for the Diversity Control Approaches used in Evolutionary Computation.

Elsewhere, investigations by Morrison (2003) led to the conclusions that heuristic initialisations neither yield significant improvement in the quality of the final solution, nor do they reduce the required number of function evaluations. But as compared to random initialisation, they can minimise stochastic effects in the end result of the evolution by reducing the variance in the solution quality across independent runs.

Further details on heuristic initialisation can be found in (Tometzki and Engell, 2011) where three different initialisation approaches are used as a pre-processing phase. Their results (Tometzki and Engell, 2011) suggested that heuristic initialisations can potentially improve convergence speed and solution quality.

2.) Multipopulation Strategies

A commonly used diversity control strategy is the multipopulation approach. This has its inspiration from the biological notion of *nicing* and *speciation* (Shir, 2012) wherein diversity is enforced by promoting species formation within a population.

Multipopulation strategies generally vary in their processes of subpools creation and in their adopted migration policies. In the majority of multipopulation strategies, subpopulations are run concurrently and evolve by optimizing a common objective. Thus, they can be classified as “*synchronous*”, see Fig. 1. In island models (Alba and Tomassini, 2002) for example, the search begins with two or more subpopulations which exchange information via periodic migrations. But the number and size of the subpopulations are mainly predetermined by the user and are then kept unchanged throughout the evolution (Alba and Tomassini, 2002; Branke et al., 1998). Other synchronous multipopulation methods dynamically create subpopulations and adjust their numbers and sizes during the course of evolution (Tenne and Armfield, 2005; Branke, 2000). Hence, such methods avoid premature convergence of the evolutionary search by continuously evolving with multiple pools.

The other category is the “*asynchronous*” operation of subpopulations (Fig. 1). In this case, the creation and evolution objectives of the subpools differ. For instance, the dual-population GA (DPGA) proposed by Taejin and Kwang (2010)

has a main and a reservoir population. The main pool has the fitness of its samples evaluated based on the problem’s original objective function, whereas the reservoir pool is evaluated with an objective that exclusively optimizes diversity. The authors (Taejin and Kwang, 2010) found that the DPGA excels on highly multimodal functions having densely populated peaks, but struggles on sparse landscapes.

Other multipopulation-based EAs include the multinational EA (Ursem, 1999) and forking GA (Tsutsui, 1997). The next section reviews some recent hybrid methodologies deployed to enhance diversity control.

3.) Hybrids and Portfolios of Algorithms

In general, population-based methods are believed to be robust in attaining global optimality via wide exploration (Blum et al., 2011; Joines and Kay, 2003; Michalewicz, 1994). Yet, their lack of intense exploitation capabilities limits their effectiveness in dealing with complex global optimization tasks. To strengthen exploration-exploitation balance and optimize diversity, various approaches that combine algorithmic models in form of hybrids or memetic algorithms (Moscatto, 1999; Gong, et al., 2019; Liu et al., 2019) were developed. Such approaches usually hybridize EAs with various local improvement procedures made from local search algorithms.

Joines and Kay (2003) examine the behaviour of a variety of hybrid algorithms. They found that hybrid models, whether based on Baldwinian, Lamarckian frameworks, tend to achieve good exploration-exploitation balance as compared to their non-hybrid counterparts. Further theoretical analyses on hybrid frameworks which seek to balance exploration-exploitation tradeoff can be found in (Jih-Yiing and Ying-Ping, 2011; Dang et al., 2019).

From the above reviewed methodologies, it is evident that the challenges in designing an effective diversity control policy require a multifaceted approach. In fact, other approaches that sought to maintain population diversity by dynamically controlling evolutionary parameters, such as mutation and crossover rates, can be found in (McGinley, 2011; Eiben, 1999).

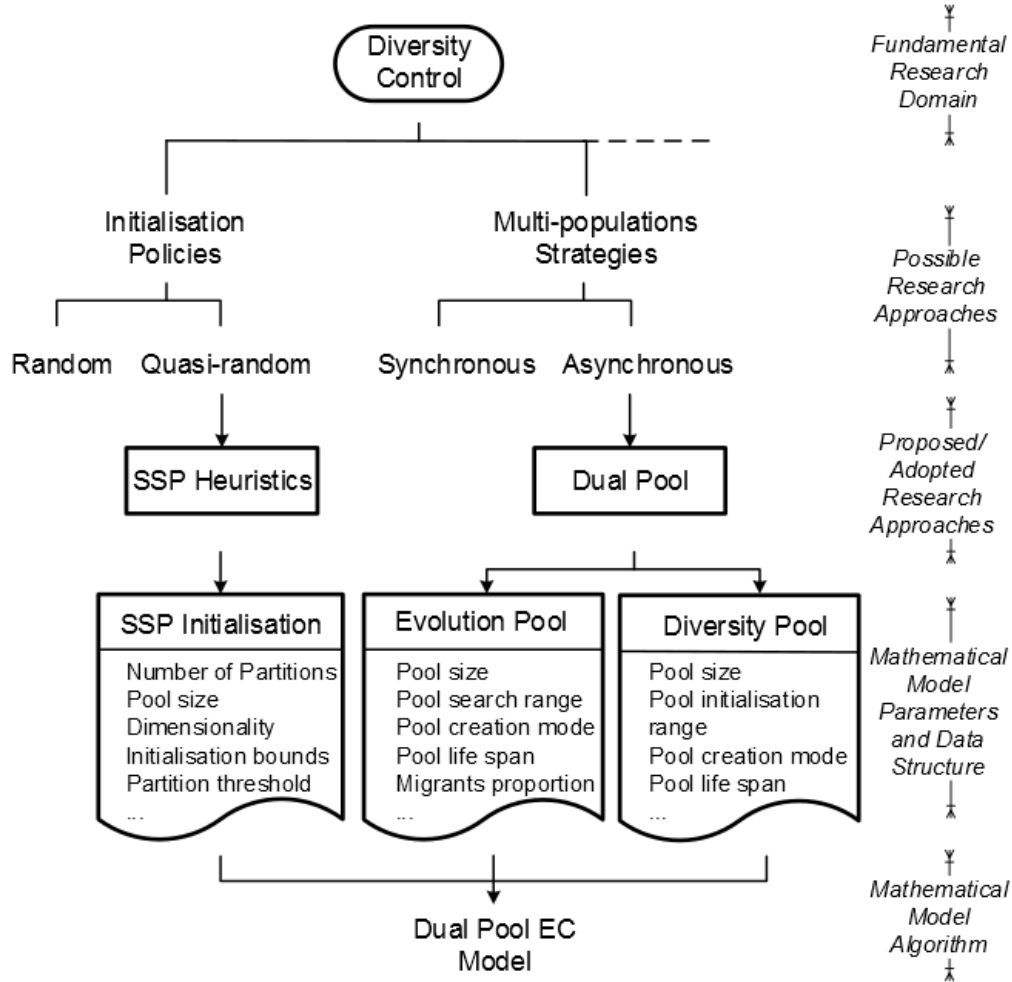


Fig. 2: Theory Research Relevance Tree: A roadmap to designing a Dual-Pool EC model with Search Space Partitioning (SSP) heuristic initialisation.

D. Prevailing Challenges in Diversity Control

Diversity control, using Multi-pool EC models, continued to receive attention (Xianshun, 2011) in the EC literature. However, conventional approaches in the literature mostly involve concurrent and continuous runs of the multiple pools. Such *synchronous* approaches often lead to a severe increase in function evaluations, resulting in increased overall computational cost. In addition, traditional multipool approaches are generally faced with huge overhead due to proliferation of secondary parameters. In general, designing multipopulation models requires a prior decision on the *creation* and *management* strategies of the multiple pools. Thus, one has to decide on the criteria upon which the subpopulations evolve within themselves and communicate with one another, i.e., the migration policies among subpools.

Additional parameters, such as the minimum and maximum pool sizes, the initial number of pools, thresholds for the minimum and maximum number of subpools (when dynamic pool creation is utilised), etc., must be decided. Crucially, the parameter tuning task quickly become

intractable since the optimum settings for such additional parameters are problem dependent.

III. A DUAL-POOL EC MODEL FOR EFFECTIVE DIVERSITY CONTROL

This section proposes a dual-pool EC model that enjoys the benefits of the multipool framework combined with a heuristic initialisation. Specifically, as shown in the theoretic research relevance tree in Fig. 2, the proposed approach integrates a quasi-random heuristic initialisation, called *search space partitioning* (SSP), into a dual population architecture to facilitate temporal diversity control. The dual-pool model is made up of an *evolution pool* (i.e., the main pool), and a *diversity pool*. The evolution pool primarily undergoes the evolutionary optimization process, whereas the diversity pool is created, on-demand, to reinstate diversity into the evolution pool. Preliminary to the development of the proposed EC model, the characteristic data sets and mathematical model parameters for the proposed techniques (SSP initialisation and

the dual-pool) are specified in Fig. 2. Details of each of these follow in turn.

A. The Dual-Pool EC Architecture

In contrast to the conventional multipool architectures which are mainly *synchronous* in nature, the proposal herein is aimed at suggesting a framework (see Fig. 2) that combines an *asynchronous* dual-pool model with a heuristic population initialisation. This enables robust diversity control by minimising the communication overhead among subpools.

1.) The Evolution Pool – Creation and Working

Playing the role of the main pool, the evolution pool P_{Evo} serves as the initialisation point for the global search. The size of this is equal to the actual population size (N) for the overall search. In order to ensure the feasibility of the initial samples, P_{Evo} is created (uniform at random) within the feasible boundaries of the search space, such that:

$$P_{Evo} \leftarrow \mathbf{x} \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}] : \mathbf{x} \in \mathbb{R}^n, \quad (12)$$

where n is the problem dimension, \mathbf{x} is the vector of design variables, $\underline{\mathbf{x}}$ and $\bar{\mathbf{x}}$ are vectors of lower and upper bounds respectively.

2.) The Diversity Pool – Creation and Working

The proposed method is based on a Dual-pool approach that runs in an *asynchronous* mode (Fig. 2). Thus, the diversity pool P_{Div} is only occasionally used to restore useful diversity into the evolution pool. Consequently, P_{Div} is created only after and whenever a sufficient convergence of the evolution pool is detected. The following section presents a mathematical model that describes the creation process of the diversity pool P_{Div} .

B. Search Space Partitioning (SSP) Heuristic Initialisation

In order to improve diversity by enforcing uniformity in the coverage of the entire feasible search space, a strategy that generates the diversity pool P_{Div} using a quasi-random heuristic called search space partitioning (SSP) is proposed. SSP partitions the search space into uniformly sized hypercubes and repeatedly creates one random sample from each hypercube until the required pool size (N) is reached.

Given any n -dimensional search space $\mathcal{D} \in \mathbb{R}^n$ (Algorithm 2), let each of its dimensions be segmented into κ equal partitions (line 4). Suppose that $\rho(\kappa) = \{m_1, m_2, \dots, m_n\}$ is the set of the resulting partition sizes for each of the $j = 1, \dots, n$ dimensions (line 5). Then, along each dimension j , the partition sizes m_j are assumed to be uniform. Therefore, SSP segments the original search space \mathcal{D} into $\phi_\kappa = \kappa^n$ equal-sized subspaces (hypercubes) (line 6). For each subspace ϕ_κ , let $\mathbf{x}_i = [\underline{x}_i, \bar{x}_i] \in \mathbb{R}^n$ be a uniformly distributed random sample generated within the boundaries of m_j . Then, SSP applies a uniform distribution to generate equal number of samples across the entirety of the partitioned search space ϕ_κ (lines 7-8).

Thus, in the proposed SSP heuristic, the required *minimum* population size N relates to the number of partitions κ according to the following model:

$$N = \kappa^n, \quad (13)$$

Algorithm 2: Search Space Partitioning Quasi-random Heuristic.

Define and set search space (\mathcal{D}) parameters

- 1: dimensions n ;
- 2: total population size N ;
- 3: bounds $\mathcal{D} \in \mathbb{R}^n = \{\mathbf{x}_j \in \mathbb{R} \mid \underline{x}_j \leq x_j \leq \bar{x}_j; j = 1, \dots, n\}$;

Define and set partition parameters

- 4: set the number of partitions to κ ;
- 5: evaluate partition sizes m for each dimension:

$$\rho(\kappa) = \left\{ m_j = \frac{\bar{x}_j - \underline{x}_j}{\kappa} \mid j = 1, \dots, n \right\};$$
- 6: segment \mathcal{D} into n -dimensional equal-sized subspaces

$$\phi_\kappa = \left\{ \kappa^n = \left(\prod_{j=1}^n m_j \right)^n \right\};$$
- 7: generate a random sample $\{\mathbf{x}_i = \cup(\phi_\kappa)\}$ from each ϕ_κ with uniform distribution;
- 8: repeat (7) until all N samples points are generated.

Note: Segmenting every dimension of the original search space \mathcal{D} into κ partitions yields $\phi_\kappa = \kappa^n$ subspaces.

Algorithm 3: The Dual-Pool EC Algorithm.

- 1: initialisation
 $N \leftarrow$ Pool size; $n \leftarrow$ problem dimension;
 $t \leftarrow 0$;
 - 2: initialise the evolution pool
 $P_{Evo}(t) \leftarrow \{X_i\} : X \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}], i = 1, \dots, N, j = 1, \dots, n$
 - 3: **while** not termination **do**
 - 4: run EC model and estimate convergence at every iteration
 $P_{Evo}(t), \tilde{C}_{Div}(t), \sigma_{Xover}(t) \leftarrow$ invoke EC ($P_{Evo}(t)$);
 - 5: check for convergence of $P_{Evo}(t)$
if ($\tilde{C}_{Div}(t) < C_{Div}^{min}$) **and** ($\sigma_{Xover}(t) < \sigma_{Xover}^{min}$)
 - 6: $P_{Evo}^R(t) \leftarrow$ rank $f(P_{Evo}(t))$;
 - 7: get the top $k\%$ best solutions (elite) in P_{Evo}
 $P_{Evo}^E(t) \leftarrow k\%(P_{Evo}^R(t))$;
 - 8: initialise the diversity pool (P_{Div}) using SSP heuristics (Algorithm 2)
 $P_{Div} \leftarrow \{X_i\} : X \in [\underline{\mathbf{x}}, \bar{\mathbf{x}}], i = 1, \dots, N, j = 1, \dots, n$
 - 9: evaluate and rank P_{Div} by distance from the elite
 $P_{Div}^R \leftarrow$ rank $\left| |X_{Div}^i - X_{Evo}^E| \right| : i = 1, \dots, N$
 - 10: get the farthest samples in P_{Div}^R
 $P_{Div} \leftarrow (1 - k)(P_{Div}^R)$;
 - 11: merge evolution and diversity pools to form new P_{Evo}
 $P_{Evo}(t) \leftarrow \{P_{Evo}^E(t) \cup P_{Div}\}$;
 - 12: **end if**
 - 13: $P_{Evo}(t + 1) \leftarrow P_{Evo}(t)$;
 - 14: $t \leftarrow t + 1$;
 - 15: **end while**
-

where n is the dimensionality of the search space \mathcal{D} .

Equation (13) revealed that the higher the number of partitions κ , the larger the required pool size N to achieve maximum spread for a given dimension n . This is because the two have an exponential relation with respect to the dimensionality n . Thus, the SSP quasi-random heuristic is obviously not immune to scalability problems in high dimensional problems, a phenomenon popularly known as curse of dimensionality (Shetti, 2019). Hence, for higher dimensionality problems, the number of partitions has to be regulated.

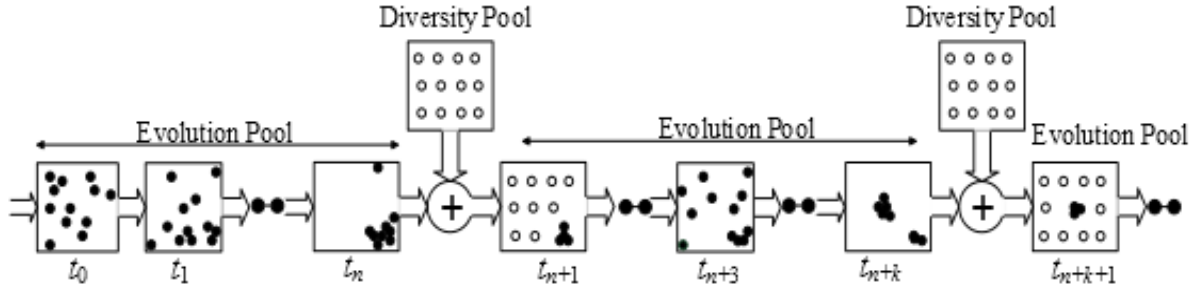


Fig. 3: The Dual-Pool EC model dynamically showing the merger of the distinct evolution pool (P_{Evo}) with the SSP created diversity pool (P_{Div}) over generations t . The periodic merging process is adaptively controlled via a robust convergence detection strategy.

C. The Proposed Dual-Pool EC Model

The complete model for the proposed dual-pool EC algorithm outlined in Algorithm 3. This model closely resembles the standard EC model (Algorithm 1) previously examined in Section II. A key distinguishing feature is in the initialisation stage (lines 1-2) where a separate evolution pool P_{Evo} is utilised. This is then later combined (line 11) with a diversity pool P_{Div} (created using the SSP heuristic initialisation, see line 8) whenever convergence is detected (line 5). The evolutionary cycle ends (line 3) when a termination condition – such as accuracy threshold or maximum evaluation limit – is reached.

Consequently, the search process proceeds such that whenever P_{Evo} converges, new samples from the diversity pool, P_{Div} , are used to restore sufficient diversity into the search process. For the proposed dual-pool EC model, the flow diagram (Fig. 3) demonstrates the dynamic merging process of the separate pools during the course of evolution. Fig. 3 reveals that most of the evolutionary cycles are run solely with the evolution pool (P_{Evo}), the diversity pool (P_{Div}) is only introduced when sufficient convergence is detected. Thus, the model allows continuous optimization via temporal exploration-exploitation cycles.

D. Visualising Diversity in EC Models

Prior to the experimental evaluations, this section examines, with the aid of a visualisation, how spatial diversity fares under both the standard EC model and the newly proposed dual-pool EC architecture.

1.) Diversity visualisation with a standard EC model

An illustration of typical temporal dynamics of the spatial diversity (\tilde{C}_{Div}) in an evolutionary pool of a standard EC algorithm (Algorithm 1) is as shown in Fig. 4c. The result comes from an EA model, applied on the Schwefel benchmark (Section IV), having a randomly initialised real-valued sample pool of size $N = 100$. The model uses BGA (Muhlenbein and Schlierkamp-Voosen, 1993)⁵ mutation and intermediate crossover operators applied at the rates of $P_m = 0.01$ and $P_c =$

1.0 respectively. A strict binary tournament selection without replacement is utilised.

It was observed that the initially diverse samples in Fig. 4(a) gradually converged towards a limited area of the search space over generations (Fig. 4(b)). To some extent, the spatial diversity falls with increasing function evaluations (Fig. 4(c)). Although this phenomenon could have been avoided by increasing the probability of mutation, it should be noted that high rates of mutation slow down the evolutionary progress and could turn the search into a random one.

On the other hand, an EA with a converged pool (such as the one in Fig. 4(b)) has lower chances of yielding any significantly different and higher quality solutions. This is because the converged pool handicapped the effect of the evolutionary variation operators. Consequently, in this case it is difficult to set up a good balance in exploration and exploitation.

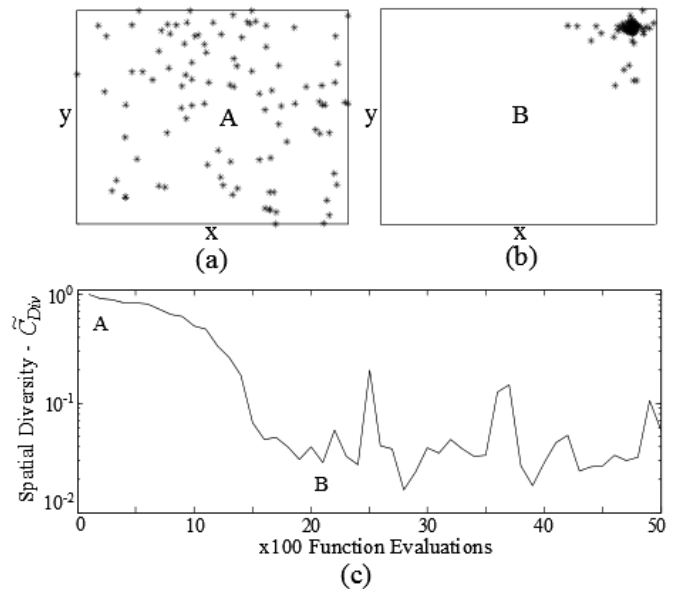


Fig. 4: A typical spatial diversity dynamics (\tilde{C}_{Div}) for a standard EC model. (a): Instantaneous 2-D view of the distribution of initial sample pool scattered all over the search space. (b): Distribution of the sample pool after several function evaluations at later stage of the evolution with the samples virtually converged. (c): Dynamics of \tilde{C}_{Div} with the regions of high level of diversity (labelled A) and low level of diversity (labelled B) marking the exploration and exploitation stages, respectively. The spatial diversity axis in plot (c) is in log scale and normalised.

⁵ The adopted mutation strategy is based on the Breeder GA (BGA) mutation algorithm (Lunacek and Whitley, 2006). It is an advanced version of Gaussian mutation.

2.) Diversity visualisation with a dual-pool EC model

In comparison to the diversity dynamics of the standard EC (Fig. 4), Fig. 5 depicts the dynamics for the proposed dual-pool EC model, described by Algorithm 3, on the same benchmark problem. The parameterisations of the dual-pool EC are as specified in Table 1. Notice that a smaller pool size, $N = 50$, is employed. In particular, Fig. 5(d) depicts the dynamics of \tilde{C}_{Div} , while Fig. 5(a-c) show the temporal interplay of the evolution and diversity pools in the dual-pool EC model.

It was found that similar to the standard EC model, the dual-pool EC enjoys an exploratory initialisation with the samples in its evolution pool P_{Evo} scouting the entire feasible search space (Fig. 5a). Then, the evolution pool gradually converges to a high quality region (see the cluster in Fig. 5b) to exploit the already learned global information of the search space. From Fig. 5(d) it is noticed that unlike with the standard EC model, the rate of convergence in this model relates more linearly with the number of function evaluations. Also, the degree to which the samples converge is considerably higher (see Fig. 5(b) and the value of \tilde{C}_{Div} at the point labelled B in Fig. 5d). As compared to a rather *weak exploitation* previously seen in the standard EC model (Fig. 4c), Fig. 5(d) indicates the ability of the dual-pool EC model to allow *deep exploitation* of the promising areas of the search space. Whilst the two algorithms share the same underlying parameterisation, the deep exploitation witnessed here could be a result of using relatively smaller sized pools (see Table 1). This was possible since the dual-pool framework is able to maintain sufficient diversity even with small sample sizes.

Furthermore, after the merger of P_{Evo} with diversity pool P_{Div} (Fig. 5c); the newly introduced diverse samples restore a full-scale spatial diversity into the previously converged evolution pool. It should be noted that while the new samples in P_{Div} draw the evolutionary search towards exploring other unexplored regions of the search space, the previously learned information is carried forward in the elite samples P_{Evo}^E inherited from the previous evolution pool (Algorithm 3, line 7). Hence, this sequence of exploration-exploitation phases guarantees continuous global searching – by preserving diversity – even when a small sized pool is utilised.

IV. EVALUATION OF THE DUAL-POOL EC ALGORITHM

This section evaluates and analyses the performance of the proposed dual-pool EC model on a set of multimodal global optimization benchmarks. The aim is to analyse the effect of effective diversity control on optimization of highly multimodal problems under limited population size and computational budget. A detailed parameterisation for the dual-pool EC model is presented in Table 1. Besides the specifications for the standard evolutionary parameters, Table 1 specifies the types of the evolutionary operators, their rates

and step sizes. It also specifies the creation mode for the dual populations.

A. Benchmark Test Cases – Key Features and Significance

The proposed dual-pool EC model is benchmarked on a set of global optimization test problems. The experiments empirically compare the performance of the dual-pool EC model with that of a standard EC model. The comparison is on the basis of the required function evaluations to attain a close approximation (within 10^{-3} accuracy level) of the true optimal solution.

The test problems considered are categorised into two major classes. The first class is a set of three traditional global optimization benchmarks consisting of: (i) Rastrigin; (ii) Schwefel; and (iii) Easom, test problems.

The Rastrigin and Schwefel functions have many local optimum solutions surrounding the global optimum, and hence they are highly multimodal. However, the Rastrigin function is symmetric and has a global convex topology (Lunacek and Whitley, 2006)⁶ whereas the Schwefel function does not.

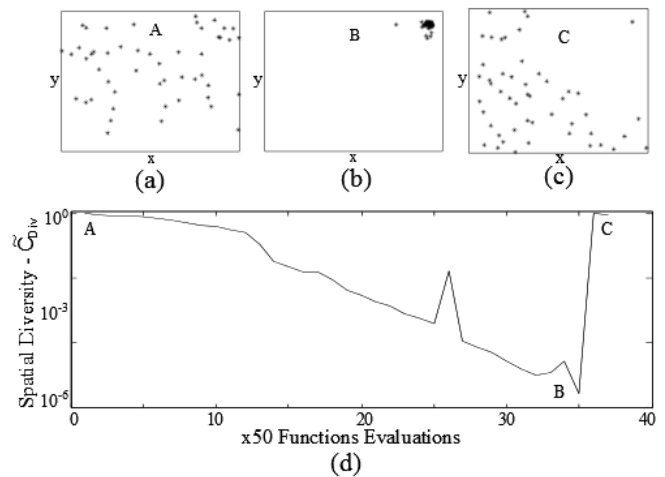


Fig. 5: A typical spatial diversity dynamics (\tilde{C}_{Div}) for the proposed Dual-Pool EC model. (a): Instantaneous 2-D view of the distribution of initial sample pool scattered all over the search space. (b): Distribution of the sample pool during the exploitation stage with the samples virtually converged. (c): Illustrates how the merger with Diversity Pool restores better coverage of the search space. (d): Diversity dynamics (\tilde{C}_{Div}) with the labels (A), (B) and (C) marking a high \tilde{C}_{Div} for exploration by the initial evolution pool, the lowest \tilde{C}_{Div} during exploitation, and a restored high level of \tilde{C}_{Div} after merger with the diversity pool, respectively. The spatial diversity axis in (d) is in log scale and normalised.

⁶ Multimodal functions having a convex global orientation are said to have global convex topology. Such functions although multimodal, appear to be

GA-easy due to the unique nature of their landscapes. They are also classified as low dispersion problems.

Table 1: Parameter Settings for the Dual-Pool EC Model.

Parameter Name	Symbol	Description/Values/Types
Population Size	N	20 to 50
Initial Population	–	SSP Heuristic initialisation
Encoding	–	Real-valued
Selection Scheme	–	Binary tournament
Evolution Pool size	P_{Evo}	N, i.e., the main population size
Diversity Pool size	P_{Div}	$(1 - k) \times N$, i.e., k% smaller than pool size N
Evolution Pool Elites	P_{Evo}^E	Only k = 5% of the evolution pool will be merged with the diversity pool after convergence of P_{Evo}
Operator type	\mathcal{C}	Crossover: Intermediate recombination operator
	\mathcal{M}	Mutation: BGA mutation operator
Crossover Probability	P_C	1.0
Mutation Probability	P_M	0.01
Recombination Parameter	α	Weighting parameter $\alpha = [0,1]$ uniform at random
Mutation Parameter	μ	Step size parameter $\mu = [0,1]$; uniform at random
Replacement Scheme	–	Generational-Elitist
Termination Criteria	E^{abs}	Absolute Error $E^{abs} \leq 10^{-4}$, or
	Max-FEs	Max. Function evaluation (10,000)

The Easom function is characterised with a single sharp peak situated in a wide plateau landscape. Easom function is quite challenging to deterministic or gradient based models because it yields no promising direction of descent/ascent. It is also popularly known as the Needle-in-Haystack (NiH) benchmark.

The second class also constitutes three test problems, namely: (i) Rastrigin2; (ii) Sphere2; and (iii) the Hybrid benchmark. These are essentially modified versions of the traditional benchmarks. They remedied some key limitations (such as separability, global convexity, symmetry, etc.) in the traditional benchmarks. Thus, they have most of the attributes of the real-world problems (Liang, 2005; Li, et al., 2008; Salomon, 1996).

In particular, the Rastrigin2 benchmark used in these experiments is a shifted and rotated version of the traditional Rastrigin function. The Sphere2 benchmark is a composition of 10 Sphere basis functions. The Hybrid benchmark is a composite of various basis functions. It consists of two basis functions from each of the Sphere, Ackley, Griewank, Rastrigin and Weierstrass benchmarks. See Table 2 in the Appendix for their detailed expressions.

B. Results

The proposed dual-pool EC model is compared with a standard EC algorithm on a set of global optimization benchmarks. The results of the evaluations of the sensitivity of these algorithms across six different pool sizes (20 to 1000) are as presented in Fig. 6; the results (detailed in the Appendix, Table 3) are averaged outcomes of 100 independent runs for statistical significance. For all the test problem types, the bar

plots in Fig. 6 show the average number of function evaluations required to reach the true optimal solutions within an absolute error of $E^{abs} = 10^{-3}$.

The horizontal dashed-lines at the top of the plots in Fig. 6 mark the limit of 10^5 function evaluations. This limit defines the maximum computational budget available for the algorithms to converge to the optimum solution. Consequently, an algorithm is considered to have converged to the true optimal solution of a given problem if and only if its bar graph has not hit the mark for the maximum function evaluation limit of 10^5 .

Furthermore, the error bars on the bar graphs (Fig. 6) represent the standard errors in the mean number of function evaluations. At the top of the bar pairs in Fig. 6, the pairs having statistically significant difference and those that have statistically insignificant difference are labelled + and –, respectively. The statistical significance results reported in Fig. 6 are based on the nonparametric Wilcoxon test.

1.) Results Analysis

Notice that the two algorithms are assessed on both *robustness* and *efficiency*; robustness of an algorithm is judged based on how often it converges to the true optimal solution within the budgeted evaluations; efficiency is rated based on the number of function calls needed to converge to the optimal solution. Thus, the efficiency is indicated by the height of the bar graphs (the lower the better).

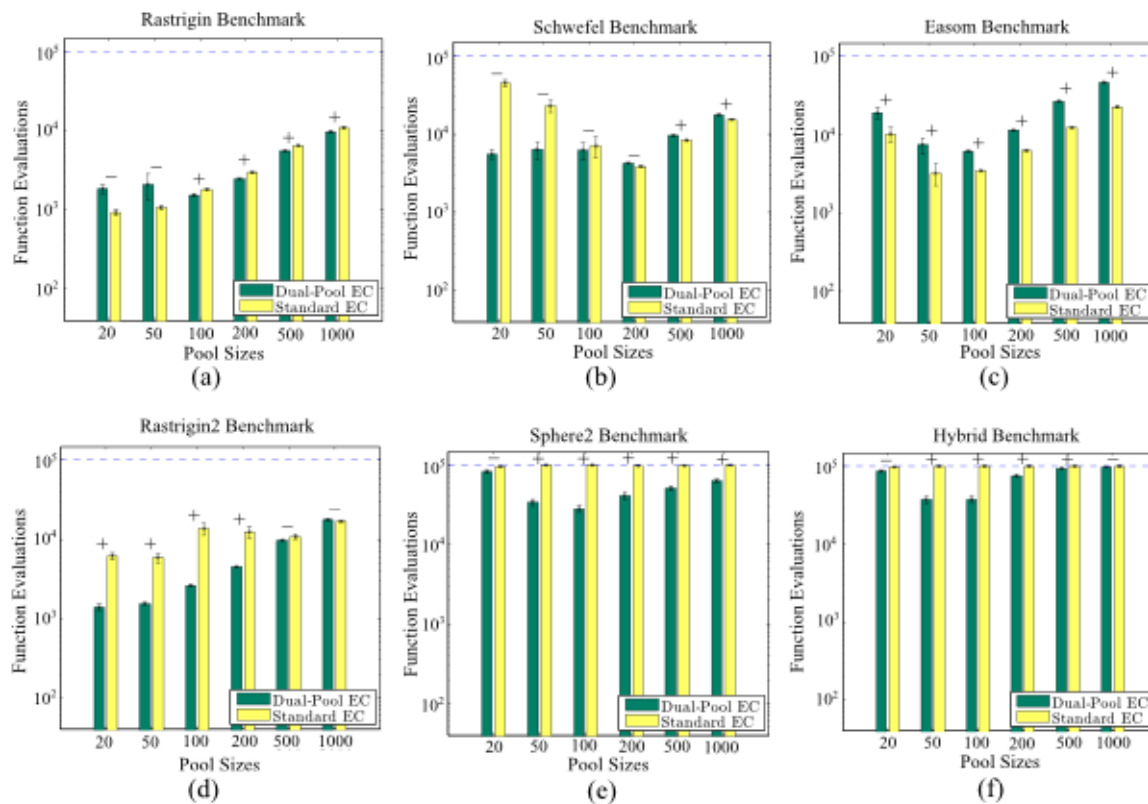


Fig. 6: Performance Comparison of the Dual-Pool EC with the Standard EC Algorithm on six global optimization benchmarks across various pool sizes. The vertical axes show the computational cost (function evaluations) in log scale. At the top of bar pairs, + symbol indicates a statistically significant difference; - symbol indicates an insignificant difference. The error bar on the bar graphs shows the standard error in the mean function evaluations. The horizontal dashed lines mark the maximum evaluation limit. All results are averages of 100 independent runs.

The simulation results (Fig. 6) are discussed in two perspectives. On one hand, we consider the results for the traditional benchmarks (Rastrigin, Schwefel and Easom), shown in Fig. 6(a to c). It was observed from Fig. 6(a to c) that both the dual-pool and standard EC algorithms have reached the required accuracy within the available (10^5) function evaluations. The two algorithms are fairly robust and equally efficient on these benchmarks. On the other hand, for the modified benchmarks shown in Fig. 6(d to f), the proposed dual-pool EC clearly outperforms the standard EC algorithm on the basis of both robustness and efficiency.

Besides the evaluation plots in Fig. 6, performance summary plots are shown in Fig. 7(a and b). Fig. 7(a) summarises the computational cost of each algorithm across all the test problems. Fig. 7(b) summarises the cost incurred by each algorithm when run with a pool of 20 to 1000 samples; its significance is to provide additional insight into the overall sensitivities of the individual algorithms to varying pool sizes.

The summary plot in Fig. 7(a) shows that for both models, the computational cost on the three traditional benchmarks is approximately around the first 10^4 function evaluations; whereas on the modified benchmarks both models needed approximately 10^5 function evaluations. Notice also that the efficiency of the proposed dual-pool EC algorithm is less efficient on the Easom benchmark (see, the point labelled (A) on Fig. 7(a)). This is not unexpected because on low

complexity problems such as the traditional benchmarks, the dual-population framework may not always translate to efficiency improvements. In fact, the central design goal is to enhance robustness on wide range of global optimization problems. Nevertheless, when summarised over all the pool sizes, the computational cost summary plot (Fig. 7(b)) revealed that the dual-pool model has always converged to the optimum solution with fewer function evaluations. This generally shows improved efficiency over the standard EC model.

Another worth noting observation from Fig. 7(b) is that both algorithms converged with fewer function evaluations when a pool size of 50 is utilised. While this indicates the true convergence efficiency for the proposed dual-pool EC, it is not the case for the standard EC in which the pool of 50 samples only converged to a local optima for the modified benchmarks (Fig. 6 (e and f)).

Overall, Fig. 7(a) reveals that the performance of both algorithms is clearly affected by the increased complexity of the test problems, i.e., from the simplest of the traditional benchmarks (Rastrigin) to the most difficult Hybrid composition benchmark.

The summary of the complete results for the six benchmark test problems, presented in Table 3, reveals that the total average computational cost for the standard EC and the proposed Dual-Pool EC algorithms is $4.01e4$ and $2.67e4$

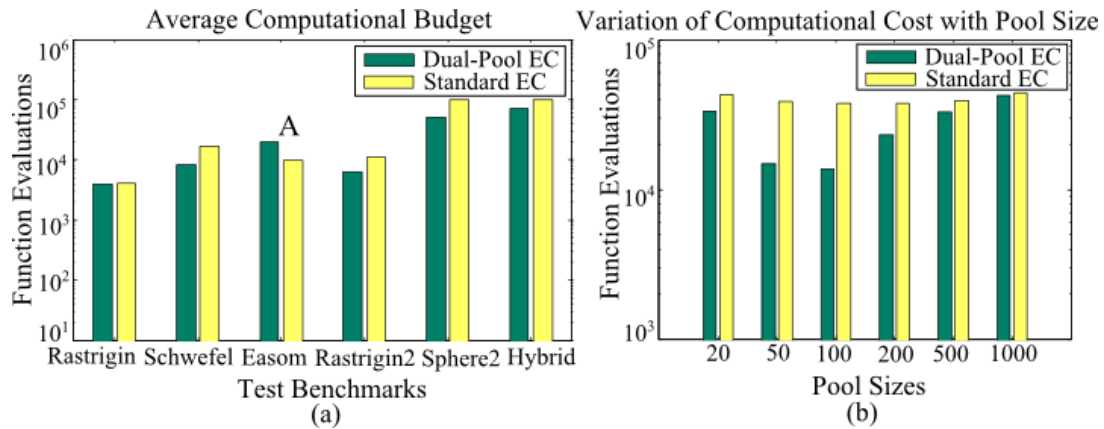


Fig. 7: Performance comparison summary for the standard EC and Dual-pool EC models on the basis of functions evaluations, the lower the better. (a): Shows the computational cost (in log scale) of the two algorithms on the individual benchmark test problems averaged over all population sizes. (b): Shows the collective cost (for all test problems) accrued by the standard EC and Dual-Pool EC models across varying pool sizes.

function evaluations respectively. This amounts to 33.34% reduction in computational cost by the proposed dual pool EC.

2.) Discussions

The earlier review on diversity control policies (Section II) revealed that use of multipopulation-based evolutionary algorithms is not entirely novel. However, the proposed criterion upon which the dual-pool EC model interacts with its separate pools suggests a new framework. Being *asynchronous*, the proposed approach harnessed the benefits of *multipool architecture* (Fig. 2) and avoided its inter-population communications difficulties.

Equally, the observed improvements in diversity control exhibited by the dual-pool EC model (Fig. 5(d)) could partly be credited to the proposed SSP heuristic initialisation (Section III). The SSP minimized the stochastic variability in the final solution by ensuring optimum uniformity in the distribution of the randomly created samples. This agrees with a number of investigations (Tometzki and Engell, 2011; Morrison, 2003) in which heuristic initialisations improved the statistical significance of the final results in EAs by minimising their stochastic variability.

In particular, two important points are noted. First, simulation results (Fig. 7) have shown that the dual-pool EC algorithm converges to the optimal solutions on both categories of benchmarks with only small to medium pool sizes. This justifies its ability to *maintain* and *restore* useful diversity into its search pool. This validates the efficacy of its diversity dynamics previously observed in Fig. 5(d). Second, since working with small sized populations often translates to reduced computational cost, the ability of the proposed model to sustain evolutionary search with small to medium sized pools links its potentials in improving *convergence efficiency*.

The above two points corroborate the key hypothesis in this work, which states that a good explorative-exploitative model crucially improves robustness in global search without compromising its efficiency.

V. CONCLUSION

This paper presented a new approach for diversity control in evolutionary computation (EC) algorithms. It addressed the challenges associated with balancing the exploration and exploitation tradeoff by using a multipopulation strategy with a heuristic initialisation. The insights obtained from the investigations in this paper have paved the way for the development of the newly proposed dual-pool EC architecture. The search space partitioning heuristic initialisation and the diversity control measures proposed in this paper facilitated effective exploration and exploitation in optimization of various global benchmark problems.

In particular, the experimental results have shown that the proposed algorithm solves problems from both the *traditional* and *modified* global optimization benchmarks with pool sizes of only 50 to 100 samples. This feature is vital for minimising the cost of solving computationally expensive problems. Specifically, the proposed method successfully yields a 33.34% reduction in the computational cost of optimizing the benchmark problems as compared to a standard EA. This outcome justifies the impact of effective diversity control on robustness and convergence efficiency of optimization methodologies.

APPENDIX: TEST CASE STUDIES AND RESULT SUMMARY

Table 2 outlines the formulations, domain specifications and the respective universal tags for the global optimization benchmarks used in this paper. Table 3 presents the complete numerical results of comparing the proposed dual-pool EC against the standard EC algorithm.

Table 2: Global Benchmark (Basic) Functions.

Name	Benchmark Function	Range
Rastrigin	$f_1(x) = 10 \cdot n + \sum_{i=1}^n (x_i^2 - 10 \cdot \cos(2\pi x_i)); n = 100.$	[-5.12,5.12]
Schwefel	$f_2(x) = \sum_{i=1}^n x_i \sin(\sqrt{ x_i }); n = 2.$	[-5.0,5.0]
Easom	$f_3(x) = \cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	[-100,100]
Sphere	$f_4(x) = \sum_{i=1}^n x_i^2; n = 2.$	[-100,100]
Weierstrass	$f_5(x) = \sum_{i=1}^n (\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (x_i + 0.))]) - n \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)];$ $a = 0.5, b = 3, k_{max} = 20, n = 2.$	[-0.5,0.5]
Griewank	$f_6(x) = \frac{1}{400} \sum_{i=1}^n (x_i)^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1; n = 2.$	[-100,100]
Ackley	$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e;$ $n = 2.$	[-32,32]

Table 3: Computational cost in terms of function evaluations required by the Dual-pool and Standard EC algorithms to converge to a 0.1% accuracy level of the global optimal solution for the six different global optimization benchmarks. The table shows sensitivities of the two algorithms to varying population sizes. All results are averages of 100 independent runs.

Pool sizes	Traditional Benchmark Test Problems						Modified Benchmark Test Problems					
	Rastrigin		Schwefel		Easom		Rastrigin2		Sphere2		Hybrid	
	DP-EC	Std-EC	DP-EC	Std-EC	DP-EC	Std-EC	DP-EC	Std-EC	DP-EC	Std-EC	DP-EC	Std-EC
20	1.86e3	0.93e3	5.54e3	4.43e4	1.93e4	1.03e4	1.41e3	6.17e3	8.35e4	9.71e4	8.74e4	9.82e4
50	2.07e3	1.07e3	6.29e3	2.29e4	7.44e3	3.27e3	1.55e3	5.83e3	3.42e4	1.00e5	3.72e4	1.00e5
100	1.54e3	1.81e3	6.17e3	6.80e3	6.24e3	3.50e3	2.60e3	1.37e4	2.84e4	9.99e4	3.77e4	1.00e5
200	2.47e3	2.98e3	4.23e3	3.85e3	1.15e4	6.27e3	4.48e3	1.25e4	4.11e4	9.95e4	7.63e4	9.99e4
500	5.59e3	6.45e3	9.46e3	8.27e3	2.69e4	1.24e4	9.69e3	1.08e4	5.13e4	9.88e4	9.42e4	1.00e5
1000	9.84e3	1.10e4	1.75e4	1.51e4	4.69e4	2.28e4	1.78e4	1.70e4	6.36e4	1.00e5	9.87e4	1.00e5
Avg. Cost	3.90e3	4.04e3	8.20e3	1.69e4	1.97e4	9.76e3	6.26e3	1.10e4	5.04e4	9.92e4	7.19e4	9.97e4

Notation: DP-EC = Dual-Pool EC algorithm, Std-EC = Standard EC algorithm, Avg. Cost = Average computational cost in terms of number of function evaluations. The **bold face** items indicate where an algorithm outperforms its counterpart.

REFERENCES

- Alba, E. and Tomassini, M. (2002).** Parallelism and Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, 6: 443-462.
- Bashir, H. A. and Neville, R. S. (2012).** Convergence Measurement in Evolutionary Computation Using Price's Theorem. IEEE Congress on Evolutionary Computation (CEC), 1961-1968.
- Bassett, J. K.; M. A. Potter and K. A. De Jong. (2004).** Looking Under the EA Hood with Price's Equation. Genetic and Evolutionary Computation – GECCO 2004. Springer Berlin / Heidelberg, 3102: 914-922.
- Blum, C.; M. Aguilera; A. Roli and M. Sampels. (2008).** Hybrid Metaheuristics: An Introduction. Hybrid Metaheuristics. Springer Berlin / Heidelberg, 114: 1-30.
- Blum, C.; J. Puchinger; G. Raidl and A. Roli. (2011).** Hybrid Metaheuristics in Combinatorial Optimization: A Survey. Applied Soft Computing, 11:4135-4151.
- Branke, J.; A. Eiben; T. Back; M. Schoenauer and H. P. Schwefel. (1998).** Creating Robust Solutions by Means of Evolutionary Algorithms. Parallel Problem Solving from Nature PPSN V. Springer Berlin / Heidelberg, 1498: 119-128.
- Branke, J.; T. Kau; I. Schmidt and H. Schmeck. (2000).** A Multi-Population Approach to Dynamic Optimization Problems. In: Parmee I.C. (eds) Evolutionary Design and Manufacture. Springer, London, UK. 299-308.
- Cheng, S.; Y. Shi; Q. Qin; Q. Zhang and R. Bai. (2015).** Population Diversity Maintenance in Brain Storm Optimization Algorithm. Journal of Artificial Intelligence and Soft Computing Research (JAISCR). DE GRUYTER. 4(2):83-97.
- Cobb, H. G. and Grefenstette, J. (1993).** Genetic Algorithms for Tracking Changing Environments. Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann. 523-530, San Francisco, United States.
- Corriveau, G.; R. Guilbault; A. Tahan and R. Sabourin. (2012).** Review and Study of Genotypic Diversity Measures for Real-Coded Representations. IEEE Transactions on Evolutionary Computation, 16: 695-710.
- Dang, M. P.; T. P. Dao; N. L. Chau and H. G. Le. (2019).** Effective Hybrid Algorithm of Taguchi Method, FEM, RSM, and Teaching Learning-Based Optimization for Multiobjective Optimization Design of a Compliant Rotary Positioning Stage for Nanoindentation Tester. Mathematical Problems in Engineering, 2019. Hindawi.
- De Jong, K. (1975).** An Analysis of the Behavior of a Class of Genetic Adaptive Systems. Doctoral Dissertation: University of Michigan.
- de-Melo, V. and Botazzo-Delbem, A. C. (2012).** Investigating Smart Sampling as a Population Initialization

Method for Differential Evolution in Continuous Problems. *Information Sciences*, 193 (1): 36-53.

Eiben, A. E.; R. Hinterding and Z. Michalewicz. (1999). Parameter Control in Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 3:124-141.

Eshelman, L. and Schaffer, D. (1991). Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann. 115-122, San Francisco, United States.

Fleming, P. J. and Purshouse, R. C. (2002). Evolutionary Algorithms in Control Systems Engineering: A Survey. *Control Engineering Practice*, 10:1223-1241.

Fogel, D. B. (1997). Evolutionary Computation: A New Transactions. *IEEE Transactions on Evolutionary Computation*, 1:1-2.

Frank, S. (1997). The Price Equation, Fisher's Fundamental Theorem, Kin Selection, and Causal Analysis. *Evolution*, 51:1712-1729.

Garcia, S.; D. Molina; M. Lozano and F. Herrera. (2009). A study on the Use of Non-parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, 15: 617-644.

Goldberg, D. E. and Holland, J. H. (1988). Genetic Algorithms and Machine Learning. *Machine Learning*, 3:95-99.

Gong, G.; Q. Deng; R. Chiong; X. Gong and H. Huang. (2019). An effective memetic algorithm for multi-objective job-shop scheduling, *Knowledge-Based Systems*, 182. Elsevier.

Greenhalgh, D. and Marshall, S. (2000). Convergence Criteria for Genetic Algorithms. *SIAM J. Comput.*, 30:269-282.

Herrera, F. and Lozano, M. (1996). Adaptation of Genetic Algorithm Parameters Based on Fuzzy Logic Controllers. *Genetic Algorithms and Soft Computing*: Springer-Verlag, Spain.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Michigan: The University of Michigan Press.

Hollander, M. and Wolfe, D. (1999). *Nonparametric Statistical Methods*, 2nd Edition: Wiley-Interscience.

Jih-Yiing, L. and Ying-Ping, C. (2011). Analysis on the Collaboration Between Global Search and Local Search in Memetic Computation. *IEEE Transactions on Evolutionary Computation*, 15: 608-623.

Joines, J. and Kay, M. (2003). Utilizing Hybrid Genetic Algorithms Evolutionary Optimization. *Springer US*, 48: 199-228.

Li, C.; S. Yang; T. T. Nguyen; E. L. Yu; X. Yao; Y. Jin; H. G. Beyer and P. N. Suganthan. (2008). Benchmark Generator for CEC'2009 Competition on Dynamic Optimization. *Computer Science*, University of Leicester, U.K.

Liang, J. J.; P. N. Suganthan and K. Deb. (2005). Novel Composition Test Functions for Numerical Global

Optimization. *IEEE Proceedings on Swarm Intelligence Symposium, SIS 2005*. 68-75.

Liu, J.; H. A. Abbass and K. C. Tan. (2018). *Evolutionary Computation and Complex Network. Technology & Engineering*. Springer International Publishing, Spain.

Lozano, M.; F. Herrera and J. Cano. (2008). Replacement Strategies to Preserve Useful Diversity in Steady-State Genetic Algorithms. *Inf. Sci.*, 178:4421-4433.

Lunacek, M. and Whitley, D. (2006). The Dispersion Metric and the CMA Evolution Strategy. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, Seattle, Washington, 477-484, USA: ACM.

Maaranen, H.; K. Miettinen and M. M. Makela. (2004). Quasi-random Initial Population for Genetic Algorithms. *Comput. Math. Appl.*, 47:1885-1895.

McGinley, B.; J. Maher; C. O'Riordan and F. Morgan. (2011). Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection. *IEEE Transactions on Evolutionary Computation*, 15:692-714.

Michalewicz, Z. (1994). *Genetic algorithms + data structures = evolution programs (2nd, extended ed.)*: Springer-Verlag New York, Inc.

Molina, D.; M. Lozano; C. G. Martinez and F. Herrera. (2010). Memetic algorithms for continuous optimisation based on local search chains. *Evol. Comput.*, 18:27-63.

Morrison, R. W. (2003). Dispersion-based Population Initialization. *Proceedings of the 2003 International Conference on Genetic and Evolutionary Computation: Part I Chicago, IL, USA: Springer-Verlag*.

Moscato, P. (1999). *Memetic Algorithms: A Short Introduction*. New Ideas in Optimization: McGraw-Hill Ltd., UK, 219-234.

Muhlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive Models for the Breeder Genetic Algorithm I - Continuous Parameter Optimization. *Evol. Comput.*, 1: 25-49.

Pengfei, G.; W. Xuezhi and H. Yingshi. (2010). The Enhanced Genetic Algorithms for the Optimization Design. 3rd International Conference on Biomedical Engineering and Informatics (BMEI), Yantai, China. 2990-2994. *IEEE*.

Potter, M. A.; J. K. Bassett and K. A. De Jong. (2003). Visualizing Evolvability with Price's Equation. *Congress on Evolutionary Computation, CEC '03*, 4: 2785-2790.

Rahnamayan, S.; H. R. Tizhoosh and M. M. Salama. (2007). A Novel Population Initialization Method for Accelerating Evolutionary Algorithms. *Computers & Mathematics with Applications*, 53:1605-1614.

Reeves, C. R. (1993). *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publishing, Oxford, UK.

Salomon, R. (1996). Re-evaluating Genetic Algorithm Performance under Coordinate Rotation of Benchmark Functions. *A Survey of Some Theoretical and Practical Aspects of Genetic Algorithms*. *BioSystems*, 39: 263-278.

Shetti, B. (2019). Curse of Dimensionality. Available online at: <https://builtin.com/data-science/curse-dimensionality>. Accessed on September 25, 2019.

- Shir, O. M. (2012).** Niching in Evolutionary Algorithms. Handbook of Natural Computing. Springer, Berlin, Heidelberg.
- Segura, C.; A. Hernández-Aguirre; F. Luna and E. Alba. (2017).** Improving Diversity in Evolutionary Algorithms: New Best Solutions for Frequency Assignment. IEEE Transactions on Evolutionary Computation, 21 (4): 539 - 553.
- Sharma, H.; J. B. Chand and K. V. Arya. (2013).** Diversity Measures in Artificial Bee Colony. Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012). Advances in Intelligent Systems and Computing. 201 (1): 299-314.
- Squillero, G. and Tonda, A. (2016).** Promoting Diversity in Evolutionary Algorithms: An Updated Bibliography. Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (GECCO '16). Denver, Colorado. 943-944. ACM. USA.
- Taejin, P. and Kwang, R. R. (2010).** A Dual-Population Genetic Algorithm for Adaptive Diversity Control. IEEE Transactions on Evolutionary Computation, 14:865-884.
- Tenne, Y. and Armfield, S. W. (2005).** Efficiently Minimizing Expensive Cost Functions with a Hybrid Evolutionary Algorithm using Clustering and a Derivative-free Optimizer: Preliminary Results. IEEE Congress on Evolutionary Computation, 3:1937-1944.
- Thierens, D. (1998).** Selection Schemes, Elitist Recombination, and Selection Intensity. Proceedings of 7th International Conference on Genetic Algorithms San Francisco, CA: Morgan Kaufmann, 152-159.
- Tometzki, T. and Engell, S. (2011).** Systematic Initialization Techniques for Hybrid Evolutionary Algorithms for Solving Two-Stage Stochastic Mixed-Integer Programs. IEEE Transactions on Evolutionary Computation, 15:196-214.
- Tsutsui, S.; Y. Fujimoto and A. Ghosh. (1997).** Forking Genetic Algorithms: GAs with Search Space Division Schemes. Evol. Comput., 5: 61-80.
- Ursem, R. K. (1999).** Multinational Evolutionary Algorithms. Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99. 3:1640.
- Wen, T. L.; W. S. Xiao; Q. H. Yong; F. L. Shu and Z. Jiang. (2010).** A Hybrid Optimization Algorithm and Its Application for Conformal Array Pattern Synthesis. IEEE Transactions on Antennas and Propagation, 58:3401-3406.
- Xianshun, C.; O. Yew-Soon; L. Meng-Hiot and K. C. Tan. (2011).** A Multi-Facet Survey on Memetic Computation. IEEE Transactions on Evolutionary Computation, 15: 591-607.
- Xu, W.; C. Chen; S. Ding and P. M. Pardalos. (2019).** A bi-objective dynamic collaborative task assignment under uncertainty using modified MOEA/D with heuristic initialization. Expert Systems with Applications, Elsevier, 140 (1): 112-144.
- Yi, J.; W. Ling and C. Li. (2008).** A Hybrid Dynamical Evolutionary Algorithm for Classification Rule Discovery. Second International Symposium on Intelligent Information Technology Application, IITA '08, 76-79.