Exploration and Exploitation in Evolutionary Algorithms: A Survey

3 authors, including:

Shih-hsi Liu
California State University, Fresno
45 PUBLICATIONS **929** CITATIONS

SEE PROFILE

Marjan Mernik
University of Maribor
244 PUBLICATIONS **5,814** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Evolutionary Computation View project

Exploration and Exploitation in Evolutionary Algorithms: A Survey

MATEJ ČREPINŠEK, University of Maribor
SHIH-HSI LIU, California State University, Fresno
MARJAN MERNIK, University of Maribor

"Exploration and exploitation are the two cornerstones of problem solving by search." For more than a decade, Eiben and Schippers' advocacy for balancing between these two antagonistic cornerstones still greatly influences the research directions of evolutionary algorithms (EAs) [1998]. This article revisits nearly 100 existing works and surveys how such works have answered the advocacy. The article introduces a fresh treatment that classifies and discusses existing work within three rational aspects: (1) what and how EA components contribute to exploration and exploitation; (2) when and how exploration and exploitation are controlled; and (3) how balance between exploration and exploitation is achieved. With a more comprehensive and systematic understanding of exploration and exploitation, more research in this direction may be motivated and refined.

1. INTRODUCTION

Every search algorithm needs to address the exploration and exploitation of a search space. *Exploration* is the process of visiting entirely new regions of a search space, whilst *exploitation* is the process of visiting those regions of a search space within the neighborhood of previously visited points. In order to be successful, a search algorithm needs to establish a good ratio between exploration and exploitation. In this respect, evolutionary algorithms (EAs) [De Jong 2002; Eiben and Smith 2008], such as genetic algorithms (GAs) [Michalewicz 1996; Goldberg 2008], evolutionary strategies (ES) [Bäck 1996], evolutionary programming (EP) [Fogel 1999], and genetic programming (GP) [Koza 1992], to name the more well-known instances, are no exception. Herrera and Lozano [1996] emphasized this by saying, "The genetic algorithm behaviour is determined by the exploitation and exploration relationship kept throughout the run." Many researchers believe that EAs are effective because of their good ratio

---

between exploration and exploitation. Michalewicz [1996] stated, "Genetic Algorithms are a class of general purpose (domain independent) search methods which strike a remarkable balance between exploration and exploitation of the search space." In spite of the fact that exploration and exploitation are fundamental concepts, they are often misunderstood by EA practitioners or even EA researchers.

Eiben and Schippers [1998] provided an early discussion on evolutionary exploration and exploitation. Their work raised several questions and demonstrated the need for more research. The aim of their work was to question the common belief that exploitation in EAs is done by selection, whilst exploration is performed by search operators (e.g., mutation and crossover). Eiben and Schippers also reviewed ten papers in order to consider existing views, at the time of their writing, about exploration and exploitation in EAs. They found that there was no generally accepted agreement on this topic. Yet, it seems that their paper, cited by only 32 papers[1], has been mainly overlooked. Perhaps the problem of how to control and measure exploration and exploitation is still too difficult and/or implicit.

The main objective of this article is to remedy this situation by providing a more complete treatment of evolutionary exploration and exploitation. Hence, common misunderstandings and incorrect beliefs about exploration and exploitation in EAs could be gradually diminished. More importantly, better EAs could be developed by developing a better understanding of exploration and exploitation. For example, a better understanding could clarify why one selection mechanism within a particular setting is better than another, or which crossover/mutation operator to choose. Furthermore, differences among EAs might be better understood. Bäck and Schwefel [1993] conducted an early discussion on differences among GAs, ESs, and EP. They were astonished by the differences that drive evolution in EAs:

> "It is a remarkable fact that each algorithm emphasizes different features as being most important for a successful evolution process. . . . Both ESs and EP concentrate on mutation as the main search operator, while the role of (pure random) mutation in canonical GAs is usually seen to be of secondary (if any) importance. On the other hand, recombination plays a major role in canonical GAs, is missing completely in EP, and is urgently necessary for use in connection to self-adaptation in ESs. One of the characteristics of EP is the strict denial of recombination being important for the search. Finally, both canonical GAs and EP emphasize on a necessarily probabilistic selection mechanism, while from the ESs point of view selection is completely deterministic without any evidence for the necessity of incorporating probabilistic rules. In contrast, both ESs and EP definitely exclude some individuals from being selected for reproduction, i.e., they use extinctive selection mechanisms, while canonical GAs generally assign a nonzero selection probability to each individual, which we term a preservative selection mechanism."

These differences among EAs can be difficult to understand, not only by beginners within the field of EAs but also by many experienced EA practitioners, without the notion of balance between exploration and exploitation. Differences not only appear among EA instances but also inside the same type of EA. For example, diametrical recommendations might confuse many if the rationale for including such a feature is unexplained or misunderstood. Similarly, how to explain that some successful GAs prevent inbreeding (e.g., by incest prevention [Eshelman and Schaffer 1991]) whilst, on the other hand, the others (e.g., shifting balance GA [Oppacher and Wineberg 1999]) promote inbreeding? How do we understand such dichotomies? Actually, it is all about

---

[1]Search performed on July 12, 2011, using SCOPUS.

achieving a proper ratio between exploration and exploitation. No evolutionary process should be studied in isolation, and balance should be achieved synergistically. For example, the aforementioned inbreeding dichotomy may be explained in the following manner. Instead of trying to enhance variety by penalizing similarities (e.g., incest prevention), you could use inbreeding (e.g., shifting balance GA) that may cause genetic drift if a population is small enough (subpopulations). This will make it possible to surpass valleys and promote exploration. A stabilizing effect is achieved when such subpopulations are integrated into a large population with high selection pressure, achieving equilibrium between exploration and exploitation. Another example may be found in the recombination of (dis)similar parents, where Horn et al. [1994] concluded that "information from very different types of trade-offs could be combined to yield other kinds of good trade-offs," whilst Ishibuchi et al.'s empirical study [2008] on NSGA-II showed that similar parents improved the diversity of solution without degrading their convergence.

Yet another example: inexperienced EA users might find it difficult to understand that the same EA behaves equally, in terms of the best solution found, despite very different control parameter settings. Smit and Eiben [2009] showed that by using algorithmic parameter tuning to solve a simple optimization problem, the Rastrigin function performed equally well using the following settings: population size 14 vs. 448, tournament proportion 0.8782 vs. 0.3503, and generational gap 0.8443 vs. 0.0215. The proportion of the population size that is replaced by each generation was much larger than in the case of smaller populations; there is also a distinctive difference in the selection pressure. We are convinced that all the aforementioned differences are much easier to understand in terms of exploring and exploiting the search space, especially in understanding the balance between exploration and exploitation. The fact that until now exploration and exploitation have only been implicitly defined in EAs comes as a big surprise.

From existing papers on survey/taxonomy in metaheuristics, Blum and Roli [2003] provided different classifications of metaheuristics (nature-inspired vs. non-nature inspired, population-based vs. single point search, dynamic vs. static objective function, one vs. various neighborhood structures, and memory usage vs. non-memory usage), described inner workings of different metaheuristics (e.g., simulated annealing, tabu search, evolutionary algorithms, ant colony optimization), and provided a unified view on diversification and intensification. (Exploration and exploitation by Blum and Roli [2003] refers to rather short-term strategies tied to randomness, while diversification and intensification refer to medium- to long-term strategies based on the usage of memory). Another classification of hybrid metaheuristics was introduced by Talbi [2002]. Yet, diversification and intensification were excluded as the criteria for taxonomy. Liu et al. [2009] also classified EAs into uniprocess- and multiprocess-driven approaches regarding how the balance between exploration and exploitation is achieved. In order to make the classification [Liu et al. 2009] more thorough, another objective of this article is to provide more comprehensive literature studies and new classifications for existing approaches.

This article is organized as follows. Section 2 presents the reviews of how exploration and exploitation can be achieved in EAs. We reemphasize that delimitation of the two cornerstones is difficult and as yet unachievable within the existing work. Section 3 presents the discussions of when and how to control exploration and exploitation. We specifically emphasize diversity measurements because of their core role in delimiting exploration from exploitation. Section 4 provides a comprehensive review of existing work done in the field of balancing exploration and exploitation in terms of diversity-driven and other direct criteria. Section 5 concludes the article by indicating future research directions.

2. ACHIEVING EXPLORATION AND EXPLOITATION IN EAS

The question of how exploration and exploitation are achieved in EAs may seem trivial, but this is not so. For example, the discussion on this topic where Eiben and Schippers [1998] noticed that a common opinion about EAs is that search space is explored by crossover/mutation operators, whilst exploitation is done by selection, is at least questionable. Moreover, ultimately Eiben and Schippers concluded that there was no generally accepted perception about exploration and exploitation in EAs. Another important conclusion derived from Eiben and Schippers [1998] is that more intensive research is needed for a deeper understanding of the fundamentals of evolutionary search processes. Overall, in many papers, oversimplified views are perceived on this subject. For example, Wong et al. [2003] wrote: "In order to optimize the efficiency and effectiveness, Genetic Algorithms (GAs) must maintain a balance between the exploitation of beneficial aspects of existing solutions (by crossover) in order to improve them, and the exploration of the solution space (by mutation) so as to increase the probability of finding the optimal solution. This balance is determined by the crossover rate and the mutation rate."

In this article, more complete treatment on evolutionary exploration and exploitation is given, along with those selection processes and other factors that were ignored in the aforementioned quote. Selection drives the search toward the regions of the best individuals. Hence, it can be mainly seen as an exploitation operator. However, Bäck [1994] showed that selection processes can control the level of exploration or exploitation by varying selection pressure. Higher selection pressure pushes the search towards more exploitation, and lower selection pressure urges the search towards more exploration. Maintaining accurate selection pressure and hence a balance between exploration and exploitation is needed for many optimization problems [Goldberg and Deb 1991]. Additionally, a mutation operator randomly modifies individuals with a given probability and thus increases the structural diversity of a population. From this point of view, a mutation operator is more of an exploration operator. Such an operator facilitates the recovery of genetic diversity lost during the selection phase and explores new solutions. Conversely, a mutation operator can also be seen as an exploitation operator because most of the genetic materials are preserved. The role of mutation in different EAs is slightly different, too. In ES, mutation is more of an exploration operator, whilst in GAs, mutation is more of an exploitation operator if the locality property [Galván-López et al. 2010], described in Section 3.1, holds. A crossover operator combines two or more parents to generate the possibility of better offspring. Such a combination can be derived from the idea that an exchange of information between good individuals will generate an even better offspring. From this perspective, a crossover operator is more of an exploitation operator. However, a good crossover operator should also generate individuals within the exploration zone. The potential number of ways in which an operator can generate a new individual is called its exploratory power [De Jong and Spears 1992]. In many cases, it is difficult to predict if newly generated individuals produced by a crossover and/or mutation operator will fall into the exploration or exploitation zones. It is crucial to understand that we can never call a crossover/mutation operator a pure exploration/exploitaton operator. At best, we can hope that most generated individuals will fall into that particular zone.

As can be seen from the preceding discussion, exploration and exploitation in EAs is achieved by selection, mutation, and crossover. But it is difficult to distinguish between exploration from exploitation in these processes. The line between exploration and exploitation is blurred. Furthermore, these are not the only factors that have an impact on exploration and exploitation. Population size and the representation of individuals have important impacts, too. Directing an evolution process towards exploration or

exploitation is also possible by population resizing [Harik and Lobo 1999; Smith and Smuda 1995]. With a larger population size, the search space is explored more than with a smaller population size. This is an easier way to maintain the diversity but is often an unsatisfactory solution. Without proper handling, even larger populations can converge and waste processing time. Moreover, population size also influences other operators. For example, Spears [1995] showed that uniform crossover outperforms two-point crossover regarding a small population on a particular problem, but just the opposite is true with a large population. It is almost impossible, or at least less possible, to study a particular feature in isolation. Another tricky point is representation. Mutation and crossover operators that influence exploration and exploitation mostly depend on representation of individuals. Hence, representation really matters, and only in a few cases was representation independence achieved (e.g., geometric crossovers/mutations derived from the theories of topology and geometry [Moraglio et al. 2007]). It is important to know at which levels mutation/crossover operators work: at the individual, sub-individual, or gene level [Smith and Fogarty 1997]. Indeed, many researchers point out that representation is a critical factor in the success of an EA (e.g., Ishibuchi et al. [2010b] introduced a representation-dependent nongeometric binary crossover to further improve diversity without degrading convergence in evolutionary multiobjective optimization (EMO)). But, the relationship between an individual's representation and the balance between exploration and exploitation is still not well understood, and more research is needed.

The next important question is how the balance between exploration and exploitation is achieved in an EA. What is observed from the previous discussion is that the selection process and variation operators (e.g., crossovers and mutations) are able to somehow establish and, in most cases, find a good ratio between exploration and exploitation of the search space. Until now, achieving balance between exploration and exploitation has been managed by proper control-parameter settings. If crossover and mutation rates are very high, much of the space will be explored, but there is a high probability of missing good solutions and of failing to exploit existing solutions. In such a case, EAs head toward random search. If crossover and mutation rates are low, the search space is unexplored. In such a case, EAs are closer to hill climbing algorithms. Moreover, crossover and mutation operators are usually into interactions, and optimizing both parameters (crossover and mutation rates) cannot be done independently. In such cases, all combinations of crossover and mutation rates should be experimented. Which control-parameter settings are most likely to produce the best results is a question that every EA developer/user has to face. In the EA community the following approaches have been tried [Lobo et al. 2007].

(1) Trial-and-error approach, which is a time-consuming and tedious method, usually performed in an ad-hoc manner.
(2) Following general guidelines (e.g., [De Jong 1975; Harik and Lobo 1999; Schaffer et al. 1989]), which are often inapplicable for specific cases. It is often found that recommended parameter settings from literature do not lead to the best solutions for particular cases (e.g., [Smit and Eiben 2009; Zielinski et al. 2009]).
(3) Using parameter-less EA [Harik and Lobo 1999] or GAs without parameters [Bäck et al. 2000], which are robust but mostly less-efficient approaches.
(4) Using experiences from previous similar applications, which is inapplicable when such experiences do not exist.
(5) Identifying the features of fitness landscapes by a classifier in order to propose good control parameter settings [Bogon et al. 2011].
(6) Statistical analysis of control parameter interactions and their effect on algorithms' performances [Czarn et al. 2004].

(7) Using mathematical models, which is important but often too simple to be realistic or too difficult to be understood by ordinary users.

(8) Algorithmic parameter tuning, where the search for the best control parameters is seen as an optimization problem that can be solved using specific methods. The more well-known examples are the meta-evolutionary approach (evolution of evolution) [Grefenstette 1986], racing algorithm [Birattari et al. 2002], sequential parameter optimization [Bartz-Beielstein et al. 2005], meta-estimation of distribution algorithm [Nannen and Eiben 2006], hyper-heuristics [Ochoa et al. 2008], multiobjective optimization [Dréo 2009], and reinforcement learning [Montero and Riff 2011]. Work in this category has been recently surveyed and categorized into sampling methods, model-based methods, screening methods, and meta-evolutionary algorithms [Eiben and Smit 2011].

It is well known among EA researchers that the control parameter setting is problem-dependent. The control parameter setting which produces the best result for a particular problem might not lead to the best result for another problem [Smit and Eiben 2009; Zielinski et al. 2009]. From the exploration/exploitation perspective, this means that different problems require different amounts of exploration and exploitation. This implies that a good ratio between exploration and exploitation, and hence a proper or good balance, is problem-dependent too. For example, for unimodal function optimization, less exploration is probably needed than for multimodal function optimization. Therefore, the goal of any search algorithm is to inherently find a good balance between exploration and exploitation for different problems. To make the problem even more difficult, different values for control parameters might be optimal at different stages of an evolution process, and hence different amounts of exploration and exploitation are needed during the evolution process. For example, in the early stages, a larger population is needed than in the later stages when fine tuning of suboptimal solutions is done. Eiben et al. [1999] provided an excellent overview of this problem, where parameter tuning and parameter control were distinguished. For parameter tuning, parameters are defined before a run (also called offline approach) and do not change during the evolution process. All the aforementioned approaches (e.g., trial-and-error, algorithmic parameter tuning) are examples of parameter tuning. Conversely, the parameters in the latter approach are changed during the run (also called the online approach). Eiben et al. [1999] classified the methods on how parameters are controlled into deterministic, adaptive, and self-adaptive categories.

(1) The deterministic category adjusts the parameters by deterministic rules.

(2) The adaptive category utilizes the feedback of an evolution process to control the directions and magnitudes of the parameters.

(3) The self-adaptive category encodes parameters into individuals and undergoes mutation and recombination (simultaneous evolution of evolution).

Deterministic and adaptive parameter control can also be expressed in algorithmic ways by using a domain-specific language [Mernik et al. 2005] for programming parameter control [Liu et al. 2004]. Note that *classic* meta-evolutionary approaches are mainly classified under parameter tuning (offline approach), because control parameters, evolved at the meta level, remain unchanged during an evolution process at the base level. Recently, some hybridization between parameter tuning and parameter control has been proposed (e.g., meta GA combined with an adaptation strategy for the GA control parameters [Fernandez-Prieto et al. 2011]). As for the meta-EP mentioned in Bäck and Schwefel [1993], the parameters are encoded and evolved along with the process and hence are self-adaptive. We also utilize Eiben et al.'s classification [1999] for classifying the explicit control of exploration and exploitation.

One common belief is that EAs should start with exploration and then gradually change into exploitation. Such a policy can be easily described with deterministic approaches, where the mutation rate decreases along with the evolution (e.g., [Fogarty 1989]). This is generally correct, but such a policy tends to face difficulties when solving certain problems (e.g., multimodal functions with many optima [Liu et al. 2009; Yao et al. 1999] or when dynamic environments are evolved), since premature takeover of exploitation over exploration occurs. Hence, more promising (self-)adaptive approaches have been proposed. They range from utilizing simple information available from current processes, such as fitness values (e.g., [Harik and Lobo 1999]), the number of generations with no improvements (e.g., [Eiben et al. 2004]), diversity (e.g., [Ursem 2002]), and entropy (e.g., [Rosca 1995]), to more advanced techniques, such as adaptive genetic operators and selection [Herrera and Lozano 1996], adaptive representation [Whitley et al. 1991], and adaptive fitness functions [Majig and Fukushima 2008]. Yet, we envisage that new and better (self-)adaptive approaches will be proposed in the future.

It is important to note that with the different settings of control parameters (before or during the run) one explicitly controls specific processes (e.g., selection, mutation, and crossover) but only implicitly controls exploration and exploitation. In order to answer the question of how a good ratio between exploration and exploitation is achieved, Liu et al. [2009] classified two approaches.

(1) Uniprocess-driven approach.
(2) Multiprocess-driven approach.

For the uniprocess-driven approaches, a good balance between exploration and exploitation is achieved separately by an individual process (e.g., selection, mutation, crossover). Each process is independently responsible for the balance, and there is no coordination between processes for achieving the balance. An illustrative example is the crossover operator itself. In the beginning, whilst the population is still diverse, crossover performs exploration. At the end of the search, the population loses diversity, and by recombining, similar individuals' crossover exploratory powers are lost. The crossover operator itself, without changing the probability of crossover, shifts gradually from exploration to exploitation. In a similar manner, the non-uniform mutation operator [Zhao 2011] performs search uniformly at the early stage and very locally at the later stage. Another example is described in Gao and Xu [2011], where the Henon mutation operator is used as a global or local mutation operator. There is much additional work categorized within this area (e.g., [Bäck 1994; Eiben et al. 2004; Harik and Lobo 1999; Ronald 1995; Smith et al. 1993; Tsutsui et al. 1997b], amongst others). Note that uniprocess approaches do not mean that other processes are uninvolved in an exploration/exploitation process. Instead, a unique process is focused on balancing exploration and exploitation by using different techniques (e.g., adjusting corresponding control parameters). The challenge in this approach is how to achieve synergy among operators if they are uncooperative. To a lesser extent, researchers try to achieve balance using a multiprocess-driven approach. Balance is then coordinated among different processes (e.g., with high selection pressure for exploitation, a high rate mutation/crossover operator is chosen that is biased towards more exploration). An early example of multiprocess-driven approaches was the CHC algorithm [Eshelman 1991] that combined selection that always preserves the best individuals so far (exploitation) with a crossover operator that produces maximally different individuals (exploration). Another good example was reported in Fonseca and Fleming [1995], where it was found that proper collaboration between selection (fitness sharing) and crossover (mating restriction) significantly improved the performance of multiobjective GA. McGinley et al. [2011] also introduced ACROMUSE GA that synergistically

employs crossover for exploitation, mutation for exploration, and adjustable selection pressure for both exploration and exploitation.

3. ON CONTROLLING EXPLORATION AND EXPLOITATION

When and how should exploration and exploitation be controlled? Let us start with the easier part of this question, namely when.

Exploration and exploitation can be controlled offline by the proper settings of control parameters that will have an influence on algorithm search capabilities. However, developed algorithms will be applied to a vast variety of optimization problems, which will require different amounts of exploration and exploitation. Since a problem is unknown in advance, the algorithm's search capabilities could be enhanced if the amounts of exploration and exploitation were to be dynamically changed during a run. On the other hand, due to the conceptual advantages of parameter control over parameter tuning and existing experimental studies (e.g., [Alba and Dorronsoro 2005; Brest et al. 2006; Liu et al. 2009; Pan et al. 2011]), exploration and exploitation should be controlled online during the run.

A tricky issue is to determine on what occasion? Here, again, some deterministic schema (e.g., every k generations [Hesser and Männer 1991]) or adaptive schema (e.g., when best fitness did not change for several generations [Eiben et al. 2004], and when the diversity of a population drops under some threshold value [Shimodaira 1997; Ursem 2002]), can be applied. There is a need for intelligently controlling the balance between exploration and exploitation at different stages. The more intelligent the control, the better the results can be expected and/or the faster the algorithm will converge.

How to control exploration and exploitation balance is the more difficult part of the question. Balance between exploration and exploitation is implicit in EAs, and as such, directly controlling balance is difficult. But before controlling it, we need to know how to measure it. This is a fact in all scientific and engineering disciplines. How to measure exploration and exploitation is an open question in EAs [Beyer and Deb 2001], as far as we are aware. Intrinsic to this problem is that we need to know how these two phases are identified. If, in each process, both phases can be clearly identified, then some direct measures can be invented. Currently, indirect measures for exploration and exploitation are mostly used (see Section 4).

We describe and classify different diversity measures currently in use in the following section, because exploration is possible only if populations are diverse enough. Those diversity measures can be used to define the neighborhood relationships needed to delimit between exploration and exploitation.

3.1. Diversity

Diversity refers to differences among individuals, which can be at the genotype or phenotype levels. It is widely accepted within the EA community that the high diversity of a population greatly contributes to EA performance [Michalewicz 1996]. McPhee and Hopper [1999] vividly described this viewpoint. "Progress in evolution depends fundamentally on the existence of variation of population. Unfortunately, a key problem in many Evolutionary Computation (EC) systems is the loss of diversity through premature convergence. This lack of diversity often leads to stagnation, as the system finds itself trapped in local optima, lacking the genetic diversity needed to escape." Until now, diversity has been extensively investigated only in GP [Burke et al. 2004], whilst in other EAs, there has been no such treatment. Mattiussi et al. [2004] pointed out that diversity measures for individuals of a population with variable lengths and structures are much more complicated and might be more computationally extensive. Burke et al. [2004] investigate how one could improve fitness by controlling the diversity in GP,

whilst in this article, the similarity between individuals and their offspring, computed using different diversity measures, will define the exploration or exploitation zones.

There exist many different measures for diversity—genotypic as well as phenotypic—but there is no single measure that fits all problems and different types of EAs. As some authors [Burke et al. 2004; Galván-López et al. 2010; Paenke et al. 2009] have already pointed out, diversity measures are problem-specific. Moreover, if such diversity measures are going to be used in guiding evolution processes, we need to investigate if positive correlations exist. Burke et al. [2004] showed that there is not always a positive correlation between diversity measures and fitness in GP. In such cases, controlling diversity to improve fitness is unsuccessful.

It is worth mentioning that diversity is only roughly related to exploration and exploitation. High diversity does not necessarily mean that a diverse population was obtained by a good ratio between exploration and exploitation. Such a diverse population can be obtained by mere exploration without requiring a good balance between exploration and exploitation. Moreover, a diverse population does not mean that individuals are fit, just that they are different from each other (e.g., [Bosman and Thierens 2003] presented some good examples in multiobjective evolutionary algorithms (MOEAs)). We are interested in diversity that helps to find fit individuals. For this purpose, Mahfoud [1995] introduced the term *useful diversity*. A poor but diverse population is less attractive. Nevertheless, a diverse population is a prerequisite for exploration in order to avoid premature convergence to local optima. On the other hand, promoting diversity at all stages of an evolutionary process might even be counterproductive in a phase where high exploitation is needed. The relationship between diversity and exploration and exploitation is still unclear, and more research is needed, especially when identifying the types (phenotypic/genotypic) and amounts of diversity at different evolutionary stages [Burke et al. 2004].

As already mentioned, diversity can be measured at three levels.

(1) Genotype level (structural/syntactic/genotypic): differences among genomes within a population.
(2) Phenotype level (behavioural/semantic/phenotypic): differences among fitness values within a population.
(3) A complex or composite measure: a combination of the previous two cases [Burke et al. 2004]. Hence, we do not explicitly cover it in this article.

In most cases, identical genotypes will produce the same phenotype in EAs. So one might assume that a decrease in genotype diversity would necessarily cause a decrease in phenotype diversity. However, the relationship between genotype and phenotype is not always straightforward (e.g., noisy functions [Liu et al. 2007], changing environments). Multiple genes can influence a single phenotypic variable, or a single gene can influence multiple phenotypic variables. Another important issue is the concept of *locality* that tells us how well neighboring genotypes correspond to neighboring phenotypes, which is a key concept affecting exploration and exploitation of EAs. Galván-López et al. [2010] demonstrated that for an efficient search, the neighborhood has to be preserved. Namely, neighboring genotypes are mapped to neighboring phenotypes. This is called 'high locality'. If this is not the case, the distances at both levels do not correspond, and a search could be misleading. Representation with the property of high locality enables a more efficient search. Hence, representation really matters. Wineberg and Oppacher [2003] argued that diversity measures should not only include individuals, but also whole populations. In other words, distance between populations is needed.

High genotype diversity does not necessarily mean high phenotype diversity. This is true when one-to-one mapping between genotype and phenotypes does not exist. Such a mapping depends on the representation used and on the problem to be solved. For

example, Darwen and Yao [2001] reported that with the Iterated Prisoner's Dilemma, high genetic diversity can perversely correspond to low phenotypical diversity. Tsutsui et al. [1997a; 1997b] experimented with genotype and phenotype diversities. They found that performance depends on the type of problem under consideration, despite the fact that phenotype-based measures, usually perform better. There are also some other benefits of phenotype-based measures, such as the independence of the used representation schemes. It is often easier and less costly to calculate phenotypic diversity. This is especially true for GP [Burke et al. 2004]. Moreover, in multiobjective optimization, phenotype-based measures are more useful, because in this case, the goal is to find several nondominated solutions. In the following, we categorize and summarize the work using genotype and phenotype measures, respectively. These studies discuss population diversity in a qualitative manner, whilst Leung et al. [1997] provided quantitative analyses of population diversity. Their analyses revealed that to prevent premature convergence, increases in population size are more important than the role of variation regarding mutation probability. In Section 3.1.3, we will use different diversity measures to define neighborhood relationships, which are needed to delimit exploration from exploitation.

3.1.1. Genotype Measures. We classified genotypic diversity measures into the following.

—*Difference-based.* Numerous measures can be classified into this group—from counting different genotypes [Langdon 1998] and counting differently activated neurons representing particular search regions [Amor and Rettinger 2005], to counting frequencies of alleles [De Jong 1975; D'haeseleer and Bluming 1994]. Two early measures of allele frequencies proposed by De Jong [1975] are lost alleles and converged alleles. McPhee and Hopper [1999] proposed a simple measure, namely the number of different nodes in GP, and compared it with ancestry-based measures. The latter approach was extended to subtree variety measures in GP [Burke et al. 2002], which are the ratios of unique subtrees to total subtrees.
—*Distance-based.* This is probably the most widely used type of diversity measure nowadays. Various distances are taken into account: Hamming distance, Minkowski distance (Euclidean distance and Manhattan distance are special cases), cosine distance of similarity [Fister et al. 2010], edit distance [De Jong et al. 2001], distance to average point [Ursem 2002], to name a few. One of the first such measures to monitor population diversity was proposed by Whitley and Starkweather [1990], where the Hamming distance between the best and worst individuals is calculated. In diversity control oriented genetic algorithms (DCGA) [Shimodaira 1997], the Hamming distance between the best individual and candidates that determines candidates' selection pressure has been utilized, and more exploration or exploitation can be determined accordingly. Genotypical-forking GA, as proposed by Tsutsui et al. [1997a], also utilizes canonical Hamming distance as one of the conditions for performing forking. McGinley et al. [2011] employed Euclidean distance to adaptively control mutation and crossover rates. Additionally, healthy population diversity (HPD) is proposed for controlling selection pressure. Each individual is weighed—expressed by its fitness proportion to total fitness—when computing Euclidean distance so that more fit and diverse individuals may be selected.
—*Entropy-based.* Entropy is a very succinct measure for diversity which is gaining popularity. It represents the amount of population disorder, where increase in entropy represents increase in diversity. An additional benefit over previous approaches is that the distribution of values is also included within this measure, albeit the precise distribution is unknown. Entropy was shown to be a useful measure for genotypic diversity [Li et al. 2004; Liu et al. 2007; Masisi et al. 2008; Misevičius 2011].

—*Probability-based.* Simpson's diversity index falls into this group, which is often used to quantify the biodiversity of a habitat in ecology. It takes into account the number of species present, as well as the abundance of each species. Simpson's diversity index D measures the probability of whether two individuals, randomly selected from a sample, belong to the same species. $D = 0$ denotes that the diversity of a population is infinite, and conversely $D = 1$ denotes no diversity. Simpson's diversity index has been recently applied in EAs [Masisi et al. 2008; Paenke et al. 2009].

—*Ancestry (History)-based.* Diversity measure is obtained by contrasting the current population with those populations of previous generations, hence taking into account ancestries or history. McPhee and Hopper [1999] proposed several techniques for measuring population diversity based on genetic history. They noticed that an indicator of genetic diversity in the final population can be the number of individuals from the initial population that contributed genetic materials to the final population. They found that this number is extremely low in GP. Moreover, they found that in 90% of their runs, there was a single individual who was the root ancestor of every individual within the final population. In most cases, this leads to premature convergence.

3.1.2. Phenotype Measures. We classify phenotypic diversity measures into the following.

—*Difference-based.* The simplest difference-based diversity measure is the number of different phenotypes [Rosca 1995]. This measure can be easily extended to count different numbers of classes/ranks, where similar fitness values conform to the same class/rank producing histograms [Hutter and Legg 2006]. Luerssen [2005] proposed different ranking techniques, such as mean rank difference, mean rank difference across fitness cases, and mean rank difference across fitness cases for nondominated solutions. Luerssen reported that difference-based measures were overall significantly more effective than distance-based measures on his set of problems. Another example of a difference-based measurement is to count the number of individuals within the neighborhood hypercube, as used in phenotypic forking GA introduced by Tsutsui et al. [1997a].

—*Distance-based.* Various distance measures (e.g., Euclidian distance) can be used to find similarity between individuals within a population. For example, average distance to other individuals within a population is particularly popular in MOEAs, where the average distance between the nondominated solutions to the true Pareto front has been used [Zitzler et al. 2000], or similarly, a distance between an individual and a nondominated individual [Chaiyaratana et al. 2007]. Adra and Fleming [2011] also introduced a diversity indicator (I_s), which is the normalization measurement of the Euclidean distance of "the diagonal of the hypercube with vertices set to the extreme objective values observed in the achieved approximation set" with respect to the optimal spread. Another example was presented in Ursem [2002], where the distance to average-point was used. Distances can also be computed as differences in ranks in order to reduce any bias caused by a specific fitness function.

—*Entropy-based.* Entropy as a measure of diversity was first proposed by Rosca [1995] and since then has mainly been used as a succinct measure of phenotypic diversity [Burke et al. 2004; Liu et al. 2009].

—*Probability-based.* Simpson's diversity measure can also be applied to phenotypic diversity [Paenke et al. 2009].

Note that under phenotypic diversity measures, the ancestry (history)-based classification has been omitted. To the best of our knowledge, no such measures currently exist.

3.1.3. Definition of Exploration and Exploitation by Diversity/Similarity. Despite the fact that many researchers in their works have frequently mentioned exploration and exploitation, to date, both processes have never been formally defined. Often informal and somewhat vague definitions of exploration and exploitation processes have been used, similar to the informal definitions in Section 1, where exploration was defined as the process of visiting entirely new regions of a search space, whilst exploitation was defined as the process of visiting those regions of a search space within the neighborhood of previously visited points. Using different genotype and phenotype diversity measurements, as defined in Section 3, we are now able to define both processes, exploration and exploitation, in a sound manner. Let $d(\cdot, \cdot)$ denote the diversity/similarity measurements between two individuals within a population P. Formally, d is a function $P \times P \to \mathbb{R}$ measuring the similarities between two individuals at genotype or at phenotype levels (see Sections 3.1 and 3.2).

An example of computing similarity between two individuals $\mathbf{x}$ and $\mathbf{y}$ within a Euclidian space $\mathbb{R}^n$ is the Euclidian distance d_E. This is an example of a distance-based measurement.

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{1}$$

Yet another example of computing similarity between two individuals $\mathbf{x}$ and $\mathbf{y}$ using the difference-based measurement d_R is

$$d_R(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } R_x \neq R_y, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where two individuals $\mathbf{x}$ and $\mathbf{y}$ belong to the class/rank R_x and R_y, respectively.

Crucial for delimiting exploration from exploitation is a definition of similarity to the closest neighbor $\mathcal{SCN}$. However, when a new individual ind_{new} is created, a similarity measurement to the closest neighbor $\mathcal{SCN}$ can be defined in several ways.

—As a similarity to its parent(s), ind_{parent}.
—As a similarity to the most similar individual within the whole population P.
—As a similarity to the most similar individual in the subpopulation $P' \wedge (P' \subset P)$ (e.g., only to individuals which belong to the same niche).
—As a similarity to the most similar individual throughout the history of populations $\{P^t \mid t = 0..current\_gen\}$, where P^t denotes a population within a generation t.

In the first case, similarity to the closest neighbor $\mathcal{SCN}$ can be defined as follows.

$$\mathcal{SCN}(ind_{new}, P) = d(ind_{new}, ind_{parent}), \text{ where } ind_{parent} \in P. \tag{3}$$

If two or more parents $(ind_{parent_i}, i = 1..k \wedge k > 1)$ are used for creating a new individual, then ind_{parent} is the most similar contributing parent.

$$ind_{parent} = ind_{parent_i} \text{ with } \min_{i = 1..k} d(ind_{new}, ind_{parent_i}). \tag{4}$$

In the second case, similarity to the closest neighbor $\mathcal{SCN}$ can be defined as follows.

$$\mathcal{SCN}(ind_{new}, P) = \min_{\substack{ind \in P \\ ind_{new} \neq ind.}} d(ind_{new}, ind). \tag{5}$$

In the third case, similarity to the closest neighbor $\mathcal{SCN}$ can be defined as follows.

$$\mathcal{SCN}(ind_{new}, P') = \min_{\substack{ind \in P' \wedge (P' \subset P) \\ ind_{new} \neq ind.}} d(ind_{new}, ind). \tag{6}$$

In the fourth case, similarity to the closest neighbor $\mathcal{SCN}$ can be defined as follows.

$$\mathcal{SCN}(ind_{new}, P^t) = \min_{\substack{ind \in P^t, t = 0..current\_gen \\ ind_{new} \neq ind}} d(ind_{new}, ind).\tag{7}$$

Note that in the first case, we are only looking for similarity to the closest parent, whilst in the second and the third cases, the whole population or subpopulation is taken into account. According to the informal definition of exploitation, where visiting is within the neighborhood of previously visited points (e.g., not only from current contributing parent(s) or within a current population), the fourth definition of $\mathcal{SCN}$ seems to be the most appropriate.

Finally, the process of exploration happens when $\mathcal{SCN}(ind_{new}, P) > X$, where X is a threshold value that defines the boundary of the neighborhood of the closest neighbor and is problem-dependent. In other words, the exploration process visits points which are outside of the current neighborhood of the closest neighbor. Similarly, the process of exploitation happens when $\mathcal{SCN}(ind_{new}, P) \leq X$.

$$\mathcal{SCN}(ind_{new}, P) > X \qquad \text{(exploration);}\tag{8}$$

$$\mathcal{SCN}(ind_{new}, P) \leq X \qquad \text{(exploitation).}\tag{9}$$

4. CLASSIFICATION OF CURRENT APPROACHES FOR BALANCING BETWEEN EXPLORATION AND EXPLOITATION

The diversity of a population has already been recognized as one of the important factors from the early years of EAs. From the exploration and exploitation viewpoints, an increase in diversity indicated that EA was in the phase of exploration, whilst a decrease in diversity indicated that EA was in the phase of exploitation. The relationship between exploration and diversity is indeed profound, and the exploration operators can also be defined as those that allow for maintaining the diversity of the population [Soza et al. 2011]. One of the simplest methods for achieving a good balance between exploration and exploitation is to maintain a diverse population. However, a diverse population is simply a prerequisite for a good balance between exploration and exploitation rather than a guarantee. For previously mentioned reasons, our classification is heavily based on diversity, although balance between exploration and exploitation can also be achieved by other means. The simplest one is fitness. A number of approaches utilize fitness to indirectly guide exploration and exploitation. For example, the 1/5 success rule uses fitness to determine whether an individual has mutated successfully and then decides if the mutation rate needs to be changed. On the other hand, in such cases, diversity is frequently included in the individual's fitness, and the boundary between both approaches becomes blurred.

We classify the current approaches regarding the balance between exploration and exploitation into achieving exploration and exploitation balance through diversity maintenance, through diversity control, diversity learning, and other more direct approaches. Whilst the former three groups are primarily interested in population diversity and hence only implicitly address exploration and exploitation, the last group more directly tackles the problem of balance between exploration and exploitation. This classification is also presented in Figure 1, whilst Table I chronologically presents the cited papers.

4.1. Achieving Exploration and Exploitation Balance through Diversity Maintenance

In this section, our focus is on maintaining diversity through different techniques. Instead of measuring diversity and using this measure as feedback for further enhanced

Approaches for controlling exploration and exploitation
balance through diversity

maintaining

non-niching

population-based
- varying population size
- duplicate elimination
- infusion techniques
- external archives
- migration between
  subpopulations

controlling
- through selection
- through crossover/mutation
- through changing population

selection-based
- changing selection pressure
- replacement restrictions

crossover/mutation-based
- mating restrictions
- disruptive operators

hybrid

learning
- cultural learning
- self-organizing maps
- binary space partitioning trees
- estimation of distribution

niching

fitness-based
- explicit and implicit
  fitness sharing
- clearing
- modified clearing
- clustering

other direct approaches
- using different subpopulations
  for exploration and exploitation
- triggers cause alternation between
  exploration and exploitation
- ancestry-based

replacement-based
- deterministic crowding
- probabilistic crowding
- restricted tournament selection

preservation-based
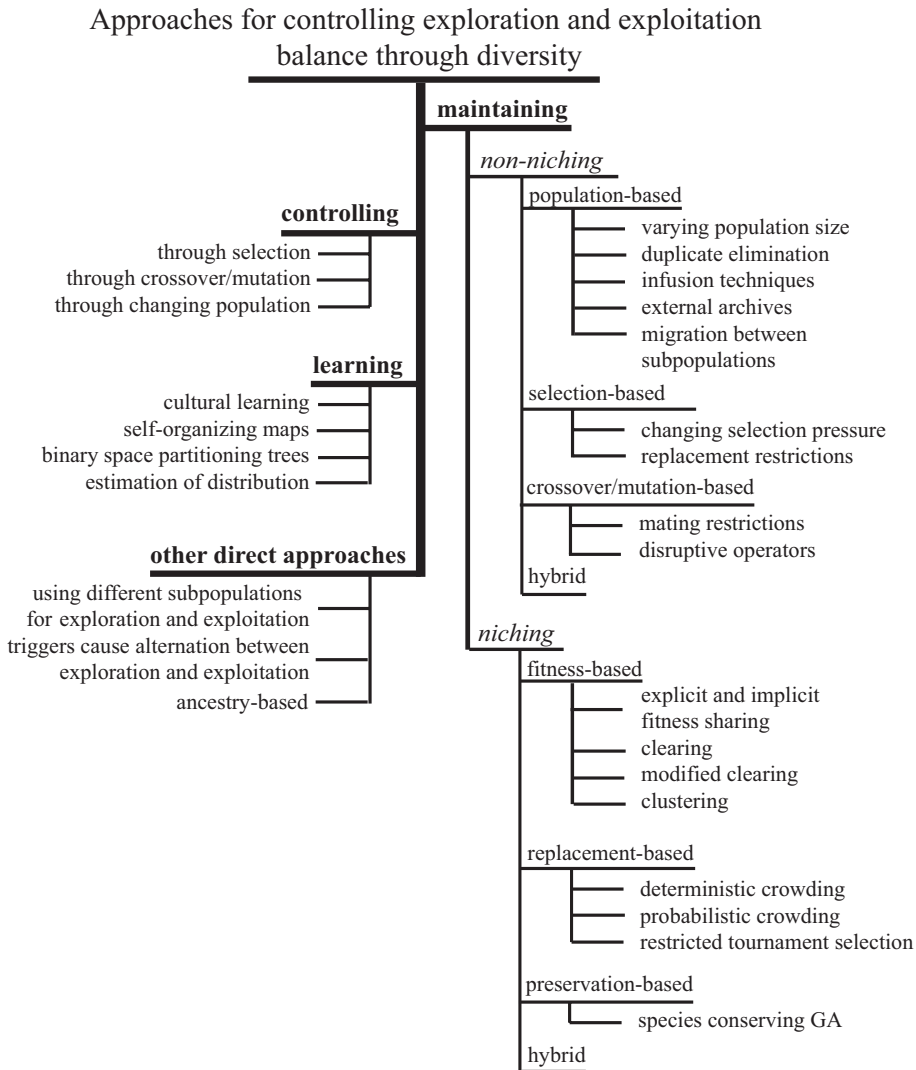- species conserving GA

hybrid

Fig. 1.   Current approaches to balancing exploration and exploitation.

diversity, and to improve the balance between exploration and exploitation, it is simply
assumed that the proposed techniques will maintain diversity per se, and hence the
balance between exploration and exploitation will be achieved. However, it is difficult
to determine a useful amount of diversity. High diversity is needed to escape from local
optima, whilst low diversity is needed to fine-tune the solutions. Numerous different
methods, developed mostly in the 1990s, have been used for diversity maintenance.
Mahfoud's study was one of the first towards a comprehensive theory of diversity
for GAs [Mahfoud 1995], where methods were classified as non-niching and niching.
Both non-niching and niching methods are able to maintain a population of diverse
individuals, whilst niching methods are also capable of locating multiple optimal
solutions. Since this classification also appears in other work (e.g., [Toffolo and Benini

Table I. Classification of the Cited Work in Chronological Order

Maintenance (Non-niching)	
Population-based	[Mauldin 1984; Grefenstette 1992; Koza 1992; Ramsey and Grefenstette 1993; McPhee and Hopper 1999; Greenwood et al. 1999; Martin et al. 1999; Zitzler and Thiele 1999; Krink et al. 2000; Ursem 2000; Zitzler et al. 2000; Leung and Wang 2001; Bosman and Thierens 2003; Koumousis and Katsaras 2006; Chaiyaratana et al. 2007; Gong et al. 2008; Rahnamayan et al. 2008; Yang 2008; Zhang and Sanderson 2009; Araujo and Merelo 2011; Jia et al. 2011; Li and Wang 2011; Wang et al. 2012]
Selection-based	[De Jong 1975; Grefenstette 1986; Michalewicz 1996; Gen and Cheng 1997; Matsui 1999; Hutter and Legg 2006; Lozano et al. 2008; Chen et al. 2009]
Crossover/Mutation-based	[Deb and Goldberg 1989; Eshelman and Schaffer 1991; Eshelman 1991; De Jong and Spears 1992; Cobb and Grefenstette 1993; Ronald 1995; Joan-Arinyo et al. 2011]
Hybrid	[Ghosh et al. 1996; Harik et al. 1999; Paenke et al. 2009; Qin et al. 2009; Lee et al. 2011; Mallipeddi et al. 2011]
Maintenance (Niching)	
Fitness-based	[Holland 1975; Goldberg and Richardson 1987; Smith et al. 1993; Yin and Germay 1993; Petrowski 1996; Singh and Deb 2006]
Replacement-based	[Mahfoud 1995; Harik 1995; Mengshoel and Goldberg 1999]
Preservation-based	[Li et al. 2002]
Hybrid	[Yu and Suganthan 2010; Liang and Leung 2011]
Controlling	
Through selection	[Mori et al. 1995; Shimodaira 1997; Bersano-Begey 1997; Bosman and Thierens 2003; Wong et al. 2003; Chaiyaratana et al. 2007; Adra and Fleming 2011; McGinley et al. 2011]
Through crossover and mutation	[Whitley and Starkweather 1990; Srinivas and Patnaik 1994; Wong et al. 2003; Li et al. 2004; Jose-Revuelta 2007; McGinley et al. 2011]
Through changing population	[Rosca 1995; Tsujimura and Gen 1998; Liu et al. 2009]
Learning	
Cultural learning	[Jin and Reynolds 1999; Curran and O'Riordan 2006; Becerra and Coello Coello 2006; Soza et al. 2011]
Self-organizing maps	[Amor and Rettinger 2005]
Binary space-partitioning tree	[Yuen and Chow 2009; Chow and Yuen 2011]
Estimation of distribution	[Mühlenbein and Paaß 1996]
Other Direct Approaches	
Using different sub-populations for exploration and exploitation	[Tsutsui et al. 1997b, 1997a; Oppacher and Wineberg 1999; Goh et al. 2003]
triggers cause alternation between exploration and exploitation	[Hart 1994; Freisleben and Merz 1996; Moscato 1999; Merz and Freisleben 2000; Ursem 2002; Ishibuchi et al. 2003, 2010a; Alba and Dorronsoro 2005; Krasnogor and Smith 2005; Ong et al. 2006; Misevičius and Rubliauskas 2008; Nguyen et al. 2009; Liao 2010; Blum et al. 2011; Mashinchi et al. 2011; Lin and Chen 2011; Mongus et al. 2012]
Ancestry-based	[Črepinšek et al. 2011]

2003; Zitzler and Thiele 1999]), this article adopts it. This article further classifies non-niching methods into the following.

—*Population-based.* Diversity is achieved by varying population size, duplicate elimination, infusion techniques, external archives, or migration between subpopulations.
—*Selection-based.* Diversity is achieved by changing selection pressure or replacement restrictions.

—*Crossover/mutation-based.* Diversity is achieved by mating restrictions or disruptive operators.
—*Hybrid.* Diversity is achieved by ensembles or other specific approaches.

Brief descriptions of non-niching methods are shown as follows.

—*Increasing population size.* It is among the simplest approaches, but several studies (e.g., [McPhee and Hopper 1999]) reported that increasing population size does not always lead to correspondingly increased diversity.
—*Duplicate elimination.* By using this approach, an individual that already exists in a population is eliminated, and another randomly generated individual is inserted. Duplicate elimination is an important part of the algorithms presented in Chaiyaratana et al. [2007]. An improvement in duplicate elimination is the uniqueness-assurance method by Mauldin [1984], where individuals, instead of being eliminated, are mutated as long as they become sufficiently different (at least for k bits, where k is decreasing over generations). Hence, this approach can also be regarded as an example of infusion, which is described next.
—*Infusion techniques.* New individuals are randomly inserted after a certain number of generations or special initialization techniques are used (e.g., chaotic initialization [Li and Wang 2011], orthogonal design [Leung and Wang 2001]). The former approach is also called reseeding or extinction [Greenwood et al. 1999; Krink et al. 2000]. An early approach is Grefenstette's random immigrants approach [1992], where random individuals are inserted into the population every generation. Koza [1992] introduced *decimation*, where a substantial percentage of a population has been replaced by random individuals at regular intervals. Reinitialization has been proven successful in dynamic and changing environments, where the re-seeding of good individuals from past cases has been proposed every 150 generations [Ramsey and Grefenstette 1993]. Koumousis and Katsaras [2006] proposed a Saw-Tooth GA with periodic reinitialization and variable population sizes achieving better population diversity and overall performance. Rahnamayan et al. [2008], instead of inserting random individuals, recently proposed the concept of *opposition*, inserting opposite individuals for diversity maintenance. A more sophisticated approach is presented by Jia et al. [2011], where a chaotic local search was proposed. Since local search may result in premature convergence, the authors enhanced diversity and hence exploration abilities by chaotic random local search, where the search space of the chaotic local search shrinks with the growth of the function evaluations. Orthogonal crossover [Leung and Wang 2001], which is based on orthogonal design, can also be seen as a systematic and rational local search maintaining a diversity of population [Gong et al. 2008; Wang et al. 2012]. If replacement is done as a response to a lack of progress or a change in a dynamic environment, then these techniques can be classified as diversity control (as described in the next section). Yang [2008] recently proposed new immigrant schemes—memory-based and elitism-based immigrants—for dynamic optimization problems using historical information.
—*External archives.* External archives have been also used as a diversification technique. Whilst archives in EMO have been used primarily for keeping track of nondominated solutions [Zitzler and Thiele 1999; Zitzler et al. 2000; Bosman and Thierens 2003] and inherently maintaining a diverse population, there have also been some other proposals for using external archives during single-objective optimization. For example, Zhang and Sanderson [2009] also used external archives to maintain diverse population by archiving recently explored inferior solutions.
—*Migration between subpopulations (islands).* Many (parallel) GAs introduce diversity and prevent premature stagnation by simply exchanging individuals between subpopulations. Migrants can replace less fit individuals, randomly chosen individuals,

or the most similar individuals, etc. The main idea behind this approach is that different subpopulations can maintain different promising regions of the search space and hence maintain diversity [Martin et al. 1999]. The dynamics of exploration and exploitation are further influenced by the number of subpopulations, the sizes of these subpopulations, differences in subpopulation interconnections (often referred to as communication topology), frequency of communication (epoch length), and the quality of the migrants. Ursem [2000] proposed a multinational genetic algorithm (MGA), where subpopulations, called nations, are self-formed based on a hill-valley detection algorithm. A nation consists of a population, government (fittest individuals), and policy (point in the search space to which a nation is approaching). The hill-valley detection is responsible for the migration of individuals among nations, the creation of new nations within unexplored regions, and the merging of nations if they have the same policy. Ursem showed that diversity maintenance in MGA could lead to better performance and the ability to track multiple peaks simultaneously, and as such, could also be regarded as a niching method. Araujo and Merelo [2011] introduced a MultiKulti algorithm, where subpopulations are constructed as a ring topology, and migrations will only occur between neighboring subpopulations. Migrants are selected from a subpopulation if they possess the most different genotype(s) from the selected representative of the receiving/neighboring subpopulation, where either the best individual or the consensus sequence (i.e., the sequence that reflects the most common base or amino acid found at each position in a genome [Watson et al. 2004]) is picked as the representative. Araujo and Merelo showed that such selection policies and topology increased the diversity and outperformed nonadaptive random and best-individual migration policies on two discrete optimization problems.

—*Changing selection pressure.* Many techniques have been proposed to prevent selection being biased towards highly fit individuals. Two of the simplest techniques are ranking selection [Michalewicz 1996] and scaling [Grefenstette 1986]. Some more sophisticated selection mechanisms that try to ensure that the fittest individual will not always be selected and the weakest individual not always rejected are briefly described next. Matsui [1999] proposed two selection methods—correlative tournament selection and correlative family-based selection—for improving population diversity. Hutter and Legg [2006] maintained genetic diversity in terms of fitness diversity. Fitness values are divided into several classes, and each class has an equal opportunity of survival and hence preserving diversity through selection. Chen et al. [2009] recently proposed several different selection schemes (e.g., integrating power-law distribution and tournament selection) to enhance population diversity in EP. On the other hand, if the search needs to be more exploitative, elitist strategies are often employed [Gen and Cheng 1997].

—*Replacement restrictions.* De Jong introduced crowding [1975], called standard crowding, to maintain diversity. Each newly generated individual replaces the most similar existing individual from a random subpopulation of size CF (the crowding factor). Lozano et al. [2008] proposed a hybrid replacement scheme where individuals of poor fitness and low contribution to diversity are replaced, thereby promoting exploitation (highly fit individuals) and exploration (high contribution to diversity). Both objectives, that is, optimizing fitness function and enhancing population diversity, are fulfilled in this manner.

—*Mating restrictions.* Individuals are allowed to mate if they fulfill special conditions. A typical example is incest prevention [Eshelman and Schaffer 1991]. Ronald [1995] introduced genders into mating, where selection of the second mate is based on a seduction function between the first and second mates. Deb and Goldberg's [1989] mating restrictions, where individuals are allowed to mate only if they are sufficiently distinct, also fall into this category.

—*Disruptive operators*. Until now, several disruptive operators have been proposed which address diversity issues explicitly. A prime example is the hyper-mutation operator [Eshelman 1991; Cobb and Grefenstette 1993; Joan-Arinyo et al. 2011]. However, De Jong and Spears [1992] pointed out that disruption does not necessarily imply useful exploration.

—*Hybrid approaches*. There are some other approaches that can not be easily classified into the aforementioned classes. This is due to the use of unconventional concepts (e.g., concepts of age and compact GA) or blending different approaches (e.g., ensembles [Mallipeddi et al. 2011]). Inevitably, there may be some approaches we have missed. Gosh et al. [1996] developed an aging genetic algorithm (aGA), where the concept of age for an individual is included into the fitness function as well as into mating. As in nature, adult individuals are considered more fit for mating. On a given problem, Gosh et al. reported that aGA maintains more diversity in the population, whilst its performance has also been improved. Paenke et al. [2009] showed that longer individual lifetime slows down the genotypic diversity loss. The compact genetic algorithm (cGA) [Harik et al. 1999] used a probability vector for current population in order to obtain offspring. An element of probability, vector pv_i denotes a probability to assign 1 to the ith gene of an individual. Diversity is simply achieved by increasing/decreasing pv_i by $\frac{1}{pop\_size}$. Lee et al. [2011] extend cGA with belief vectors, where each element has probability distribution with a mean and a variance. Diversity is adaptively controlled by calculating entropy and updating the variance of a belief vector. The SaDE algorithm [Qin et al. 2009] attempted to maintain a good balance between exploitation and exploration by choosing different strategies in differential evolution (DE) [Storn and Price 1997] (e.g., DE/rand/1/bin, DE/best/1/bin) for each individual. Authors have taken advantage of the fact that different strategies have different exploration/exploitation capabilities.

On the other hand, niching methods promote the formation and maintenance of stable subpopulations within the neighborhood of optimal solutions, thereby achieving a population diversity. Niching methods have already been well studied, and readers are referred to Mahfoud [1995] and Sareni and Krähenbühl [1998] for more information. We have classified niching methods into the following.

—*Fitness-based*. Fitness sharing (explicit [Goldberg and Richardson 1987; Holland 1975] and implicit [Smith et al. 1993]), clearing [Petrowski 1996], modified clearing [Singh and Deb 2006], clustering [Yin and Germay 1993].

—*Replacement-based*. Deterministic crowding [Mahfoud 1995], probabilistic crowding [Mengshoel and Goldberg 1999], restricted tournament selection [Harik 1995].

—*Preservation-based*. Species conserving GA [Li et al. 2002].

—*Hybrid*. Adaptive elitist-population based GA [Liang and Leung 2011], ensemble of niching algorithms [Yu and Suganthan 2010].

Although most work done using non-niching and niching methods simply expects an increase in diversity and hence better balance between exploration and exploitation, not all diversification is useful. In this respect, niching methods can be regarded as stronger diversification methods [Friedrich et al. 2008]. There are only a few theoretical works on diversity mechanisms, and we would like to mention the following. De Jong and Spears [1992] provided the theoretical analysis of crossover operator (multipoint crossover vs. uniform crossover) with respect to disruptive effect, recombination potential, and exploratory power. Friedrich et al. [2007] provided a rigorous analysis of simple diversification mechanisms (duplicate elimination at genotype and phenotype levels) and analyzed it with respect to the runtime behaviour of an EA, whilst Storch [2004] analyzed duplicate elimination with respect to population size. Friedrich

et al. [2008] also provided the theoretical analysis of niching methods (fitness sharing and deterministic crowding) on simple bimodal functions. They rigorously proved that without any diversification method or when only genotype/phenotype duplicate elimination is used in $(\mu + 1)$ EA, is the probability of being trapped into local optima almost $\frac{1}{2}$, whilst the probability of finding both optima using fitness sharing or deterministic crowding is very high.

It is worth mentioning that diversity maintenance is very important to EMO [Smith et al. 1993; Toffolo and Benini 2003; Zitzler and Thiele 1999]. For example, in order to maintain the diversity among nondominated solutions, Zitzler et al. [2002] proposed the k-nearest neighbor clustering technique. Zitzler et al. noticed that not only the average distance between the nondominated solutions to the true Pareto front is important, but also the distribution of the nondominated solutions.

4.2. Achieving Exploration and Exploitation Balance through Diversity Control

The main difference between diversity maintenance and diversity control is that in the latter case, population diversity, individual fitness, and/or fitness improvements are measured and used as a feedback to steer an evolution process towards exploration or exploitation. We classify different approaches based on the process/operator responsible for diversification. Another possibility would be to take into account how population diversity is measured (genotypic or phenotypic diversity measures, see Section 3).

—*Diversity is controlled and preserved through selection.* Survival probability can be computed based on population diversity, or diversity can be included within those fitness functions that further drive the selection process. Several approaches fall into this category, and we mention the following. Mori et al. [1995] proposed a thermo-dynamical genetic algorithm (TDGA), which adopts the concept of temperature and entropy within the selection rule for better control of population diversity. Shimodaira [1997] proposed a diversity-control-oriented genetic algorithm (DCGA), where individuals that are farther from the best individual, but which always survive, have higher survival probabilities. Chaiyaratana et al. [2007] proposed a modified DCGA, where survival probability depends on similarity at the phenotype level. Wong et al. [2003] controlled population diversity by repelling population from the representative one. This was achieved by including diversity within a fitness function. In such a manner, individuals with rare alleles would be fitter, and the survival probabilities of such individuals would be increased. Bersano-Begey [1997] also controlled diversity through a fitness function, which keeps track of how many individuals have solved a particular fitness case. In this manner, it is possible to detect when the population is locked in on a partial solution. This approach can be augmented by tracking over how many generations this situation has persisted. McGinley et al. [2011] introduced healthy population diversity as feedback to adaptively controlled selection pressure through tournament size, in order to avoid premature convergence or exploitation from the same cluster of highly fitted individuals continuously. Adra and Fleming [2011] employed the diversity indicator (I_s) mentioned in Section 3.1.2 to activate a diversity mechanism for further improving the performance of NSGA-II in terms of convergence and diversity. Binary tournament selection with random tie breaking will be inactive if $I_s < 1$ so that "the exploitation of diversity should not precede the exploitation of proximity" [Bosman and Thierens 2003].

—*Diversity is controlled through crossover and mutation.* This is an easy and natural way to control diversity. Hence, the majority of approaches fall into this category. The idea is to increase/decrease the probability of crossover and/or mutation after population diversity, fitness, and/or fitness improvements are computed. Approaches differ from each other mainly on how diversity is computed: explicitly by different

diversity measures (see Section 3) and/or implicitly by fitness or fitness improvements. Selected approaches are described as follows. Whitley and Starkweather [1990] applied the adaptive probability of mutation based on Hamming distance in order to sustain population diversity. Srinivas and Patnaik [1994] proposed an adaptive genetic algorithm (AGA), where the adaptive probabilities of mutation and crossover are used to control the diversity of the population and to achieve good convergence. The probabilities of crossover and mutation are based on the fitness values of individuals, and in order to achieve balance between exploration and exploitation, they increase the probabilities of mutation and crossover when the population tends to get stuck at local optima, and decrease the probabilities when the population is scattered within the search space. The PRAM algorithm [Wong et al. 2003] adapted the probabilities of crossover and mutation in order to determine an appropriate balance between exploration and exploitation. Based on fitness improvement, they used a set of greedy rules (e.g., expand rule, stay rule, left move rule, and right move rule) to adapt control parameters. Another example is the diversity-guided microgenetic algorithm (DGμGA) [Jose-Revuelta 2007], where a fitness-proportional entropy-based diversity measure is used to guide genetic algorithms. The authors correctly recognized that high entropy can be obtained when a population is diverse but far from global optima (usually at the beginning stage), and when a population is diverse but close to global optima (at the latest stage). This phenomenon is called entropy ambiguity. In order to eliminate such a phenomenon, the authors used an additional measure—how fast the convergence occurs—expressed as an average mean value of individuals' fitness over k successive generations. When convergence is unsatisfactory, the whole population is randomly reinitialized. Li et al. [2004] proposed the adaptive genetic algorithm with diversity-guided mutation and crossover (AGAD), where mutation and crossover probabilities are adaptively controlled by diversity measures based on entropy. Note that McGinley et al. [2011] (categorized in the previous group) also utilized Euclidean distance to adaptively control mutation and crossover rates, respectively, for exploration and exploitation.

—Diversity is controlled through a changing population. After measuring population diversity, either the population size or the population alone is changed. Not many approaches fall into this category. Liu et al. [2009] presented an entropy-based exploration and exploitation approach for controlling an evolution process. It tends toward exploration when entropy is low and tends toward exploitation when entropy is high. Linear, Gaussian, fitness proportional, and cluster entropies were introduced by extending Rosca's entropy [Rosca 1995]. The extension was done by redefining how individuals are classified into different entropy classes, and whether the class boundaries are changed adaptively or self-adaptively. Despite the fact that this approach mainly adapts the probabilities of mutation and crossover, some experiments also adaptively changed population size. Tsujimura and Gen proposed an entropy-based genetic algorithm (EBGA) [1998], where the diversity of loci is measured by entropy. When the population diversity is too low, the diversity is improved by swapping loci with low entropy values.

4.3. Achieving Exploration and Exploitation Balance through Diversity Learning

The main difference between diversity control and diversity learning is that, in the former case, short-term history (e.g., current population) is often used during diversity computation, whilst, in the latter case, long-term history is used in combination with different machine learning techniques to learn (un)explored search areas. The history of a population has also been an important criterion for the classification of EAs in Calegary et al. [1999]. Nevertheless, diversity learning approaches are rarely proposed, and currently researchers are using cultural learning, self-organizing maps, binary

space partitioning trees, and an estimation of distribution, in order to learn about promising locations within the search space.

—Curran and O'Riordan [2006] investigated population diversity through cultural learning, where individuals were able to pass their knowledge to individuals in subsequent generations. Researchers in machine learning classify learning into population learning, lifetime learning, and cultural learning. Population learning is well known in the EA community and represents an evolving population through generations, whilst lifetime learning can be achieved in evolutionary algorithms by local search (memetic algorithms [Moscato 1999]). In this type of learning, individuals acquire knowledge during their lifetimes, for example, by local search. Using population and lifetime learning, global search and local optimization are combined. Hence, a balance between exploration and exploitation can be achieved (this method of achieving exploration and exploitation balance is further discussed in Section 4.4). An interesting learning introduced into cultural learning is that acquired knowledge can be passed to subsequent generations without altering their genomes. This is possible in several different ways. Curran and O'Riordan [2006] used a teacher/pupil scenario, where pupils (individuals of the current generation) imitate the behaviour of teachers (highly fit individuals in previous generations). Individuals are generated from their genetic code and are immediately exposed to teaching. Curran and O'Riordan showed that cultural learning on their test cases significantly increased genotypic as well as phenotypic diversity. This is credited to a selection process that no longer selects individuals based on their genetic materials, but instead on an individual's ability to learn. Several cultural algorithms have been developed (e.g., [Becerra and Coello Coello 2006; Soza et al. 2011]), where the search was improved by extracting domain knowledge (e.g., normative knowledge about intervals for decision variables where good solutions had been found in the past). The learning outcome influenced the variation operator and promoted diversity of population. In a similar manner, Jin and Reynolds [1999] used a cultural algorithm for avoiding infeasible regions and for promoting the exploration of feasible regions.

—Amor and Rettinger [2005] introduced intelligent exploration of the search space using self-organizing maps (SOM) [Kohonen 2001]. GASOM (GA using SOM) takes into account not only the diversity of a current population, but also across the whole evolution process. This concept is called *novelty*. By using SOM, unexplored regions can be learned about efficiently by incorporating novelty into the fitness function. Exploration of unexplored regions was encouraged, whilst exploitation was still promoted by an objective function. And, finally, by reseeding a population with individuals of high novelty, Amor and Rettinger were able to achieve diversity and a good balance between exploration and exploitation. Furthermore, the authors showed how the ratio between exploration and exploitation, albeit very rough, can be computed.

—Yuen and Chow [2009] proposed a novel genetic algorithm, called the non-revisiting genetic algorithm with parameter-less adaptive mutation (NrGA), which never revisits previous solutions, and hence population diversity is automatically assured. This is achieved in an efficient way using a binary space-partitioning (BSP) tree, with each node carrying the densities of the evaluated solutions. Not only have previously visited solutions been recorded, but in the case of revisiting, NrGA is able to find the nearest unvisited subspace and then perform a random mutation within this subspace. Such a mutation is adaptive and parameter-less, because the smaller the unvisited subspace, the smaller the mutation step, and vice versa. Yuen and Chow showed that diversity can be maintained by learning already visited solutions. The proposed algorithm outperforms canonical GAs with and without simple diversity mechanisms, a generic real-coded genetic algorithm, as well as the

covariance matrix adaptation evolution strategy (CMA-ES), on several benchmark functions. Chow and Yuen [2011] also proposed a history-driven evolutionary algorithm (HdEA), where a BSP tree records the searched individuals and their fitness values during the evolution process. Guided anisotropic search is introduced to govern the search direction based on the history of the BSP tree: an individual is either exploitatively mutated towards the nearest historical optimum or randomly mutated if the individual is already a local optimum. HdEA outperforms eight benchmark real-coded evolutionary algorithms on several multimodal functions.
—Mühlenbein and Paaß [1996] proposed an estimation of distribution algorithm (EDA) that uses machine learning techniques to learn about locations within the more promising regions of the search space.

4.4. Other Direct Approaches

Although these approaches, an exception being the approach presented in Črepinšek et al. [2011], suggested that there is explicit control between exploration and exploitation and that two phases are clearly identified, actually both phases are still interleaved. For example, even though the process can be in the exploration phase, an individual can be generated by exploiting an already existing region. We have identified three different subcategories. In the first subcategory, different subpopulations are used for a particular phase, whilst in the second subcategory, triggers cause switching between phases. The third subcategory presents an ancestry tree-based approach, which is currently the only available direct measure for exploration and exploitation.

—Subpopulations are used to delimit exploration from exploitation. Some subpopulations are exclusively used for a particular phase.
  —Bi-population GA (bGA) [Tsutsui et al. 1997b] introduced two separate populations respectively for exploration and exploitation, and implicitly maintained diversity. Explorer sub-GA mainly performs global exploration of the search space (in addition, a restart mechanism is implemented that avoids the subpopulation being trapped in local optima), whilst the exploiter sub-GA performs exploitation of the fit local areas of the search space around the neighborhood of the best-so-far solutions. The word "mainly" is important here, because a solid line between the two phases is still undrawn. In the explorer sub-GA, authors use coarse-grained mutations, whilst in the exploiter sub-GA, fine-grained mutation is used. However, the same crossover operator is used in both populations. Moreover, in the explorer sub-GA, a larger population is used than in the exploiter sub-GA. Overall, there is no guarantee that during the exploitation phase, an individual is generated which is in the neighborhood of the current individual.
  —Tsutsui et al. [1997a] proposed forking GA, where a multipopulation scheme with one parent population for the exploration mode and one or more child populations for the exploitation mode are used. Populations can be divided/forked on genotypic as well on phenotypic search spaces. In the former case, a subspace is defined by salient schema, whilst in the latter case, a subspace is defined by a neighborhood hypercube around the current best individual. For genotypic forking GA, whenever all of the following conditions occur, the population is forked: the best so far has not been updated for a specified number of generations, the population converges to smaller diversity where canonical Hamming distance is used, and the order of the salient schema is more than a specified constant. Similarly, phenotypic forking GA forks if the best individual so far has not been updated for a specified number of generations, and the number of individuals within the neighborhood hypercube is more than its predefined threshold. Tsutsui et al. [1997a] were able to divide the entire search space into subspaces, and different child populations

were responsible for exploiting particular subspaces, whilst the parent population was responsible for the rest of the space. However, in the parent population, new individuals might be generated, which is in the exploitation zone of the parent population. In this case, the parent population is not solely exploring the region. Nevertheless, the forking GA maintains good balance within the exploring parent population and when exploiting child populations. Finally, it was shown that phenotypic forking GA can also be effectively used for niche formation and, as such, this approach is also mentioned in Section 4.1.

—Oppacher and Wineberg [1999] proposed a shifting balance genetic algorithm (SBGA) to address multimodal optimization within a dynamic environment (moving peak problem). The SBGA categorizes a population into core and colony groups. The objectives of the core and colony groups are to exploit and explore the fitness landscape, respectively. A *containment* factor, derived from Hamming distance, is introduced to determine the ratio of individuals to be selected inside the core for exploitation and outside the core (i.e., inside colonies) for exploration. It is interesting to note that smaller subpopulations (colonies) are now responsible for exploration (opposite to forking GA [Tsutsui et al. 1997a]). The rationale for this is Wright's shifting balance theory, where small populations are more sensitive to random genetic drift, which allows populations to cross the low fitness valleys more easily and better explore that region. On the other hand, selection is more effective within larger populations. Again, it is hard to delimit exploration from exploitation on the core population.

—Goh et al. [2003] randomly separated a population into two sexes, female and male, which are responsible for exploration and exploitation, respectively. The rationale comes from nature, where for some species, female choice selection is in play. Females will always mate, while only more attractive males will be selected for mating. While the balance between exploration and exploitation is promoted, there is still no explicit control over either phase, since there is nothing to prevent an individual from being generated, either within an exploration or exploitation zone.

—Population is the same, but different triggers cause alternation between exploration and exploitation phases.

—EAs with local search can be regarded as a prime example of this case. These algorithms are also known as hybrid EAs [Blum et al. 2011; Liao 2010; Mashinchi et al. 2011; Mongus et al. 2012; Miševičius and Rubliauskas 2008], genetic local search algorithms [Merz and Freisleben 2000], or memetic algorithms [Moscato 1999; Krasnogor and Smith 2005; Ong et al. 2006]. An important part of these algorithms is a local search step where one or several local optimizers can be applied to individuals to further refine a solution by exploiting its neighborhood. As such, this step can be regarded solely as an exploitation operator. The local search step is usually done after the recombination operators, just prior to the selection step (one exception is that of Ishibuchi et al. [2010a], who found that better results were obtained for some multiobjective combinatorial optimization problems by using local search after the selection step). Hence, the exploitation phase is triggered explicitly by calling the local search method. Even though this approach may seem to be regarded as an explicit approach to control exploration and exploitation, this is not so. It is because other steps (e.g., mutation and crossover) can generate a solution in both zones. Hence, exploration and exploitation are still incompletely separated. Since these algorithms additionally expose the exploitation phase, special attention must be paid to guarantee that a proper balance between exploration and exploitation has still been achieved [Ishibuchi et al. 2003]. This can lead to extensive tuning of the control parameters, too. Freisleben and Merz

[1996] reported that a local search can lead to a disastrous loss of diversity, thus requiring a completely new selection scheme. Nevertheless, the success of memetic algorithms has often been credited to a good trade-off between the exploration abilities of EAs and the exploitation abilities of the local search [Krasnogor and Smith 2005]. Due to the fact that exploration and exploitation in memetic algorithms still can not be measured and hence controlled, many open problems still remain (e.g., how often local search should be applied, which individuals in the population should be exposed to local search? [Hart 1994]). Two selected recent works regarding such open problems are Nguyen et al. [2009] and Lin and Chen [2011]. The former introduced a probabilistic model based on the concept of basin of attraction. Such a model considers exploration and exploitation as independent processes from probabilistic perspectives and introduces a theoretical individual learning intensity upper bound. During runtime, if the average of the intensity of k nearest neighbors is lower than or equal to the upper bound, the upper bound will be increased so that individual learning (i.e., exploitation) may last longer. Conversely, the latter defines those quasi-basin classes, considered as local search zones within the search space, using the concept of subthreshold seekers. As such, a subthreshold seeker starts to exploit a quasi-basin class (represented as a graph) until no neighbor is available and then performs a global search until another subthreshold point is encountered.

—Ursem's [2002] diversity-guided evolutionary algorithm (DGEA) proposed a simple diversity measure to guide exploration and exploitation. Diversity is measured as the sum of Euclidean distance between all individuals and the average point. Such a measure is then used as a threshold to alternate between exploration and exploitation phases. During the exploration phase, only mutation is used in DGEA, whilst during the exploitation phase, only selection and crossover are applied. According to the discussion in Section 2, this is an oversimplified view, and the boundary between both phases is still unclear. This is because the crossover operator can also introduce individuals into an exploration zone, and similarly, the mutation operator can generate individuals within an exploitation zone. However, Ursem assumed that such cases would be rare, and the whole phase is named as an exploration/exploitation phase. DGEA outperformed other algorithms on selective benchmark functions. Ursem's main conclusion was that improvements more or less appeared during the exploitation phase, whilst exploration was indispensable for discovering promising regions.

—Alba and Dorronsoro [2005] proposed an adaptive control of topology in cellular GAs, where an individual can interact only with nearby neighbors. By changing the topology of a grid, a neighborhood is changed and exploration is promoted, whilst exploitation takes place inside each neighborhood using genetic operators. The exploration/exploitation ratio is defined as the quotient between neighborhood and topology radii. They found that similar ratios show similar selection pressures regardless of different topologies. Hence reducing the ratio actually means promoting exploration. Alba and Dorronsoro proposed several deterministic and adaptive approaches for controlling the ratio and, hence, adaptively controlling exploration and exploitation. For example, when convergence speed falls below a specified threshold value, then the grid topology is changed to square, which has a higher ratio, and the evolution process is switched to exploitation. In contrast, when the convergence speed is higher than the specified threshold value, then the grid topology is changed to a narrower shape, which has a lower ratio, and the evolution process is switched to exploration. Such behaviour mimics automated shifting between the exploration and exploitation phases. Adaptive control outperformed tuning and deterministic control, whilst among adaptive control techniques, those

which start with exploration first (narrow-shaped grid) were more successful than others. This again supports our belief that a search process needs to identify promising regions first and then gradually search within the neighborhood to avoid premature convergence. However, exploration and exploitation are again not explicitly defined, and this approach suffers the same problem as others. Namely, even during the exploration phase, an individual can be generated within the exploitation zone, since both phases use the same crossover and mutation operators.

—Črepinšek et al. [2011] introduced an ancestry tree-based approach for explicitly measuring exploration and exploitation. The evolution of a whole population through generations is recorded on an ancestry tree. Different diversity measures (see Section 3) can then be applied to find similarities between individuals on the ancestry tree. Whenever similarity between an individual and its direct offspring is bigger than a predefined threshold value, which defines neighborhood boundary, the ancestry tree is split and forms a new exploitation tree. The root node of the exploitation tree is then obtained by exploration and the other nodes by exploitation. If the root nodes of exploitation trees are connected, an exploration tree is derived. Different exploration and exploitation measures (e.g., progressiveness and the influence of selection pressure) can be obtained by simply identifying various tree characteristics (e.g., width and height). If the exploration and exploitation measures are computed online, then the exploration and exploitation can be controlled.

5. CONCLUSION AND FUTURE DIRECTIONS

Exploration and exploitation are fundamental concepts of any search algorithm, and it is surprising that these concepts are not better understood in EAs. We also noticed in many papers that there are oversimplified beliefs on how EAs explore and exploit the search space. Hence, one of the goals of this article is to encourage a fresh treatment of exploration and exploitation in EAs, namely, those parts of EAs that contribute to exploration and exploitation: selection, variation operators, population size, and representation; how the balance between exploration and exploitation is achieved: implicitly by parameter tuning and parameter control using uniprocess or multiprocess driven approaches; when the balance between exploration and exploitation should be controlled: online using deterministic, adaptive, or self-adaptive approaches; and how the balance between exploration and exploitation can be controlled: by diversity maintenance, diversity control, diversity learning, or other direct approaches.

Exploration and exploitation are conflicting objectives of the search process. Even though tension between exploration and exploitation is a recurring theme in EAs since Holland's seminal work [1975], there are still many open problems which need to be solved. Our survey also showed many opportunities for further work. More specifically, some of the following issues needing further attention.

—First of all, phases of exploration and exploitation must be formally defined by also taking into account problem dependency. New regions and neighborhoods of previously visited points need a precise mathematical definition and metrics.

—Regarding which parts of EAs contribute to exploration and exploitation, more research is needed for understanding how different operators (selection, crossover, mutation) contribute to exploration and exploitation during uniprocess- and multiprocess-driven approaches. In order to solve this problem, new direct measures need to be developed (e.g., [Črepinšek et al. 2011]). Another useful direction for work is to discover good ratios between exploration and exploitation for different types of problems (e.g., single-objective vs. multiobjective) and what kind of operators synergistically achieve such results. Furthermore, the relationship between an individual representation and exploration and exploitation should be better understood, as well

as how the properties of the search space (problem being solved) influence exploration and exploitation. Hence, a formal analysis specifying the *balance equation* [De Jong and Spears 1992] involving all the previously mentioned factors is still unavailable.

—Regarding how the balance between exploration and exploitation is achieved, more research work is needed to understand how the control-parameter setting influences exploration and exploitation. Currently, indirect measures for exploration and exploitation are mostly used, and consequently, control of exploration and exploitation during the run (online) is hard to achieve. By proposing new direct measures for exploration and exploitation, we would be able to better control exploration and exploitation by deterministic, adaptive, or even self-adaptive approaches, thus achieving better balance between exploration and exploitation during different phases of an evolutionary run.

—Regarding when the balance between exploration and exploitation should be controlled, there are plenty of open questions. Even though the control of exploration and exploitation is preferred whilst online, there are still many other issues that require answers. For example, in what order should the phases of exploration and exploitation appear? Intuitively, the exploration phase is usually performed first in order to discover the potential search zones, then the exploitation phase fine-tunes in order to reach a global optimal. Currently, this is not under user control, and the phases of exploration and exploitation are interleaved. However, interleaved phases, where exploration and exploitation are done in parallel, might work quite well or even better than a situation where the phases are separated. This issue is sparsely covered in EA research currently and needs more attention. Besides the orders of both phases, an open issue is how often they should appear in consequence. Should these phases alternate within each generation? Should they be present within each generation? Another unresolved problem is whether control parameters and whole populations should be preserved when we switch from one phase to the other.

—Regarding how the balance between exploration and exploitation can be controlled, we envisage that many sophisticated approaches will be developed over the forthcoming years. More research work is needed, especially in the areas of diversity learning and direct approaches to control exploration and exploitation. One of the urgent steps for future research work is to better understand the influence of diversity for achieving good balance between exploration and exploitation. Theoretical analysis of more advanced diversification techniques (see Section 4.1) are needed [Friedrich et al. 2007, 2008].

—Since direct measures for exploration and exploitation are rare, we need to investigate how effective the proposed measures are for explicit control of exploration and exploitation. By using direct measures for the exploration and exploitation comparisons, the explorative and exploitative power of different EAs would become possible. For example, an interesting examination would be to compare the exploration and exploitation powers of a single-objective EA for unimodal and multimodal problems, comparing behavior of EMO on two-objectives versus many-objectives problems, or an examination of exploration and exploitation in the context of dynamic/adaptive problems. An open issue is whether direct measures for exploration and exploitation can further improve the performances of the more competitive EAs. Moreover, many other different analyses are realizable: analysis of different parameter controls for exploration and exploitation, analysis of different diversity maintenance/diversity control/diversity learning approaches to exploration and exploitation, etc.

We hope that this work will trigger more interest in these issues and more research will be devoted to the fundamental understanding of how EAs explore and exploit the search space. We are convinced that such knowledge would help EA researchers

and practitioners to better understand a particular EA, to improve it, or to develop a new one.

REFERENCES

ADRA, S. F. AND FLEMING, P. J. 2011. Diversity management in evolutionary many-objective optimization. *IEEE Trans. Evol. Comput. 15,* 2, 183–195.

ALBA, E. AND DORRONSORO, B. 2005. The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *IEEE Trans. Evol. Comput. 9,* 3, 126–142.

AMOR, H. B. AND RETTINGER, A. 2005. Intelligent exploration for genetic algorithms: Using self-organizing maps in evolutionary computation. In *Proceedings of the 7th Genetic and Evolutionary Computation Conference*. 1531–1538.

ARAUJO, L. AND MERELO, J. J. 2011. Diversity through multiculturality: Assessing migrant choice policies in an island model. *IEEE Trans. Evol. Comput. 15,* 4, 456–468.

BÄCK, T. 1994. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proceedings of the 1st Conference on Evolutionary Computing*. 57–62.

BÄCK, T. 1996. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press.

BÄCK, T., EIBEN, A. E., AND VAN DER VAART, N. A. L. 2000. An empirical study on gas without parameters. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, 315–324.

BÄCK, T. AND SCHWEFEL, H.-P. 1993. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput. 1,* 1, 1–23.

BARTZ-BEIELSTEIN, T., LASARCZYK, C. W. G., AND PREUSS, M. 2005. Sequential parameter optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 773–780.

BECERRA, R. L. AND COELLO COELLO, C. A. 2006. Cultured differential evolution for constrained optimization. *Comput. Methods Appl. Mech. Eng. 195,* 33–36, 4303–4322.

BERSANO-BEGEY, T. 1997. Controlling exploration, diversity and escaping local optima in GP: Adopting weights of training sets to model resource consumption. In *Proceedings of the Late Breaking Papers at the Genetic Programming Conference*. 7–10.

BEYER, H.-G. AND DEB, K. 2001. On self-adaptive features in real-parameter evolutionary algorithms. *IEEE Trans. Evol. Comput. 5,* 3, 250–270.

BIRATTARI, M., STÜTZLE, T., PAQUETE, L., AND VARRENTRAPP, K. 2002. A racing algorithm for configuring metaheuristics. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 11–18.

BLUM, C., PUCHINGER, J., RAIDL, G. A., AND ROLI, A. 2011. Hybrid metaheuristics in combinatorial optimization: A survey. *Appl. Soft Comput. 11,* 6, 4135–4151.

BLUM, C. AND ROLI, A. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv. 35,* 3, 268–308.

BOGON, T., POURSANIDIS, G., LATTNER, A. D., AND TIMM, I. J. 2011. Extraction of function features for an automatic configuration of particle swarm optimization. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*. 51–60.

BOSMAN, P. AND THIERENS, D. 2003. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans. Evol. Comput. 7,* 2, 174–188.

BREST, J., GREINER, S., BOŠKOVIČ, B., MERNIK, M., AND ŽUMER, V. 2006. Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems. *IEEE Trans. Evol. Comput. 10,* 6, 646–657.

BURKE, E., GUSTAFSON, S., AND KENDALL, G. 2004. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Trans. Evol. Comput. 8,* 1, 47–62.

BURKE, E., GUSTAFSON, S., KENDALL, G., AND KRASNOGOR, N. 2002. Advanced population diversity measures in genetic programming. In *Proceedings of Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 2439, Springer, 341–350.

CALEGARY, P., CORAY, G., HERTZ, A., KOBLER, D., AND KUONEN, P. 1999. A taxonomy of evolutionary algorithms in combinatorial optimization. *J. Heuristics 5,* 2, 145–158.

CHAIYARATANA, N., PIROONRATANA, T., AND SANGKAWELERT, N. 2007. Effects of diversity control in single-objective and multi-objective genetic algorithms. *J. Heuristics 13,* 1–34.

CHEN, G., LOW, C. P., AND YANG, Z. 2009. Preserving and exploiting genetic diversity in evolutionary programming algorithms. *IEEE Trans. Evol. Comput. 13,* 3, 661–673.

CHOW, C. K. AND YUEN, S. Y. 2011. An evolutionary algorithm that makes decision based on the entire previous search history. *IEEE Trans. Evol. Comput. 15,* 6, 741–769.

COBB, H. G. AND GREFENSTETTE, J. J. 1993. Genetic algorithms for tracking changing environments. In *Proceedings of the 5th International Conference on Genetic Algorithms*. 523–530.

ČREPINŠEK, M., MERNIK, M., AND LIU, S.-H. 2011. Analysis of exploration and exploitation in evolutionary algorithms by ancestry trees. *Int. J. Innovative Comput. Appl. 3,* 1, 11–19.

CURRAN, D. AND O'RIORDAN, C. 2006. Increasing population diversity through cultural learning. *Adapt. Behav. 14,* 4, 315–338.

CZARN, A., MACNISH, C., VIJAYAN, K., TURLACH, B., AND GUPTA, R. 2004. Statistical exploratory analysis of genetic algorithms. *IEEE Trans. Evol. Comput. 8,* 4, 405–421.

DARWEN, P. J. AND YAO, X. 2001. Why more choices cause less cooperation in iterated prisoner's dilemma. In *Proceedings of the Congress of Evolutionary Computation*. 987–994.

DE JONG, E. D., WATSON, R. A., AND POLLACK, J. B. 2001. Reducing bloat and promoting diversity using multi-objective methods. In *Proceedings of the 3rd Genetic and Evolutionary Computation Conference*. 11–18.

DE JONG, K. A. 1975. An analysis of the behavior of a class of genetic adaptive systems. Ph.D. dissertation, University of Michigan, Ann Arbor, MI.

DE JONG, K. A. 2002. *Evolutionary Computation*. MIT Press, Cambridge, MA.

DE JONG, K. A. AND SPEARS, W. 1992. A formal analysis of the role of multi-point crossover in genetic algorithms. *Ann. Math. Artif. Intell. 5,* 1, 1–26.

DEB, K. AND GOLDBERG, D. E. 1989. An investigation of niche and species formation in genetic function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. 42–50.

D'HAESELEER, P. AND BLUMING, J. 1994. *Advances in Genetic Programming*. MIT Press, Cambridge, MA.

DRÉO, J. 2009. Using performance fronts for parameter setting of stochastic metaheuristics. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*. 2197–2200.

EIBEN, A. E., HINTERDING, R., AND MICHALEWICZ, Z. 1999. Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput. 3,* 2, 124–141.

EIBEN, A. E., MARCHIORI, E., AND VALKO, V. A. 2004. Evolutionary algorithms with on-the-fly population size adjustment. In *Proceedings of Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 3242, Springer, 41–50.

EIBEN, A. E. AND SCHIPPERS, C. 1998. On evolutionary exploration and exploitation. *Fundamenta Informaticae 35,* 35–50.

EIBEN, A. E. AND SMIT, S. K. 2011. Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol. Comput. 1,* 1, 19–31.

EIBEN, A. E. AND SMITH, J. E. 2008. *Introduction to Evolutionary Computing*. Springer, Berlin.

ESHELMAN, L. J. 1991. The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination. *Found. Genetic Algorith. 1,* 265–283.

ESHELMAN, L. J. AND SCHAFFER, J. 1991. Preventing premature convergence in genetic algorithms by preventing incest. In *Proceedings of the 4th International Conference on Genetic Algorithms*. 115–122.

FERNANDEZ-PRIETO, J. A., CANADA-BAGO, J., GADEO-MARTOS, M. A., AND VELASCO, J. R. 2011. Optimisation of control parameters for genetic algorithms to test computer networks under realistic traffic loads. *Appl. Soft Comput. 11,* 4, 3744–3752.

FISTER, I., MERNIK, M., AND FILIPIČ, B. 2010. A hybrid self-adaptive evolutionary algorithm for marker optimization in the clothing industry. *Appl. Soft Comput. 10,* 2, 409–422.

FOGARTY, T. C. 1989. Varying the probability of mutation in the genetic algorithm. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. 104–109.

FOGEL, L. J. 1999. *Intelligence Through Simulated Evolution: Forty Years of Evolutionary Programming*. Wiley, Itoboken, NJ.

FONSECA, C. M. AND FLEMING, P. J. 1995. Multiobjective genetic algorithms made easy: Selection, sharing and mating restriction. In *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. 45–52.

FREISLEBEN, B. AND MERZ, P. 1996. A genetic local search algorithm for solving symmetric and asymmetric traveling salesman problems. In *Proceedings of the International Conference on Evolutionary Computation*. 616–621.

FRIEDRICH, T., HEBBINGHAUS, N., AND NEUMANN, F. 2007. Rigorous analyses of simple diversity mechanisms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1219–1225.

FRIEDRICH, T., OLIVETO, P. S., SUDHOLT, D., AND WITT, C. 2008. Theoretical analysis of diversity mechanisms for global exploration. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 945–952.

GALVÁN-LÓPEZ, E., MCDERMOTT, J., O'NEILL, M., AND BRABAZON, A. 2010. Towards an understanding of locality in genetic programming. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*. 901–908.

GAO, H. AND XU, W. 2011. Particle swarm algorithm with hybrid mutation strategy. *Appl. Soft Comput. 11,* 8, 5129–5142.

GEN, M. AND CHENG, R. 1997. *Genetic Algorithms and Engineering Design.* John Wiley and Sons, Itoboken, NJ.

GHOSH, A., TSUTSUI, S., AND TANAKA, H. 1996. Individual aging in genetic algorithms. In *Proceedings of the Australian New Zealand Conference on Intelligent Information Systems*. 276–279.

GOH, K. S., LIM, A., AND RODRIGUES, B. 2003. Sexual selection for genetic algorithms. *Artif. Intell. Rev. 19*, 123–152.

GOLDBERG, D. E. 2008. *Genetic Algorithms in Search, Optimization and Machine Learning*. Dorling Kindersley, London.

GOLDBERG, D. E. AND DEB, K. 1991. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*. Morgan Kaufmann, Burlington, MA, 69–93.

GOLDBERG, D. E. AND RICHARDSON, J. 1987. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the 2nd International Conference on Genetic Algorithms*. 41–49.

GONG, W., CAI, Z., AND JIANG, L. 2008. Enhancing the performance of differential evolution using orthogonal design method. *Appl. Math. Comput. 206,* 1, 56–69.

GREENWOOD, G. W., FOGEL, G. B., AND CIOBANU, M. 1999. Emphasizing extinction in evolutionary programming. In *Proceedings of the Congress of Evolutionary Computation*. 666–671.

GREFENSTETTE, J. J. 1986. Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst., Man Cybernetics 16,* 1, 122–128.

GREFENSTETTE, J. J. 1992. Genetic algorithms for changing environments. In *Proceedings of Parallel Problem Solving from Nature*, Elsevier, Amsterdam, 137–144.

HARIK, G. R. 1995. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*. 24–31.

HARIK, G. R. AND LOBO, F. 1999. A parameter-less genetic algorithm. Tech. rep., University of Illinois at Urbana-Champaign, IL.

HARIK, G. R., LOBO, F., AND GOLDBERG, D. E. 1999. The compact genetic algorithm. *IEEE Trans. Evol. Comput. 3,* 4, 287–297.

HART, W. E. 1994. Adaptive global optimization with local search. Ph.D. dissertation, University of California, San Diego, CA.

HERRERA, F. AND LOZANO, M. 1996. Adaptation of genetic algorithm parameters based on fuzzy logic controllers. *Genetic Algorith. Soft Comput.* 95–125.

HESSER, J. AND MÄNNER, R. 1991. Toward an optimal mutation probability for genetic algorithms. In *Proceedings of the 1st Conference of Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 496, Springer, 23–32.

HOLLAND, J. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.

HORN, J., NAFPLIOTIS, N., AND GOLDBERG, D. E. 1994. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*. 82–87.

HUTTER, M. AND LEGG, S. 2006. Fitness uniform optimization. *IEEE Trans. Evol. Comput. 10,* 5, 568–589.

ISHIBUCHI, H., HITOTSUYANAGI, Y., WAJAMATSU, Y., AND NOJIMA, Y. 2010a. How to choose solutions for local search in multiobjective combinatorial memetic algorithms. In *Proceedings of the 11th Conference of Parallel Problem Solving from Nature: Part I*, Lecture Notes in Computer Science, vol. 6238, Springer, 516–525.

ISHIBUCHI, H., NARUKAWA, K., TSUKAMOTO, N., AND NOJIMA, Y. 2008. An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization. *Europ. J. Oper. Res. 188,* 1, 57–75.

ISHIBUCHI, H., TSUKAMOTO, N., AND NOJIMA, Y. 2010b. Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization. *IEEE Trans. Evol. Comput. 14,* 6, 985–998.

ISHIBUCHI, H., YOSHIDA, T., AND MURATA, T. 2003. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. *IEEE Trans. Evol. Comput. 7,* 2, 204–223.

JIA, D., ZHENG, G., AND KHAN, M. K. 2011. An effective memetic differential evolution algorithm based on chaotic local search. *Inform. Sci. 181,* 15, 3175–3187.

JIN, X. AND REYNOLDS, R. 1999. Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: A cultural algorithm approach. In *Proceedings of the Congress on Evolutionary Computation*. 1672–1678.

JOAN-ARINYO, R., LUZON, M. V., AND YEGUAS, E. 2011. Parameter tuning of pbil and chc evolutionary algorithms applied to solve the root identification problem. *Appl. Soft Comput. 11,* 1, 754–767.

JOSE-REVUELTA, L. M. S. 2007. A new adaptive genetic algorithm for fixed channel assignment. *Inform. Sci. 177*, 2655–2678.

KOHONEN, T. 2001. *Self-Organizing Maps* 3rd Ed. Springer-Verlag, New York, NY.

KOUMOUSIS, V. AND KATSARAS, C. P. 2006. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. *IEEE Trans. Evol. Comput. 10,* 1, 19–28.

KOZA, J. R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA.

KRASNOGOR, N. AND SMITH, J. 2005. A tutorial for competent memetic algorithms: Model, taxonomy, and design issues. *IEEE Trans. Evol. Comput. 9,* 5, 474–488.

KRINK, T., RICKERS, P., AND THOMSEN, R. 2000. Applying self-organised criticality to evolutionary algorithms. In *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, Springer-Verlag, New York, NY, 375–384.

LANGDON, W. B. 1998. *Data Structures and Genetic Programming: Genetic Programming + Data Structures = Automatic Programming*. Kluwer, Alphen aan den Rjin, Netherlands.

LEE, J.-Y., KIM, M.-S., AND LEE, J.-J. 2011. Compact genetic algorithm using belief vectors. *Appl. Soft Comput. 11,* 4, 3385–3401.

LEUNG, Y., GAO, Y., AND XU, Z. 1997. A perspective on premature convergence in genetic algorithms and its markov chain analysis. *Trans. Neural Netw. 8,* 5, 1165–1176.

LEUNG, Y. AND WANG, Y. 2001. An orthogonal genetic algorithm with quantization for global numerical optimization. *IEEE Trans. Evol. Comput. 5,* 1, 41–53.

LI, J.-P., BALAZS, M. E., PARKS, G. T., AND CLARKSON, J. P. 2002. A species conserving genetic algorithm for multimodal function optimization. *Evol. Comput. 10 3*, 207–234.

LI, M., CAI, Z., AND SUN, G. 2004. An adaptive genetic algorithm with diversity-guided mutation and its global convergence property. *J. Central South Univ. Technol. 11,* 3, 323–327.

LI, Z. AND WANG, X. 2011. Chaotic differential evolution algorithm for solving constrained optimization problems. *Inform. Technol. J. 10,* 12, 2378–2384.

LIANG, Y. AND LEUNG, K.-S. 2011. Genetic algorithm with adaptive elitist-population strategies for multimodal function optimization. *Appl. Soft Comput. 11,* 2, 2017–2034.

LIAO, T. W. 2010. Two hybrid differential evolution algorithms for engineering design optimization. *Appl. Soft Comput. 10,* 4, 1188–1199.

LIN, J.-Y. AND CHEN, Y.-P. 2011. Analysis on the collaboration between global search and local search in memetic computation. *IEEE Trans. Evol. Comput. 15,* 5, 608–622.

LIU, S.-H., MERNIK, M., AND BRYANT, B. R. 2004. Parameter control in evolutionary algorithms by domain-specific scripting language PPC_{EA}. In *Proceedings of the International Conference on Bioinspired Optimization Methods and their Applications*. 41–50.

LIU, S.-H., MERNIK, M., AND BRYANT, B. R. 2007. A clustering entropy-driven approach for exploring and exploiting noisy functions. In *Proceedings of the 22nd ACM Symposium on Applied Computing*. 738–742.

LIU, S.-H., MERNIK, M., AND BRYANT, B. R. 2009. To explore or to exploit: An entropy-driven approach for evolutionary algorithms. *Int. J. Knowl. Intell. Eng. Syst. 13,* 3, 185–206.

LOBO, F. J., LIMA, C. F., AND MICHALEWICZ, Z. 2007. *Parameter Setting in Evolutionary Algorithms*. Springer, Berlin.

LOZANO, M., HERRERA, F., AND CANO, J. R. 2008. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Inform. Sci. 178,* 23, 4421–4433.

LUERSSEN, M. H. 2005. Phenotype diversity objectives for graph grammar evolution. In *Recent Advances in Artificial Life*, World Scientific Publishing, Singapore, 159–170.

MAHFOUD, S. W. 1995. Niching methods for genetic algorithms. Tech. rep., University of Illinois at Urbana Champaign, IL.

MAJIG, M. AND FUKUSHIMA, M. 2008. Adaptive fitness function for evolutionary algorithm and its applications. In *Proceedings of the International Conference on Informatics Education and Research for Knowledge-Circulating Society*. 119–124.

MALLIPEDDI, R., SUGANTHAN, P. N., PAN, Q. K., AND TASGETIREN, M. F. 2011. Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput. 11,* 2, 1679–1696.

MARTIN, W. N., LIENIG, J., AND COHOON, J. P. 1999. *Island (Migration) Models: Evolutionary Algorithms based on Punctuated Equilibria (In Handbook of Evolutionary Computation)*. Oxford University Press.

MASHINCHI, M. H., ORGUN, M. A., AND PEDRYCZ, W. 2011. Hybrid optimization with improved tabu search. *Appl. Soft Comput. 11,* 2, 1993–2006.

MASISI, L., NELWAMONDO, V., AND MARWALA, T. 2008. The use of entropy to measure structural diversity. In *Proceedings of the IEEE International Conference on Computational Cybernetics*. 41–45.

MATSUI, K. 1999. New selection method to improve the population diversity in genetic algorithms. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*. 625–630.

MATTIUSSI, C., WAIBEL, M., AND FLOREANO, D. 2004. Measures of diversity for populations and distances between individuals with highly reorganizable genomes. *Evol. Comput. 12,* 4, 495–515.

MAULDIN, M. L. 1984. Maintaining diversity in genetic search. In *Proceedings of the National Conference on Artificial Intelligence*. 247–250.

MCGINLEY, B., MAHER, J., O'RIORDAN, C., AND MORGAN, F. 2011. Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *IEEE Trans. Evol. Computat. 15,* 5, 692–714.

MCPHEE, N. F. AND HOPPER, N. J. 1999. Analysis of genetic diversity through population history. In *Proceedings of the 1st Genetic and Evolutionary Computation Conference*. 1112–1120.

MENGSHOEL, O. J. AND GOLDBERG, D. E. 1999. Probabilistic crowding: Deterministic crowding with probabilisitic replacement. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 409–416.

MERNIK, M., HEERING, J., AND SLOANE, A. 2005. When and how to develop domain-specific languages. *ACM Comput. Sur. 37,* 4, 316–344.

MERZ, P. AND FREISLEBEN, B. 2000. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Trans. Evol. Comput. 4,* 4, 337–352.

MICHALEWICZ, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, NY.

MISEVIČIUS, A. 2011. Generation of grey patterns using an improved genetic-evolutionary algorithm: Some new results. *Inform. Technol. Control 40,* 4, 330–343.

MISEVIČIUS, A. AND RUBLIAUSKAS, D. 2008. Enhanced improvement of individuals in genetic algorithms. *Inform. Technol. Control 37,* 3, 179–186.

MONGUS, D., REPNIK, B., MERNIK, M., AND ŽALIK, B. 2012. A hybrid evolutionary algorithm for tuning a cloth-simulation model. *Appl. Soft Comput. 12,* 1, 266–273.

MONTERO, E. AND RIFF, M.-C. 2011. On-the-fly strategies for evolutionary algorithms. *Inform. Sci. 181*, 552–566.

MORAGLIO, A., KIM, Y.-H., YOON, Y., AND MOON, B.-R. 2007. Geometric crossovers for multiway graph partitioning. *Evol. Comput. 15,* 4, 445–474.

MORI, N., YOSHIDA, J., TAMAKI, H., KITA, H., AND NISHIKAWA, Y. 1995. A thermodynamical selection rule for the genetic algorithm. In *Proceedings of the 2nd IEEE International Conference on Evolutionary Computation*. 188–192.

MOSCATO, P. 1999. Memetic algorithms: A short introduction. In *New Ideas in Optimization*, McGraw Hill Ltd., Maidenhead, U.K., 219–234.

MÜHLENBEIN, H. AND PAASS, G. 1996. From recombination of genes to the estimation of distributions I. binary parameters. In *Proceedings of Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 1141, Springer, 178–187.

NANNEN, V. AND EIBEN, A. E. 2006. A method for parameter calibration and relevance estimation in evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. 183–190.

NGUYEN, Q. H., ONG, Y.-S., AND LIM, M. H. 2009. A probabilistic memetic framework. *IEEE Trans. Evol. Comput. 13,* 3, 604–623.

OCHOA, G., BILGIN, B., AND KORKMAZ, E. E. 2008. A comprehensive analysis of hyper-heuristics. *Intell. Data Anal. 12,* 1, 3–23.

ONG, Y.-S., LIM, M.-H., ZHU, N., AND WONG, K.-W. 2006. Classification of adaptive memetic algorithms: A comparative study. *IEEE Trans. Syst. Man Cybernetics (Part B)*, 141–152.

OPPACHER, F. AND WINEBERG, M. 1999. The shifting balance ga: Improving the ga in dynamic environment. In *Proceedings of the 1st Genetic and Evolutionary Computation Conference*. 504–510.

PAENKE, I., JIN, Y., AND BRANKE, J. 2009. Balancing population- and individual-level adaptation in changing environments. *Adapt. Behav. 17,* 2, 153–174.

PAN, Q.-K., SUGANTHAN, P. N., WANG, L., GAO, L., AND MALLIPEDDI, R. 2011. A differential evolution algorithm with self-adapting strategy and control parameters. *Comput. Oper. Res. 38,* 1, 394–408.

PETROWSKI, A. 1996. A clearing procedure as a niching method for genetic algorithms. In *Proceedings of the IEEE International Conference on Evolutionary Computation*. 798–803.

QIN, A. K., HUANG, V. L., AND SUGANTHAN, P. N. 2009. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput. 13,* 2, 398–417.

RAHNAMAYAN, S., TIZHOOSH, H. R., AND SALAMA, M. M. A. 2008. Opposition-based differential evolution. *IEEE Trans. Evol. Comput. 12,* 1, 64–79.

RAMSEY, C. L. AND GREFENSTETTE, J. J. 1993. Case-based initialization of genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*. 84–91.

RONALD, E. 1995. When selection meets seduction. In *Proceedings of the 6th International Conference on Genetic Algorithms*. 167–173.

ROSCA, J. 1995. Entropy-driven adaptive representation. In *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, 23–32.

SARENI, B. AND KRÄHENBÜHL, L. 1998. Fitness sharing and niching methods revisited. *IEEE Trans. Evol. Comput. 2,* 3, 97–106.

SCHAFFER, J. D., CARUANA, R. A., ESHELMAN, L. J., AND DAS, R. 1989. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. 51–60.

SHIMODAIRA, H. 1997. Dcga: A diversity control oriented genetic algorithm. In *Proceedings of the 2nd International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. 444–449.

SINGH, G. AND DEB, K. 2006. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. 1305–1312.

SMIT, S. K. AND EIBEN, A. E. 2009. Comparing parameter tuning methods for evolutionary algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation*. 399–406.

SMITH, J. E. AND FOGARTY, T. C. 1997. Operator and parameter adaptation in genetic algorithms. *Soft Comput. 1,* 2, 81–87.

SMITH, R. E., FORREST, S., AND PERELSON, A. S. 1993. Searching for diverse, cooperative subpopulations with genetic algorithms. *Evol. Comput. 1,* 2, 127–149.

SMITH, R. E. AND SMUDA, E. 1995. Adaptively resizing populations: Algorithm, analysis, and first results. *Complex Syst. 9*, 47–72.

SOZA, C., BECERRA, R. L., RIFF, M. C., AND COELLO COELLO, C. A. 2011. Solving timetabling problems using a cultural algorithm. *Appl. Soft Comput. 11,* 1, 337–344.

SPEARS, W. M. 1995. Adapting crossover in evolutionary algorithms. In *Proceedings of the Evolutionary Programming Conference*. 367–384.

SRINIVAS, M. AND PATNAIK, L. M. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. Syst. Man Cybernetics 24*, 656–667.

STORCH, T. 2004. On the choice of the population size. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 748–760.

STORN, R. AND PRICE, K. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim. 11*, 341–359.

TALBI, E. G. 2002. A taxonomy of hybrid metaheuristics. *J. Heuristics 8,* 5, 541–564.

TOFFOLO, A. AND BENINI, E. 2003. Genetic diversity as an objective in multi-objective evolutionary algorithms. *Evol. Comput. 11,* 2, 151–167.

TSUJIMURA, Y. AND GEN, M. 1998. Entropy-based genetic algorithm for solving tsp. In *Proceedings of the 2nd International Conference on Knowledge-Based Intelligent Electronic Systems*. 285–290.

TSUTSUI, S., FUJIMOTO, Y., AND GHOSH, A. 1997a. Forking genetic algorithms: GAs with search space division schemes. *Evol. Comput. 5,* 1, 61–80.

TSUTSUI, S., GHOSH, A., CORNE, D., AND FUJIMOTO, Y. 1997b. A real coded genetic algorithm with an explorer and an exploiter populations. In *Proceedings of the 7th International Conference on Genetic Algorithms*. 238–245.

URSEM, R. 2000. Multinational GAs: Multimodal optimization techniques in dynamic environments. In *Proceedings of the 2nd Genetic and Evolutionary Computation Conference*. 19–26.

URSEM, R. 2002. Diversity-guided evolutionary algorithms. In *Proceedings of Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 2439, Springer, 462–471.

WANG, Y., CAI, Z., AND ZHANG, Q. 2012. Enhancing the search ability of differential evolution through orthogonal crossover. *Inform. Sci. 185,* 1, 153–177.

WATSON, J., BAKER, T., BELL, S., GANN, A., LEVINE, M., AND LOSICK, R. 2004. *Molecular Biology of the Gene*. Benjamin Cummings, San Francisco, CA.

WHITLEY, D., MATHIAS, K., AND FITZHORN, P. 1991. Delta coding: An iterative search strategy for genetic algorithms. In *Proceedings of the 4th International Conference on Genetic Algorithms*. 77–84.

WHITLEY, D. AND STARKWEATHER, D. 1990. Genitor-ii: A distributed genetic algorithm. *J. Exp. Theor. Artif. Intell. 2,* 3, 189–214.

WINEBERG, M. AND OPPACHER, F. 2003. Distance between populations. In *Proceedings of the International Conference on Genetic and Evolutionary Computation*. 1481–1492.

WONG, Y.-Y., LEE, K.-H., LEUNG, K.-S., AND HO, C.-W. 2003. A novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Comput. 7*, 506–515.

YANG, S. 2008. Genetic algorithms with memory- and elitism-based immigrants in dynamic environments. *Evol. Comput. 16,* 3, 385–416.

YAO, X., LIU, Y., AND LIN, G. 1999. Evolutionary programming made faster. *IEEE Trans. Evol. Comput. 3,* 2, 82–102.

YIN, X. AND GERMAY, N. 1993. A fast genetic algorithm with sharing scheme using cluster analysis method in multi-modal function optimization. In *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*. 450–457.

YU, E. L. AND SUGANTHAN, P. N. 2010. Ensemble of niching algorithms. *Inform. Sci. 180,* 15, 2815–2833.

YUEN, S. Y. AND CHOW, C. K. 2009. A genetic algorithm that adaptively mutates and never revisits. *IEEE Trans. Evol. Comput. 13,* 2, 454–472.

ZHANG, J. AND SANDERSON, A. C. 2009. Jade: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput. 13,* 5, 945–957.

ZHAO, X. 2011. Simulated annealing algorithm with adaptive neighborhood. *Appl. Soft Comput. 11,* 2, 1827–1836.

ZIELINSKI, K., WEITKEMPER, P., LAUR, R., AND KAMMEYER, K.-D. 2009. Optimization of power allocation for interference cancellation with particle swarm optimization. *IEEE Trans. Evol. Comput. 8,* 2, 128–150.

ZITZLER, E., DEB, K., AND THIELE, L. 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput. 8,* 2, 173–195.

ZITZLER, E., LAUMANNS, M., AND THIELE, L. 2002. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evol. Meth. Design: Optim. Control*, 95–100.

ZITZLER, E. AND THIELE, L. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput. 3,* 4, 257–271.