Evolutionary Optimization: Pitfalls and Booby Traps

Thomas Weise[1], Member, IEEE, Raymond Chiong[2], Member, IEEE, and Ke Tang[1], Member, IEEE

[1] *Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China (USTC), Huangshan Road/Feixi Road, Hefei 230027, Anhui, China*

[2] *Faculty of Higher Education, Swinburne University of Technology, 50 Melba Avenue, Lilydale, Victoria 3140, Australia*

E-mail: tweise@ustc.edu.cn, rchiong@swin.edu.au, ketang@ustc.edu.cn

Received August 31, 2011

Abstract Evolutionary Computation (EC), a collective name for a range of metaheuristic black-box optimization algorithms, is one of the fastest-growing areas in computer science. Many manuals and "how-to"s on the use of different EC methods as well as a variety of free or commercial software libraries are widely available nowadays. However, when one of these methods is applied to a real-world task, there can be many pitfalls and booby traps lurking – certain aspects of the optimization problem that may lead to unsatisfactory results even if the algorithm appears to be correctly implemented and executed. These include the convergence issues, ruggedness, deceptiveness, and neutrality in the fitness landscape, epistasis, non-separability, noise leading to the need for robustness, as well as dimensionality and scalability issues, among others. In this article, we systematically discuss these related hindrances and present some possible remedies. The goal is to equip practitioners and researchers alike with a clear picture and understanding of what kind of problems can render EC applications unsuccessful and how to avoid them from the start.

Keywords Evolutionary Computing, Difficulties, Optimization, Problems, Countermeasures

1 Introduction

Every task with the goal of finding certain configurations considered as best in the context of pre-defined criteria can be viewed as an optimization problem. If these problems are formally specified, they can be solved algorithmically either with a dedicated, problem-specific algorithm (such as Dijkstra's algorithm for finding shortest path trees on graphs) or with a more general optimization method. The set of optimization algorithms ranges from mathematical (e.g., using Lagrange Multipliers), numerical (e.g., the Regula Falsi) and simple heuristic (e.g., A*-search) approaches to randomized metaheuristics such as the Evolutionary Computation (EC) methods. The latter is the focus of this special issue.

When skimming through the articles in this issue, the reader will find many successful examples and variants of different EC techniques.[1] However, questions such as these may arise: "Why are there so many different optimization methods?", "Is optimization a complicated process? If so, why?", "What makes an optimization problem difficult to solve?", "Which are the things I should consider when tackling a particular optimization task?", and so on. In this article, our aim is to provide some answers to these questions by discussing a list of fundamental issues that are often seen as "obstacles" in the evolutionary optimization domain.

To start with, there are many design decisions in implementing EC methods. For effective optimization, it is important to understand not only the problem being studied, but also how that problem interacts with the applied technique(s). Design choices that do not address issues related to convergence, ruggedness, deceptiveness and neutrality in the fitness landscape, epistasis, non-separability, noise, dimensionality, scalability and so on can hamper the effectiveness of the optimization effort. By using clear definitions and illustrations to describe these fundamental issues, we hope to increase awareness among computer scientists and practitioners about how to avoid pitfalls and how EC can be applied more efficiently in real-world environments (see [5]).

It is necessary to note that this article is not intended to be a tutorial of how to apply a particular EC method, e.g., an Evolutionary Algorithm (EA), to specific problems [6–8] or how to address subject matters such as multi-objectivity [9], constraint handling [10], or the inclusion of problem-specific knowledge [11]. For the practical application of EC methods in general and EAs in particular, several books exist [5, 7, 12–14]. Instead, our aim is to take a closer look at what features of the problem or search space may decrease the solution quality even if the algorithm implementation appears to be correct and "make sense". Considering these features (and corresponding countermeasures) before developing an EA application (or, at least, when trying to improve its performance) may lead to significantly better results.

The article is also not a survey on problem complexity. Research studies on this topic are typically carried out from an analytical, mathematical, or theoretical perspective, with the goal to derive approximations for the expected runtime of the problem solvers [15–17]. These approaches usually focus on benchmark problems or specific classes of optimization tasks, but there is also progress towards developing more general theorems [17, 18]. Here, we do not intend to provide a rigorous theoretical treatment of pitfalls and possible traps in evolutionary optimization, but simply to present a top-down view of some "complications" that may be encountered during the optimization process.

Our focus is therefore on the design decisions of EC methods. The effectiveness of these design decisions is often influenced by their actual implementation and the associated parameter values used in the optimization process. While parameter values are important for gaining the most benefit from an EC implementation, design decisions such as problem representation, operator design, and population structure are often considered to be even more critical [19, 20].

In the remainder of this section, we will introduce the basic terminologies used throughout this article, describe some possible scenarios of the fitness landscape, and briefly discuss complexity theory. After which, we start off with the topic of convergence in Section 2, fol-

[1] A detailed overview of different EC techniques can be found in [2–4].

Table 1: Overview on topics and measures.

↓ **Problems** ↓ → **Countermeasures** →	Representation Design	Operator Design	Increase Exploration	Increase Exploitation	Restarting	Lower Selection Pressure	Higher Selection Pressure	Larger Population Size	Sharing & Niching	Clustering of Population	Self-Adaptation	Multi-Objectivization	Landscape Approximation	Hybrid Algorithms	Combined Algorithms	Use of Memory	Preventing Convergence	Novelty Search	Linkage Learning	Randomizing Objectives	Use Multiple Archives	Use Indicator Functions	Scalarizing	Limiting Search Area	Parallelization	Developmental Represent.	Adaptive Encodings	Exploiting Separability	↓ **Sections** ↓
Premature Convergence		✓	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓					✓	✓											2.1.1
Slow Convergence			✗	✓		✗	✓	✗			✓			✓											✓				2.1.1, 1.3
Bad Spread			✓	✗		✓	✗	✓	✓	✓							✓												2.1.1
Multi-Modality			✓	✗	✓	✓	✗	✓	✓	✓							✓												3, 2.1.1
Ruggedness	✓	✓			✓	✓	✗	✓	✓	✓			✓	✓	✓		✓												3
Deceptiveness	✓	✓	✓	✗		✓			✓	✓	✓					✓	✓	✓	✓										4
Neutrality	✓	✓					✓	✓	✓			✓		✓		✓													5
Needle-in-a-Haystack	✓													✓															5.1.3
Epistasis	✓	✓	✓	✗			✓	✓									✓		✓										6
Noise																				✓									7
Dimensionality ($\vec{f}$)							✓	✓													✓	✓	✓	✓					8
Scale ($\mathbb{G}$)	✓										✓								✓						✓	✓	✓	✓	9
Legend: ✓ means that the given measure maybe useful if the problem in the same row is observed, ✗ means that the measure would likely be counterproductive. → **Sections** →	1.2, 3.2.4, 4.2.1, 5.1.2, 6.2.1, 5.2.2	2.1.3, 2.2.2, 3.2.4	2.1.3, 2.2.1	2.1.3, 2.2.1	2.2.3	2.1.3, 2.2.4	2.1.3, 6.2.2, 8.2.3	2.2.4, 8.2.1	2.2.5, 4.2.2	2.2.6	2.2.7	2.2.8	3.2.2	3.2.1	3.2.3	4.2.2	4.2.3	4.2.4	6.2.3	7.2	8.2.2	8.2.4	8.2.5	8.2.6	9.2.1	9.2.2	9.2.4	9.2.5	This table gives an overview on all the problems and countermeasures discussed in this article.

lowed by other issues possibly leading to unsatisfying convergence, such as ruggedness (Section 3), deceptiveness (Section 4), or neutrality (Section 5) in the fitness landscape. One way ruggedness, neutrality, and deceptiveness can be manifested is from the genotype-phenotype mapping through a phenomenon known as epistasis (Section 6). Optimization can also become more complicated if solutions that are sought have to be robust against noise (Section 7). A high number of objective functions (Section 8) or a large problem scale (Section 9) increases the runtime requirement while also decreasing the expected quality of the solutions. As shown in the overview provided in Table 1, we discuss not only these interrelated issues in optimization, but also list their corresponding countermeasures (which is actually a m-to-n relation). If an optimization algorithm performs well in the presence of some of the problematic facets, this good performance has to be paid for with a loss of solution quality in a different situation – this fact has been formalized in the No Free Lunch Theorem, which we will discuss in Section 10. Finally, we conclude our review on the various issues with a summary in Section 11.

1.1 Basic Terminologies

Throughout this article, we will utilize terminologies commonly used in the EC community. Most of these terminologies are inspired from actual biological phenomena. Figure 1 shows the spaces involved in a typical evolutionary optimization scenario. The *candidate solutions* (or *phenotypes*) x of an optimization problem are elements of the *problem space* $\mathbb{X}$ (also called the *solution space*). Their utility is evaluated by $m \geq 1$ *objective functions* f, which embody the optimization criteria (usually subject to minimization). Together, these functions can be considered as one vector function $\vec{f} : \mathbb{X} \mapsto \mathbb{R}^m$.

The objective functions are the only direct source of information available to an EA. It uses this information to decide which candidate solutions are interesting and subsequently combines and/or modifies them in order to sample new points in the problem space. If these two processes can be conducted in a meaningful way, with a certain chance of finding better candidate solutions, the EA can progress towards an optimum – an issue which we discuss in Section 3.1 in more detail.

The search operations (such as the unary mutation or the binary recombination/crossover operation) utilized by the EA often do not work directly on the phenotypes. Instead, they are applied to the elements (the *genotypes*) of a search space $\mathbb{G}$ (the *genome*). The genotypes are encoded representations of the candidate solutions, which are mapped to the problem space by a *genotype-phenotype mapping* $\text{gpm} : \mathbb{G} \mapsto \mathbb{X}$. A traditional Genetic Algorithm (GA), for instance, may utilize a bit-string based encoding as the search space, which can be mapped to a real-valued problem space for function optimization [21–23]. In the common case that $\mathbb{G} = \mathbb{X}$, i.e., when the variables are processed in their "natural form" [24], the genotype-phenotype mapping is the identity mapping.

EAs manage a population, i.e., a set of individuals (genotype and the corresponding phenotype), which undergo evaluation, selection, and reproduction in each iteration (generation). Before selection, a single scalar *fitness* value is assigned to each individual. The fitness denotes the priority of an individual for being selected as the parent for offspring in the next generation, i.e., its chance of being chosen as the input to a search operation. This fitness, in general, is determined by a fitness assignment process that usually relies on the objective value(s) of the candidate solution stored in the individual record. It often relates these objective values to those of other candidate solutions in the population, e.g., by computing the individual's (Pareto) rank among them. The fitness may, however, also include additional information [19] such as diversity metrics (see, e.g., Section 2.2.5).

If only a single objective function is to be optimized (i.e., $m = 1$ and $\vec{f} = f$), it is sometimes referred to as the *fitness function* as well, so there exists some ambiguity in the terminology [2]. From the latter, the term "fitness landscape" is derived, which refers to the visualization of an objective function (and not of the results of a fitness assignment process).

An illustration of the spaces and sets involved in (evolutionary) optimization is given in Figures 1 and 2, where the candidate solutions are coordinate pairs decoded from bit strings (the genotypes) via the genotype-phenotype mapping. Each element of a genotype that can be modified by a search operation is called a *gene*. The term *building block* denotes groups of gene settings that together form an essential element of an individual.

1.2 Fitness Landscapes

As aforementioned, the most important information sources for an optimization algorithm are the $m \geq 1$ objective functions that rate the quality of possible solutions to an optimization problem. A function is "difficult" from a mathematical perspective in this context if it is not continuous, not differentiable, or if it has multiple maxima and minima. This understanding of difficulty comes very close to the intuitive curves in Figure 3 where we sketched a number of possible scenarios of the fitness landscape (objective function plots) that we are going to discuss in this article. The objective values in the figure are subject to minimization and the small bubbles represent candidate solutions under investigation. An arrow from one bubble to another means that the second individual is found by applying a search operation to the first one. As can be seen, there are different objective function shapes that can pose to be difficult for an optimization algorithm to proceed its search in this manner. EAs typically work on multiple solutions simultaneously and, as a result, the search space navigation can be difficult to visualize. These graphs thus provide a simplified visualization of the theories discussed rather than an accurate depiction of the EA search process. The structure of EAs enables them to often overcome some local optima, deception, ruggedness, and neutrality.

From these plots, it may also seem that the shape of the fitness landscape is defined by the objective function only. However, this is *not* true from the perspective of an EA. Here, the *representation*, i.e., the choice of search space, search operations, and the genotype-phenotype mapping, has a tremendous impact on the effective shape of the fitness landscape [25]. As outlined in the previous section, an EA conducts its search by applying the search operators to genotypes in a search space that are mapped to phenotypes in a problem space which, in turn, are evaluated by the objective functions, as illustrated in Figure 2. The concept of adjacency amongst candidate solutions from the viewpoint of an EA hence depends on the representation used and not on their proximity in the problem space (unless both spaces are the same, that is). In any case, many of the problematic issues which we will discuss in this article are closely related to the choice of representation, as can be seen directly in Table 1 and, for instance in Sections 3.2.4, 4.2.1, 5.1.2, and 6.2.1.

An important feature of the fitness landscape is that it may have different global and local structures. Figure 4 illustrates one objective function graph (in the top-left sub graph) from which regions are successively selected and "zoomed in". As can be seen, different sections of this function may exhibit different problematic features or issues. It is thus necessary to remember that the characteristic of an objective function may seem to be dynamic [26]

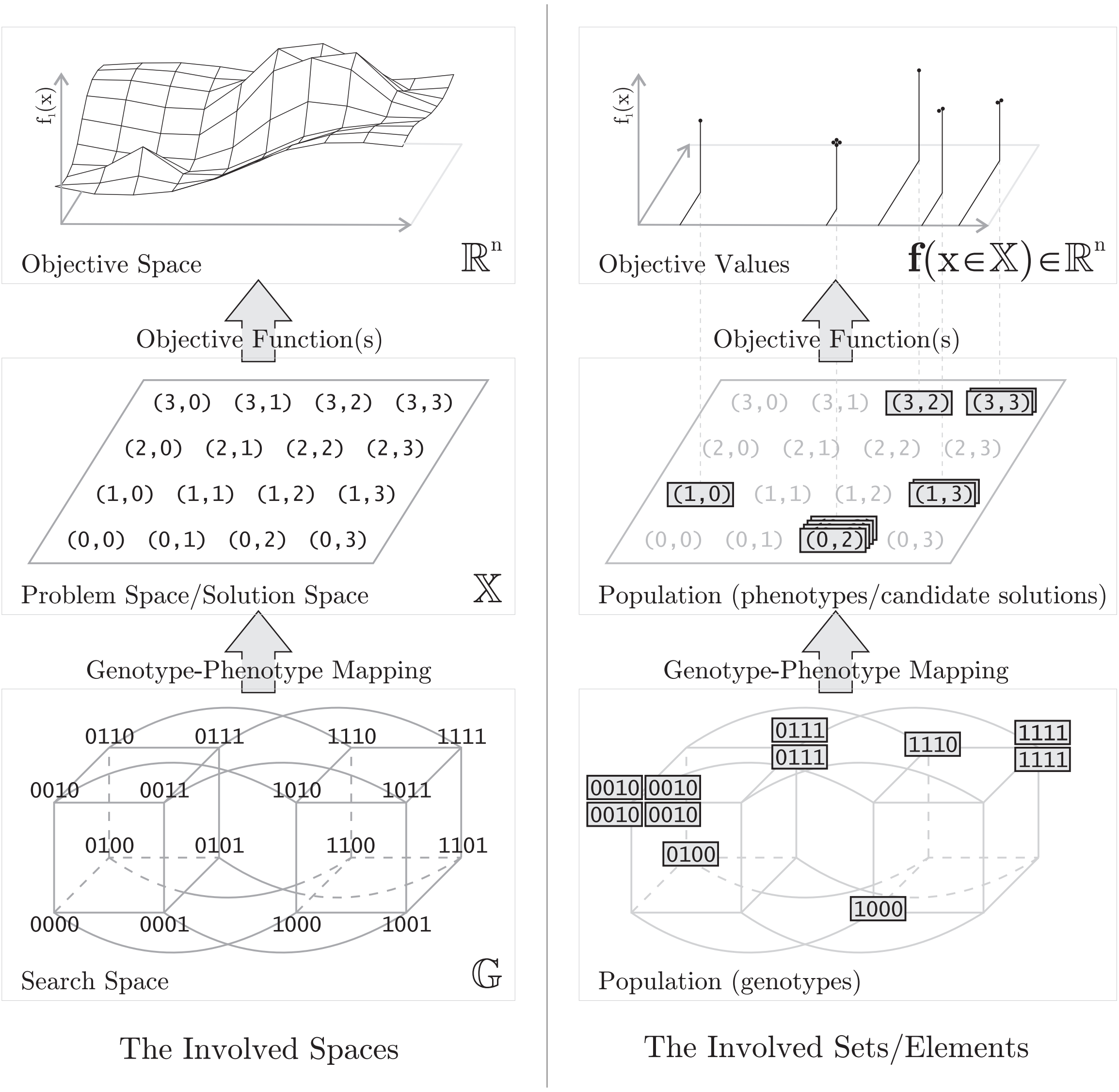


Figure 1: The involved spaces and sets in (evolutionary) optimization.

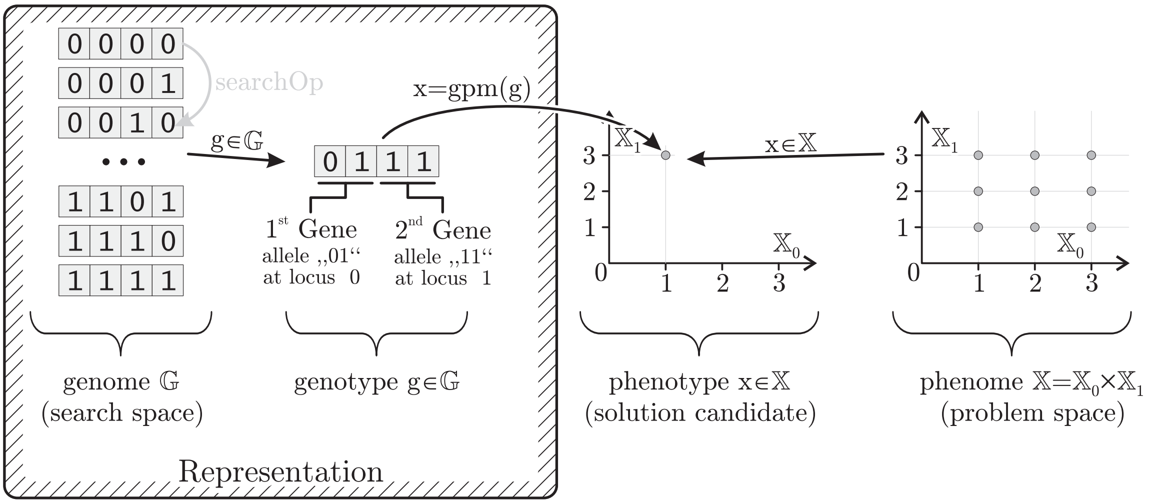


Figure 2: The relation between genotypes and phenotypes (candidate solutions) in EAs.

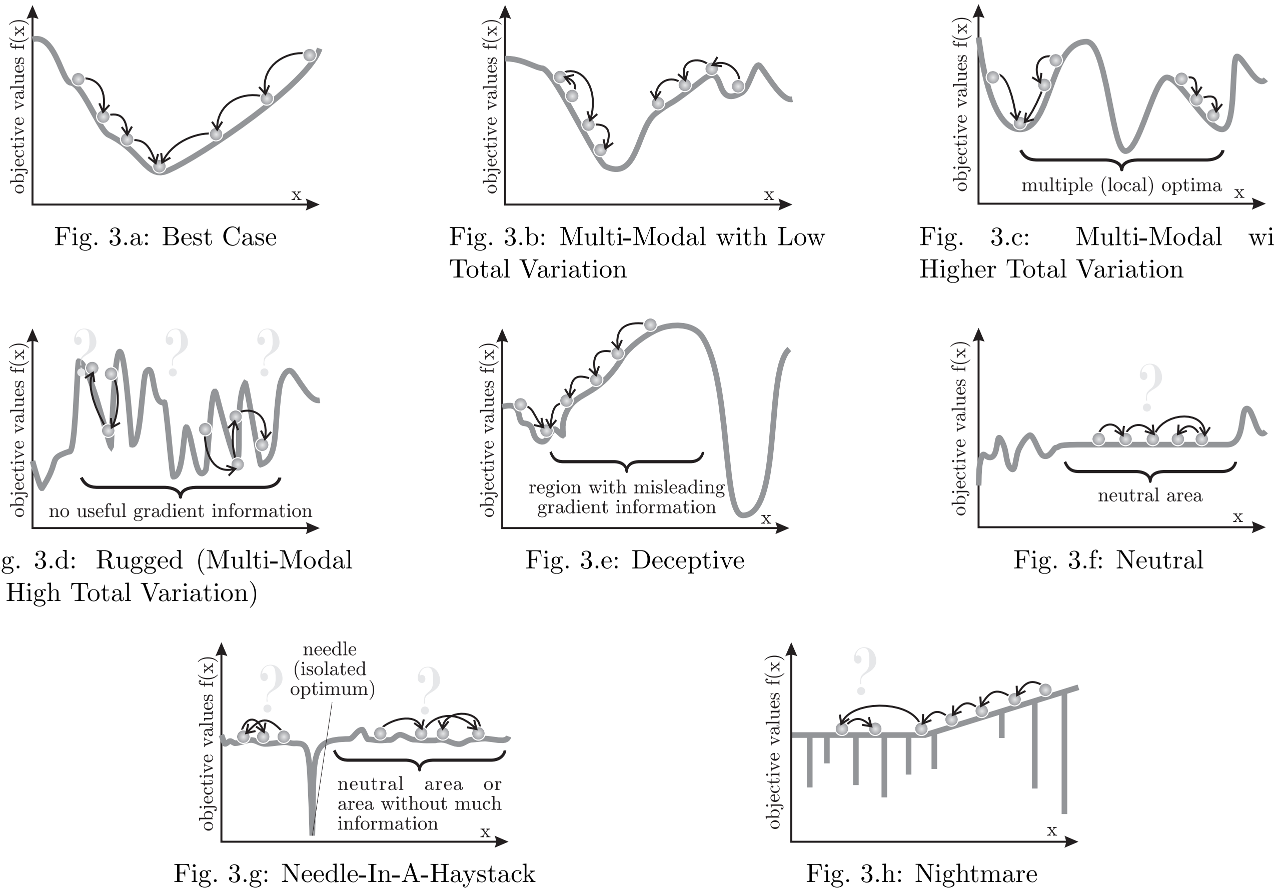


Fig. 3.a: Best Case

Fig. 3.b: Multi-Modal with Low Total Variation

Fig. 3.c: Multi-Modal with Higher Total Variation

Fig. 3.d: Rugged (Multi-Modal + High Total Variation)

Fig. 3.e: Deceptive

Fig. 3.f: Neutral

Fig. 3.g: Needle-In-A-Haystack

Fig. 3.h: Nightmare

Figure 3: Examples of different possible scenarios in the fitness landscape (under minimization).

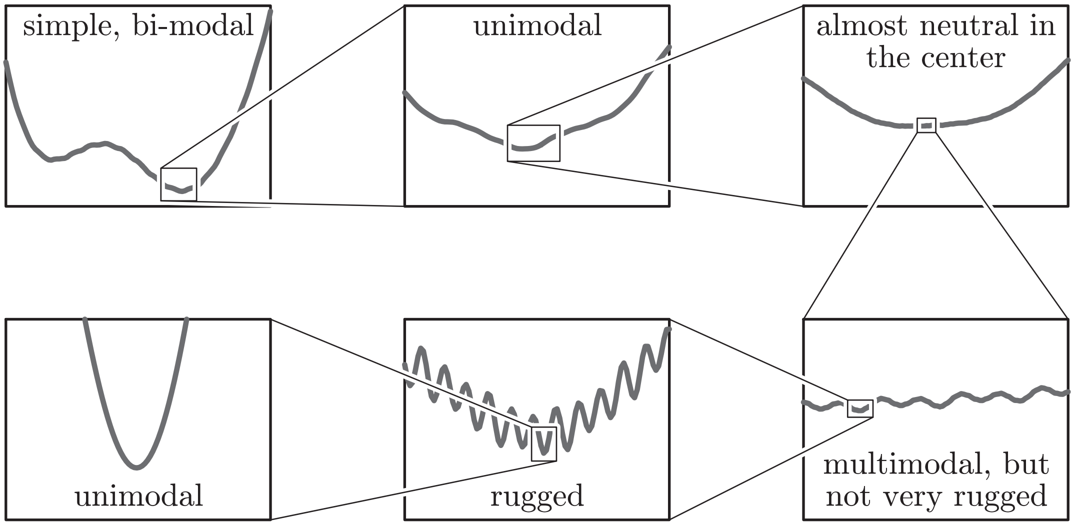


Figure 4: An artificial example of how landscape features may change depending on the selected region of the graph.

and change during the course of optimization when the global optimum is approached.

Before going into the details of difficult fitness landscape features, we would like to briefly review the term *difficult* itself from the perspectives of both traditional, deterministic, exact algorithms as well as EAs.

1.3 Problem Hardness

One of the basic goals of computer science is to find the *best* algorithm for solving a given class of problems. The performance measures used to rate an algorithm's efficiency are *(1)* the time it takes to produce the desired outcome and *(2)* the storage space it needs for internal data [27], i.e., its time and space complexity. Both of these can be described as functions of the input size of the algorithm for best, average, and worst-case input situations, which are usually simplified using the big-**O** family notations [28–30].

The computational complexity of a problem is bounded by the best algorithm known for that problem. It states how much resources are necessary for solving the given problem, or, from the opposite point of view, tells whether the given resources are sufficient for this purpose.

The set of all problem classes that can be solved on a computer[2] within polynomial time is called $\mathcal{P}$ [32]. These are problems which are said to be exactly solvable in a feasible way [33, 34]. The set of problem classes that allows solution verification in polynomial time is called $\mathcal{NP}$, which also comprises all the problems from $\mathcal{P}$ ($\mathcal{P} \subseteq \mathcal{NP}$). A problem A is *hard* for a complexity class if every other problem in this class can be *reduced* to it, i.e., if the other problems can be re-formulated so that they can also be solved with an algorithm for A. Exactly solving any $\mathcal{NP}$-hard problem is difficult as it may require super-polynomial, exponential time.

Solving such a problem to optimality is thus not always possible. When dealing with $\mathcal{NP}$-hard problems that have more than a certain number of variables, we may need to give up some solution quality in order to make the problem computationally tractable. EAs use

[2] or Deterministic Turing Machine [31]

some random process in their execution. These stochastic algorithms (usually) trade in solution correctness, i.e., the guarantee to find the global optimum, for a lower runtime. In other words, if we apply an EA, we would normally not expect to find the global optima but some reasonably good approximations within feasible time. The limits of this speed-up are discussed in Section 9.1.

While $\mathcal{NP}$-hard problems can be considered to be difficult for any exact method, the question about which problems are *GA*- or *EA-hard* arises. This question has been considered from several perspectives [35–38], and the most notable discussions can be found in [17, 39]: Problem instance classes for which the expected worst case first hitting time, i.e., the number of steps required to find a global optimum, of a particular EA has an exponential lower bound are "EA-hard" (for that EA). In [17], two such classes have been proposed for (1+1) EAs: *(1) wide-gap problems*, where there is a very low probability that the EA can escape a local optimum towards a region with higher utility, and *(2) long-path problems*, where advancement towards better objective values has a reasonably high probability, but the necessary number of such steps is very high. It should be noted that some instance classes of $\mathcal{NP}$-hard problems can be EA-easy [40].

Finding out how hard certain problems are for EAs is an active research area and much work has been devoted to finding the asymptotical complexity of these stochastic algorithms in different scenarios [15–18].

2 Convergence

An optimization algorithm has *converged (1)* if it cannot reach new candidate solutions anymore or *(2)* if it keeps on producing candidate solutions from a "small"[3] subset of the problem space [3]. Optimization processes will usually converge at some point in time. In the ideal case, convergence happens towards the global optimum. One of the problems in evolutionary optimization is that it is often not possible to determine whether the best solution currently known is situated on a local or global optimum and thus, if the convergence is acceptable. In other words, it is not clear whether the optimization process can be stopped, whether it should concentrate on refining the current optimum, or whether it should examine other parts of the search space instead. This, of course, can only become cumbersome if there are multiple (local) optima, i.e., the problem is *multi-modal* [41, 42], as depicted in Fig. 3.c. It is worthwhile to note that convergence often occurs much more quickly in the objective space than in the search and solution spaces.

2.1 The Issues

There are at least three basic problems related to the convergence of an optimization algorithm: *premature*, *non-uniform*, and *domino* convergence. The first one is considerably the most important in optimization, but the latter ones may cause a lot of inconveniences too.

2.1.1 Premature and Non-Uniform Convergence

The main goal in EC is to find solutions that are as close to the true global optimum as possible. An optimization process is considered to have *prematurely converged* to a local optimum if it is no longer able (or extremely unlikely) to explore other parts of the search space than the area currently being examined *and* there exists another region that contains a superior solution [43, 44]. In case that there is more than one global optimum, then the second goal is to discover as many of them as possible.

[3] according to a suitable metric like the number of modifications or mutations that need to be applied to a given solution in order to leave this subset

In single-objective optimization, all the global optima have the same objective values but reside on different peaks (or hyperplanes) of the objective function. The presence of multiple such optima is the focus of research on multi-modal optimization [45–47].

In multi-objective optimization, there are usually many global optima due to the trade-off of the objectives. Take the task of finding a good car, for example, where the criteria *speed* and *fuel consumption* would be traded-off. The optimization process should discover both, slower, environmentally friendly cars as well as fast cars that need more gasoline.

In some optimization problems, the number of (globally) optimal solutions is too large to provide all of them to the human operator. On these cases, the subset of delivered solutions should well represent the range of possible results, i.e., it should be a uniform sample of all possible optimal features. If only some of the optimal features are presented to the human operator, e.g., only the fast cars in the above example, the convergence is said to be *non-uniform* [48].

Figure 5 illustrates these issues on the examples of a single-objective (Fig. 5.a-c) and a bi-objective optimization task (Fig. 5.d-f); objectives are subject to minimization. Fig. 5.a shows the result of having a very good spread (or *diversity*) of solutions, but the points are far away from the optima. Fig. 5.d is a sketch of the same issue for a bi-objective problem: the discovered solutions are diverse, but distant from the true Pareto front of best trade-offs. Such results are not attractive because they do not provide optimal solutions and we would consider the convergence to be premature in this case. The second examples (Fig. 5.b and 5.e) contain solution sets that are very close to the true optima but cover them only partially, so the decision maker could lose important options. Finally, the optimization results depicted in Fig. 5.c and 5.f have the two desirable properties of good convergence (i.e., the solutions are very close to optimal) and spread (i.e., the whole trade-off curve between the two objectives is covered).

2.1.2 Domino Convergence

The phenomenon of *domino convergence* [49, 50] occurs when the candidate solutions have features contributing to significantly different degrees to the total fitness. If these features are encoded separately, they are likely to be treated with different priorities. If, for example, optimization takes place over $\mathbb{R}^\ell$ and the first element of a solution vector is much more important (from the perspective of the objective function) than the second one, its priority during the optimization process will be much higher too.

Although this seems to be legit, it can prevent us from finding the global optimum: gene values with strong positive influence on the objective values, for instance, will quickly be adopted by the optimization process (i.e., "converge"). During this time, the values of the genes with smaller contribution are ignored. Their state may remain rather random and *hitchhike* through the generations in genotypes with good configurations of the more salient genes [51]. They do not receive evolutionary pressure until the optimal configurations of these genes have been accumulated. This sequential convergence phenomenon is called *domino convergence* due to its resemblance to a row of falling domino stones [50].

In the worst case, the contributions of the less influential genes may look almost like noise and they are not optimized at all. This leads to premature convergence, since the global optimum which would involve optimal configurations of all genes will not be discovered. Here, restarting the optimization process will not help because it will turn out the same way with very high probability.

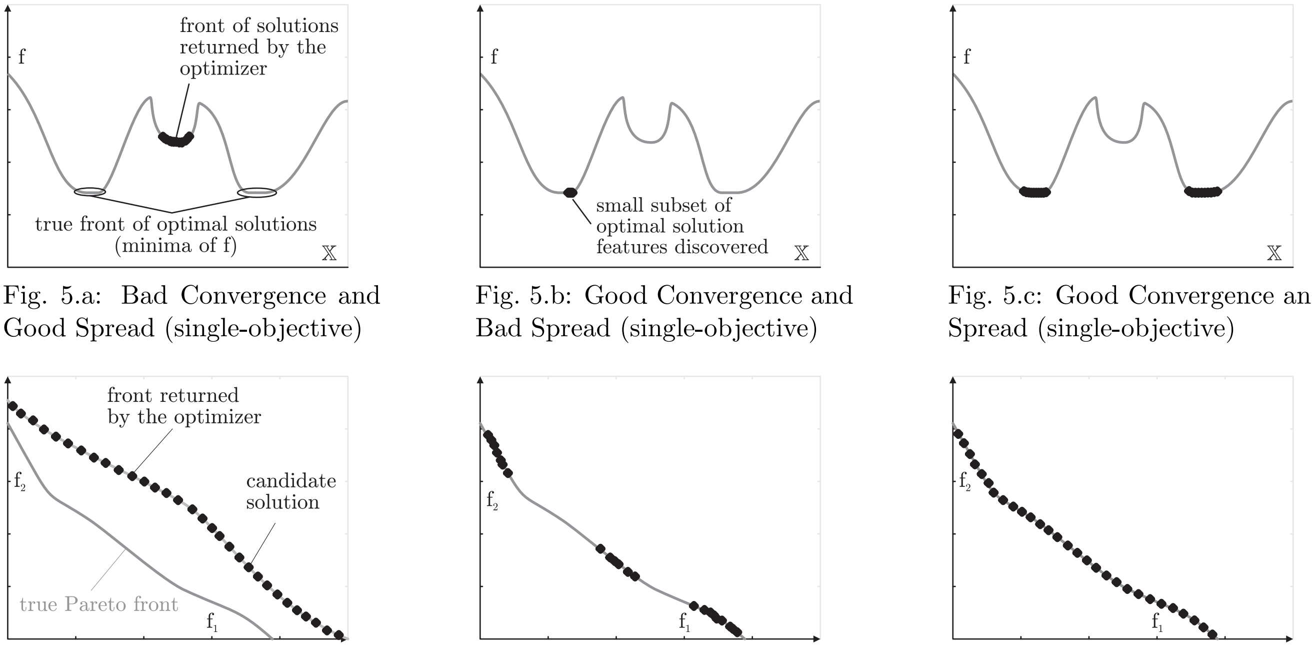


Fig. 5.a: Bad Convergence and Good Spread (single-objective)

Fig. 5.b: Good Convergence and Bad Spread (single-objective)

Fig. 5.c: Good Convergence and Spread (single-objective)

Fig. 5.d: Bad Convergence and Good Spread (bi-objective)

Fig. 5.e: Good Convergence and Bad Spread (bi-objective)

Fig. 5.f: Good Convergence and Spread (bi-objective)

Figure 5: Optimal solution approximation sets.

2.1.3 Diversity, Exploration, and Exploitation

In biology, diversity is referred to as *the variety and abundance of organisms at a given place and time* [52, 53]. Genetic diversity is the fuel of evolution and essential for a species' robustness against and adaptivity to environmental changes. In EAs, maintaining a diverse population is very important as well. Losing diversity means approaching a state where all the candidate solutions under investigation become similar to each other. Consequently, no new areas in the search space will be explored and the optimization process will not make any further progress.

The process of finding points in new areas of the search space that are rather distant from the currently investigated candidate solutions is called *exploration* [54]. Exploration increases diversity but often leads to the creation of solutions inferior to those that have already been investigated. However, like in biology, there is a small chance that new genetic material can lead to the discovery of superior traits.

On the other hand, *exploitation* is the process of improving and combining the traits of the (best) currently known solutions. Exploitation-based search operations often perform small changes in individuals, producing new, very similar candidate solutions. This would give rise to some steady improvement in fitness for a period of time, but it also reduces diversity in the population since offspring and parents become more and more similar to each other. Another problem with exploitation is that possibly existing better solutions which may be located in distant areas of the problem space will not be discovered.

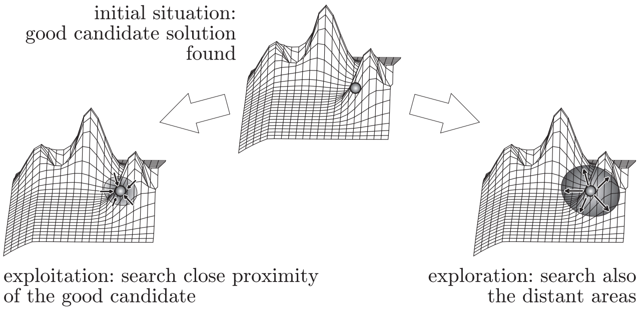


Figure 6: Exploration versus Exploitation

Exploration versus exploitation [55–57] is therefore the dilemma of deciding which of the two principles to apply and to which degree at a certain stage of optimization. It is sketched in Figure 6 and can be observed in many areas of optimization.[4]Optimization algorithms that favor exploitation over exploration have higher convergence speed but run the risk of not finding the optimal solution and may get stuck at a local optimum. Then again, algorithms that perform excessive exploration may never improve their candidate solutions well enough to find the global optimum or it may take them very long to discover it.

Almost all components of optimization strategies can either be used for increasing exploitation or in favor of exploration. Exploitation can be achieved by building unary search operations (e.g., mutation operators) that improve an existing solution in small steps. However, mutation in an EA can also be implemented in a way that introduces much randomness into the individuals, effectively turning it into an exploration operator. Selection operations choose a set of the most promising candidate solutions that will be investigated in the next iteration of the algorithm. They can either return a small group of best individuals (exploitation) or a wide range of existing candidate solutions (exploration).

A good example for the exploration vs. exploitation dilemma is the Simulated Annealing algorithm [60]. It is often modified to a faster form called *simulated quenching*, which focuses on exploitation but loses the guaranteed convergence to the optimum [61]. Another good example is given in [62, 63], where it is shown that for some problems, the selection pressure and mutation rate of an EA must be balanced extremely well in order to achieve a polynomial expected runtime. Too much exploitation or exploration may both lead to an exponential expected first hitting time.

2.2 Countermeasures

There is no general approach to prevent unsatisfying convergence as this phenomenon may have a variety of different causes. The probability of an optimization process getting caught in a local optimum depends on the characteristics of the problem at hand and the parameter settings as well as on features of the optimization algorithms applied [48, 64, 65].

2.2.1 Balanced Exploration and Exploitation

Generally, optimization algorithms should employ at least one search operation of explorative character and at least one that is able to exploit good solutions further. There exists a vast body of research on the trade-off between exploration and exploitation that optimization algorithms have to face [54, 66, 67], ranging from targeted initialization of the population [68], mining data from the optimization process [69], to devising specialized population structures [70] and specialized search operators [71].

2.2.2 Search Operator Design

A very basic measure to decrease the probability of premature convergence is to make

[4] more or less synonymous to exploitation and exploration are the terms *intensification* and *diversification* [58, 59]

sure that the search operations are *complete*, i.e., to make sure that they can (theoretically at least) reach every point in the search space from every other point. Then, it is possible to escape arbitrary local optima with non-zero probability.

A good example for this is the modification to Evolutionary Programming (EP) introduced in [72]: By replacing the usually applied normally distributed mutations with Lévy distributed ones, the probability to reach distant points in a real-coded search space within a single mutation step is increased and better results could be obtained. In [73], the large impact of search operator design on the solution quality for a combinatorial problem is confirmed.

2.2.3 Restarting

A very crude yet sometimes effective measure is to restart the optimization process at randomly or strategically chosen points in time. One example for this is the *Greedy Randomized Adaptive Search Procedure* (GRASP) [74, 75], which continuously restarts the process of creating an initial solution and refines it with local search. Still, this approach is likely to fail in domino convergence situations.

2.2.4 Low Selection Pressure and/or Larger Population Size

Generally, the higher the chance that candidate solutions with bad fitness are investigated instead of being discarded in favor of seemingly better ones, the lower the chance of getting stuck at a local optimum. This is the exact idea which distinguishes Simulated Annealing from Hill Climbing. It is known that Simulated Annealing can find the global optimum, whereas simple Hill Climbers are likely to prematurely converge since they always proceed with the best candidate solution discovered so far.

In an EA, too, using low selection pressure decreases the chance of premature convergence and can lead to a better approximation of the true global optima. However, such an approach also decreases the speed with which good solutions are exploited and thus, increases the runtime. Also, too low of a selection pressure may cause genetic drift, which we will put into the context of neutrality and evolvability in Section 5.1.1.

Increasing the population size may be useful as well, since larger populations can maintain more individuals and hence, cover many different solutions. This coverage can lead to a lower selection pressure. However, the idea that larger populations will lead to better optimization results does not always hold [76, 77]. For these reasons, both population-sizing [77, 78] and selection [16] are highly-active research areas in the EC community.

2.2.5 Sharing, Niching, and Clearing

As opposed to increasing the population size, it is also possible to "gain more" from the smaller populations. In order to extend the duration of the evolution in EAs, many methods have been devised for steering the search away from areas which have already been frequently sampled. In steady-state EAs it is common to remove duplicate genotypes from the population [79].

More generally, the exploration capabilities of an optimizer can be improved by integrating density metrics into the fitness assignment process. The most popular of such approaches are sharing and niching [45, 56, 80–84]. The Strength Pareto-type Algorithms, which are widely accepted to be highly efficient, use another idea: they adapt the number of individuals a candidate solution *dominates* as the density measure [85, 86]. In the *simple convergence prevention* method [3, 87, 88], candidate solutions with the same objective values are deleted based on a given probability. In the *clearing* approach [89, 90], all individuals are grouped according to their distance in the phenotypic or genotypic space and all but a certain number of individuals from each group receive the worst possible fitness. The efficiency of all these diversity preservation methods strongly depends on the situation – a method suitable for one scenario may cause problems in another [91].

2.2.6 Clustering of Candidate Solutions

A more explicit method to prevent premature convergence is to cluster the search space or population of an EA. This allows the optimization method to track multiple different basins of attraction at the same time and increases the chance of finding the global optimum in one of them. Particularly in the context of Estimation of Distribution Algorithms (EDAs), various such methods have been proposed [92–103].

2.2.7 Self-Adaptation

Another approach against premature convergence is to introduce the capability of self-adaptation, allowing the optimization algorithm to change its strategies or to modify its parameters depending on its current state. Such behaviors, however, are often implemented not in order to prevent premature convergence but to speed up the optimization process (which may lead to premature convergence to local optima) [104, 105].

2.2.8 Multi-Objectivization

Recently, the idea of using helper objectives [106] has emerged. Here, a single-objective problem is transformed into a multi-objective one by adding new objective functions [107–113]. In some cases, such changes can speed up the optimization process [114, 115]. The new objectives are often derived from the main objective by decomposition [115] or from certain characteristics of the problem [111]. They are then optimized together with the original objective function with some multi-objective techniques.

3 Ruggedness

Optimization algorithms generally depend on some form of trends[5] in the fitness landscape. Ideally, the objective functions would be continuous and exhibit low total variation[6] (as sketched in Fig. 3.b), so that the optimizer can track the trend easily. If an objective function is unsteady or goes up and down frequently, it becomes more complicated to find the right directions to proceed during the optimization process (see Figure 7 and Fig. 3.d). The more rugged the function gets, the harder it is to optimize it. In short, one could say ruggedness is multi-modality (see Fig. 3.c) plus steep ascends and descends in the fitness landscape.

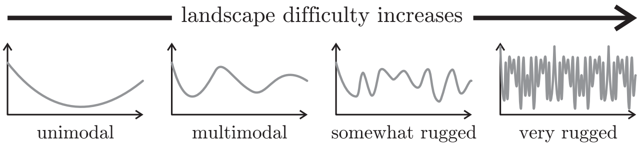


Figure 7: The landscape difficulty increases with increasing ruggedness.

3.1 The Issue: Weak Causality

During an optimization process, new points in the search space are created by the search operations. Generally, we can assume that the inputs of the search operations correspond to points that have previously been selected. Usually, the better or the more promising an individual is, the higher are its chances

[5] using the word “gradient” here would be too restrictive and mathematical

[6] http://en.wikipedia.org/wiki/Total_variation [accessed 2011-11-25]

of being selected for further investigation. Reversing this statement suggests that individuals being passed to the search operations are likely to have good fitness. Since the fitness of a candidate solution depends on its features, it can be assumed that the features of these individuals are promising, too. It should thus be possible for the optimizer to introduce small changes to these features (by modifying the genes encoding them slightly) in order to find out whether they can be improved any further. Normally, such *exploitive* modifications should also lead to small changes in the objective values and hence, in the fitness of the candidate solution.

Strong causality (locality) means that small changes in the features of an object also lead to small changes in its behavior [116–118]. In fitness landscapes with weak (low) causality, small changes in the candidate solutions often lead to large changes in the objective values. It then becomes harder to decide which region of the problem space to explore and the optimizer cannot find reliable trend information to follow. The lower the causality of an optimization problem, the more rugged its fitness landscape is, which leads to degeneration of the performance of the optimizer [119]. This does not necessarily mean that it is impossible to find good solutions, but it may take longer time to do so.

3.2 Countermeasures

Ruggedness in the fitness landscape is hard to mitigate. In population-based approaches, using large population sizes and applying methods to increase diversity can reduce the influence of ruggedness, but only up to a certain degree.

3.2.1 Hybridization with Local Search

Often, EAs are combined with a local search technique applied to each individual in the population before presenting it to the evolutionary process. Two such common approaches are Lamarckian evolution [120, 121] (performing a local search on the genotype level) and the Baldwin effect [121–124] (local search on the phenotype level). Memetic Algorithms [125–134] and other hybrid approaches [24, 135–141] also fall into this category. Since the EA only receives individuals residing in local optima resulting from the local search procedure(s), the fitness landscape may seem to be less rugged from its perspective [142, 143]. However, local search can also lead to much higher selection pressure and thus swing the pendulum to the problem of premature convergence [4].

3.2.2 Landscape Approximation

In order to smoothen out a rugged landscape, it can be approximated by parameterizing a function based on the knowledge gathered from previously sampled candidate solutions. The optimization process can then be performed on this smooth approximation, which, in turn, is updated in each step. The goal here is not to find a function that perfectly represents the fitness landscape, but to work on a much smoother function without changing the location of the global optimum. In [143], for example, a k-dimensional quadratic polynomial is used to approximate the fitness function. The second advantage of this idea is that a new candidate solution can be created by directly solving the approximation function analytically.

3.2.3 Two-Staged Optimization

Another approach is to apply a two-staged optimization process [144] where two different algorithms are applied sequentially. Here, the first optimization method should be an algorithm with strong global optimization abilities, which discovers the most promising area in the search space and is not easily distracted from rugged objectives (e.g., an EDA). Then, an algorithm that is quick to exploit and follow the trend in a landscape, such as Differential Evolution (DE), is applied to the subspace discovered by the first algorithm.

3.2.4 Better Operator and Search Space Design

Weak causality is often caused, to some extent, by bad design of the solution representation and search operations. We pointed out that exploration operations are important for minimizing the risk of premature convergence. Exploitation operators are equally important for refining the solution quality. In order to apply optimization algorithms in an efficient manner, it is necessary to find representations that allow for iterative modifications with bounded influence on the objective values [87, 88, 145, 146], i.e., exploitation. This can eventually lead to better candidate solutions. Fortunately, many problems where their formulation is inspired by a real-world problem share the feature that improved solutions can often be built from other good solutions, i.e., often exhibit strong causality. A comprehensive collection of examples for representations that exhibit this property in real-world application domains can be found in [5].

4 Deceptiveness

Especially annoying fitness landscapes show *deceptiveness* (or deceptivity). The gradient of deceptive objective functions leads the optimization process away from the optima, as illustrated in Fig. 3.e as well as Figure 8. The term deceptiveness [147] is mainly used for the GA in the context of the Schema Theorem [3, 148, 149]. Schemas describe certain areas (hyperplanes) in the search space. If an optimization algorithm has discovered an area with better average fitness compared to other regions, it will focus on exploring this region based on the assumption that highly fit areas are likely to contain the true optimum. Objective functions where this is not the case are considered to be deceptive [24, 147, 150]. It is interesting that some problems with the highest level of deceptiveness appear to be easy for GAs [24], whereas an increasing amount of deceptiveness generally leads to a steep increase in problem hardness [151].

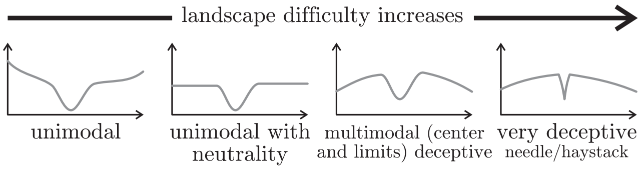

landscape difficulty increases
unimodal
unimodal with neutrality
multimodal (center and limits) deceptive
very deceptive needle/haystack

Figure 8: Increasingly difficult landscapes caused by deceptivity.

4.1 The Issue

An objective function is *deceptive* if a greedy local search algorithm would be steered in a direction leading away from all global optima in large parts of the search space. The basic problem caused by deceptiveness is that the information accumulated by an optimizer actually guides it away from the optimum. Search algorithms that strictly follow a path towards improving fitness will not be able to discover the global optimum in this case. In other words, they may perform worse than non-repeating random sampling, a random walk, or an exhaustive enumeration method in terms of the first hitting time of the global optimum. These most primitive search methods sample new candidate solutions without taking into account the utility of the already investigated solutions and hence are not vulnerable to deceptiveness.

4.2 Countermeasures

Solving tasks with deceptive objective functions perfectly involves sampling many individuals with very bad features and fitness. This contradicts the basic ideas of metaheuristics and thus, there are no really efficient countermeasures against high degrees of objective function deceptivity. Using large population sizes, maintaining high diversity (see, e.g., Sections 2.2.5 and 2.2.6), and utilizing linkage learning (see Section 6.2.3) provide at least a small chance of finding good solutions.

4.2.1 Representation Design

Like weak causality, deceptiveness can also be caused by the design of the representation. Utilizing a more suitable search space, search operations, and genotype-phenotype mapping may make an optimization problem much less deceptive. Notice that the representation is a part of the optimization algorithm which produces the inputs of the objective function (see Figure 1). Changing it can change the behavior of the objective function from the perspective of the optimization process significantly. Combining different representations in an EA may lead to better results as shown in [152]. This can be a feasible approach if the nature of the problem is too complex to manually design a non-deceptive representation.

4.2.2 Niching and Memory

Applying the diversity increasing methods mentioned in Section 2.2.5 (such as niching and the *simple convergence prevention* method) can delay the convergence of the optimization process and thus, increase the chance to escape from deceptive local optima. Recent studies of Particle Swarm Optimization (PSO) [153] show that the local memory property of the simulated particles can lead to some niching behavior, which is especially suitable for this purpose as well. Here, the *lbest* PSO with ring topology discussed in [153] is noteworthy.

4.2.3 Preventing Convergence

Another approach to counteract deceptiveness is to stop the optimization algorithm from converging altogether. If the population of an EA is prevented from collapsing to a certain area of the search space and is always kept “moving”, deceptive basins of attraction will be left eventually.

The Fitness Uniform Selection Scheme [154–157] takes the idea a step further. Instead of selecting the most promising candidate solutions, a diverse population with individuals from all fitness levels is maintained in order to avoid getting stuck at a local optimum. To achieve this, in each generation the best and worst individuals (with the smallest and largest fitness, say f_s and f_l) in the population are first determined. For each slot in the new population, a random value r uniformly distributed between f_s and f_l is drawn and the individual with the fitness closest to r will be selected. The selected candidate solutions will be diverse in terms of fitness and the population basically maintains a path of individuals out of the current local optimum.

If the optimization problem lacks causality and the fitness landscape is very rugged, however, this method may fail. If structurally similar points within a small subset of the search space may possess very different fitness, the search may get trapped within that subset.

4.2.4 Novelty Search

In Novelty Search [158–160], the objective function f is completely abandoned. The reason is that, on one hand, in the case of deceptivity f may be misleading and guide the search away from the global optima. On the other hand, it is also not clear whether f would reward stepping stones, i.e., the intermediate solutions between the initially chosen starting points and the global optimum. In many Genetic Programming (GP) applications [145, 161], for example, the intermediate steps obtained by modifying a bad program iteratively towards a perfect solution rarely form a sequence of improving fitness and even needle-in-a-haystack situations (see Section 5.1.3) are common.

Novelty Search thus does not employ a traditional fitness measure since it may not help the optimizer to discover and combine building blocks anyway. Instead, an archive of past candidate solutions is kept and updated and selection will choose the individuals that differ the most from the archived ones. As more and more candidate solutions with different behaviors are discovered, chances are that one amongst them is an acceptable solution. This method led to good results in the evolution of virtual creatures [160], walking behaviors [159], and navigation control [158, 159].

5 Neutrality

The outcome of the application of a search operation to an element of the search space is *neutral* if it yields no change in the objective values [162, 163]. It is challenging for optimization algorithms if the best candidate solution currently known is situated on a plane of the fitness landscape, i.e., all adjacent candidate solutions have the same objective values. As illustrated in Fig. 3.f and Figure 9, an optimizer cannot find any gradient information in this case and thus there is no direction as to which way to proceed in a systematic manner. From its point of view, each search operation will yield identical individuals. Furthermore, optimization algorithms usually maintain a list of the best individuals found, which will eventually overflow and require pruning.

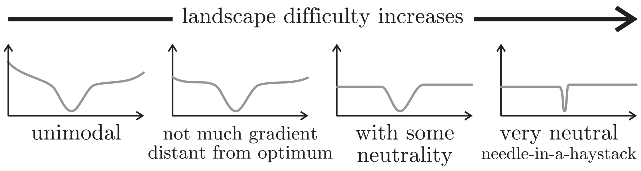

landscape difficulty increases
unimodal
not much gradient distant from optimum
with some neutrality
very neutral needle-in-a-haystack

Figure 9: Landscape difficulty caused by neutrality.

5.1 The Issues

5.1.1 Evolvability

Another metaphor in EC that has been borrowed from biological systems is *evolvability* [164, 165]. In biology, the meaning of this word is twofold [166, 167]: *1)* a biological system is evolvable if it is able to generate heritable, selectable phenotypic variations [168]; and *2)* a system is evolvable if it can acquire new characteristics via genetic changes that help the organism(s) to survive and to reproduce. In the optimization domain, the evolvability of an optimization process defines how likely the search operations will lead to candidate solutions with new (and eventually, better) objective values.[7] In Sections 4.2.3, 4.2.4, and (part of) 2.2.5, we already argued that preventing an optimization process from converging, i.e., keeping it in an evolvable state, may enable it to discover better results.

The link between evolvability and neutrality has been discussed by many researchers [167, 172]. The evolvability of neutral parts of a fitness landscape depends on the optimization algorithm used. For example, the evolvability of Hill Climbing-like approaches can be especially low, since the search operations cannot directly provide improvements or even changes in fitness. This could then degenerate the optimization process to a random walk, as illustrated in Fig. 3.f. Using the ND fitness landscapes, i.e., landscapes with a well-defined degree of neutrality, it has been shown that neutrality may "destroy" useful information such as correlation [173].

Researchers in molecular evolution, on the other hand, found indications that the majority of mutations in biology have no selective influence [174–176], and that the transformation from genotypes to phenotypes is a many-to-one mapping. Neutrality in natural genomes is often considered as beneficial if it concerns only a subset of the properties peculiar to the offspring while allowing meaningful modifications of the others [167, 177].

The theory of *punctuated equilibria* [178–180] states that species experience long periods of evolutionary inactivity, which are interrupted by sudden, localized, and rapid phenotypic evolutions [181]. It is assumed that the populations explore networks of neutral genetic changes during the time of stasis until, suddenly, a relevant change in a genotype leads to a better adapted phenotype [182] and reproduces quickly. Similar phenomena can be observed and have been utilized in EAs [183, 184].

Another example for neutrality in biology is *degeneracy*: the ability of elements that are structurally different to perform the same function or yield the same output [185] while also having additional, unique features. Similarly, degeneracy of the properties of candidate solutions introduced by the chosen solution representation in an optimization process can improve its robustness and ability to adapt [186].

The key to differentiating between "good" and "bad" neutrality is its degree in relation to the number of possible solutions maintained by an optimization algorithm. The illustrative example in Figure 10 shows that a certain amount of neutral reproduction can foster the progress of optimization. In Fig. 10.a, a scenario of premature convergence is depicted. Fig. 10.b shows that a little shot of neutrality could form a bridge to the global optimum. The optimizer now has a chance to escape the smaller peak if it is able to find and follow that bridge, i.e., the evolvability of the system has increased. If this bridge gets wider, as sketched in Fig. 10.c, the chance of finding the global optimum increases as well. Then again, if the bridge gets too wide (see Fig. 10.d), the optimization process may end up in a scenario like Fig. 3.f where it cannot find any direction.

[7] The direct *probability of success* [116, 169], i.e., the chance of search operators producing offspring fitter than their parents, is also sometimes referred to as *evolvability* in the context of EAs [170, 171].

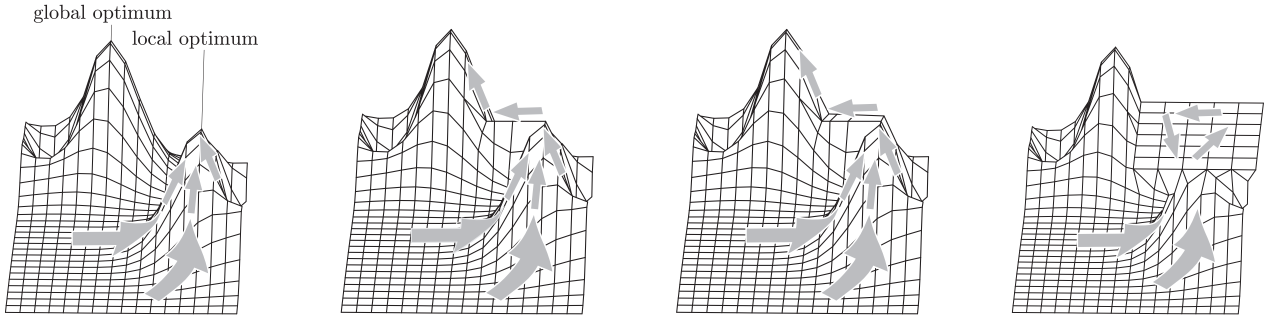


Fig. 10.a: Premature Convergence

Fig. 10.b: Small Neutral Bridge

Fig. 10.c: Wide Neutral Bridge

Fig. 10.d: Neutral Bridge too Wide

Figure 10: Possible positive and negative influence of neutrality (inspired by [187]).

Drift, a term stemming from the area of population genetics, describes the loss of population diversity resulting from the stochastic nature of selection in a finite population (in both nature and EAs) [174, 188]. In neutral parts of the fitness landscape or under low selection pressure, this effect is very likely. A reduction of diversity in the population generally is a negative effect (see Section 2.1.3).

Unlike ruggedness, which is always bad for the performance of optimization algorithms, neutrality has aspects that may further as well as hinder the process of finding good solutions. Generally, we can state that very high degrees of neutrality degenerate optimization processes to random walks. On the other hand, some forms of neutral pathways can improve evolvability and hence increase the chance of finding good solutions.

5.1.2 Redundancy

Redundancy in the context of EAs is a feature of the genotype-phenotype mapping and it means that multiple genotypes are mapped to the same phenotype, i.e., the genotype-phenotype mapping is not injective. The role of redundancy in the genome is as controversial as that of neutrality [189]. There exist many accounts of its positive influence on the optimization process. In [190, 191], redundant genotype-phenotype mappings are developed using voting (via uniform redundancy as well as a non-trivial approach), Turing machine-like binary instructions, cellular automata, and random Boolean networks (RBNs) [192]. Except for the trivial voting mechanism based on uniform redundancy, the mappings could induce neutral pathways that were beneficial for exploring the problem space. The RBN approach in particular provided very good results [190, 191].

Redundancy can have a strong impact on the explorability of the problem space. When utilizing a one-to-one mapping, the translation of a slightly modified genotype will always result in a different phenotype. If there exists a many-to-one mapping between genotypes and phenotypes, the search operations can create offspring genotypes that are different from their parent(s) but still translate to the same phenotype. The optimizer may now walk along a path through this "neutral network". If many genotypes along this path can be modified to different offspring, many new candidate solutions can be reached [190].

Further accounts of positive effects of neutrality in the genotype-phenotype mapping can be found in [193, 194]. In the Cartesian GP method, neutrality is explicitly introduced to increase evolvability [195, 196]. Yet, simple uniform redundancy is not necessarily beneficial for the optimization process and may even slow it down [25, 191]. If the population of individuals under investigation contains many *isomorphic* genotypes, i.e., genotypes that encode the same phenotype, a slow-down may also occur [79]. If this isomorphism can be identified and removed, a significant speed-up may be gained [79].

5.1.3 Needle-in-a-Haystack Problems

Besides fully deceptive problems, one of the worst cases found in fitness landscapes is the *needle-in-a-haystack* problem [37] (see Figure 9 and 3.g), where the optimum occurs as an isolated spike in a plane [146, 197]. In other words, this is the combination of small instances of extreme ruggedness with a general lack of information in the fitness landscape. Such problems are extremely hard to solve and the optimization process often will converge prematurely or take very long to find the global optimum. An example of this kind of fitness landscapes is the all-or-nothing property often inherent to GP [145, 146, 198, 199].

5.2 Countermeasures

Extreme cases of neutrality, especially the needle-in-a-haystack-type fitness landscapes, are hard to combat. Hybridization of an EA with local search is sometimes recommended in such situations [123, 124]. Multi-objectivization (see Section 8.2.2) and increasing the population size can possibly reduce the impact of neutrality too.

5.2.1 Selection Pressure

Higher selection pressure may be useful if the neutral regions in the fitness landscape still exhibit marginally different objective values that could be exploited to find a way out. It should be noted that fitness proportionate selection methods (e.g., “Roulette-Wheel Selection”) may perform very badly in such a case, since they will assign the essentially same reproduction probability to all individuals. Other methods such as Tournament Selection, which only consider the *less-then* relation instead of absolute fitness values and proportions, will be not affected.

In the case where all objective values in the neutral regions are identical, a strong emphasis on diversity, possibly achieved by sharing and niching in the problem or search space (see Section 2.2.5), may drive the search out of the neutral region faster.

5.2.2 Representation

Uniform redundancy in the genome should be avoided as it causes adverse forms of neutrality. In [24, 200], it is stated that the representation of phenotypic traits in the search space should be as short as possible. The length of different genes and the numbers of their alleles should be as small as possible. However, as we discussed earlier, non-trivial representations with a well-adjusted degree of redundancy may exhibit a higher evolvability and thus lead to a more robust and steadily improving optimization process [196].

5.2.3 Memory

In Tabu Search, recently performed search steps are memorized and not performed again. This allows the algorithm to escape small neutral areas. Similar techniques could be applied in EAs as well.

6 Epistasis, Pleiotropy, and Separability

In biology, *epistasis* is defined as a form of interaction between different genes [201]. The term was coined by Bateson [202] and originally meant that one gene suppresses the phenotypical expression of another gene. In the context of statistical genetics, epistasis was initially called "epistacy" by Fisher [203]. According to [204], the interaction between genes is epistatic if the effect of altering one gene on the fitness depends on the allelic state of other genes. In (evolutionary) optimization, epistasis is the non-linear interaction of two or more genes of the genotypes as expressed in objective function values after the genotype-phenotype mapping. Two genes interact epistatically if the contribution of one of these genes to the objective value depends on the value of the other gene [3, 205–207]. Epistasis can also be considered as the higher-order or non-main effects in a model predicting fitness values based on the interactions of the genes from the viewpoint of Design of Experiments [113, 208].

On one hand, we speak of minimal epistasis when every gene is independent of every other gene. Then, the optimization process equals finding the best value for each gene and can most efficiently be carried out by a simple greedy search iteratively applied to each gene while keeping the others constant [205]. On the other hand, a problem is maximally epistatic when no proper subset of genes is independent of any other gene [207, 209]. The effects of epistasis are closely related to another biological phenomenon: *Pleiotropy*, which denotes that a single gene is responsible for multiple phenotypical traits [164, 210, 211].

Like epistasis, pleiotropy can sometimes lead to unexpected improvements but often is harmful for an evolutionary system [171]. Both phenomena may easily intertwine. If one gene epistatically influences, for instance, two others that are responsible for distinct phenotypical traits, it has both epistatic and pleiotropic effects. We will therefore consider pleiotropy and epistasis together, and when discussing the effects of the latter, we also implicitly refer to the former.

In Figure 11, we illustrate a fictional dinosaur along with a snippet of its fictional genome consisting of four genes. Gene 1 influences the color of the creature and is neither pleiotropic nor has any epistatic relations. Gene 2, however, exhibits pleiotropy since it determines the length of the hind legs and forelegs. At the same time, it is epistatically connected with gene 3, which also influences the length of the forelegs – maybe preventing them from looking exactly like the hind legs. The fourth gene is again pleiotropic by determining the shape of the bone armors on the top of the dinosaur's skull and on its snout.

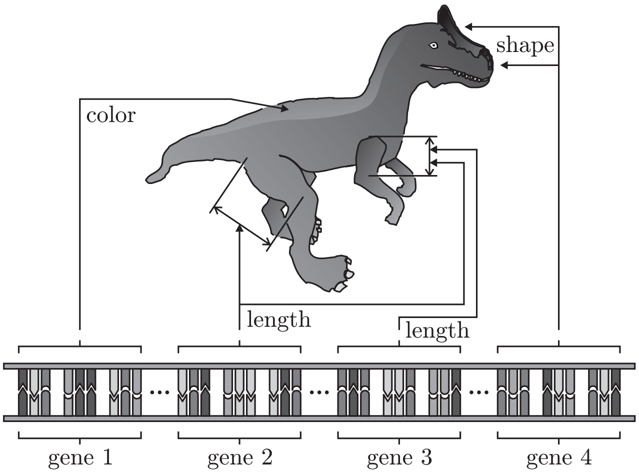


Figure 11: Pleiotropy and epistasis in a dinosaur's genome.

In the area of optimization over continuous problem spaces, epistasis and pleiotropy are closely related to the term *separability*. Separability is a feature of the objective function(s) of an optimization problem [212]. A function of $\ell_{\mathbb{X}}$ variables is separable if it can be rewritten as a sum of $\ell_{\mathbb{X}}$ functions of just one variable [71, 213, 214]. Hence, the genes involved in the problem can be optimized independently of each other, i.e., are minimally epistatic, and the problem is said to be separable. A function $f : \mathbb{R}^{\ell_{\mathbb{X}}} \mapsto \mathbb{R}$ is separable [215, 216] if and only if Equation 1 holds.

Otherwise, $f(\vec{x})$ is called a non-separable

$$\arg\min_{\left(x_1,..,x_{\ell_{\mathbb{X}}}\right)} f(x_1,..,x_{\ell_{\mathbb{X}}}) = \left(\arg\min_{x_1} f(x_1,..)\,,..,\arg\min_{x_{\ell_{\mathbb{X}}}} f(..,x_{\ell_{\mathbb{X}}})\right) \qquad (1)$$

function. If a function $f(\vec{x})$ is separable, the parameters $x_1,..,x_{\ell_{\mathbb{X}}}$ forming the candidate solution $\vec{x}$ are called independent. A separable problem is *decomposable*. A function $f : \mathbb{R}^{\ell_{\mathbb{X}}} \mapsto \mathbb{R}$ is k-non-separable if at most k of its parameters x_i are not independent. A non-separable function $f(\vec{x})$ is called fully non-separable if any two of its parameters x_i are not independent. The higher the degree of non-separability, the harder a function will usually become for optimization [215–217]. Often, the term *non-separable* is used in the sense of *fully non-separable*. In between separable and fully non-separable problems, a variety of *partially* separable problems exist.

6.1 The Issue

As sketched in Figure 12, epistasis has a strong influence on many of the previously discussed issues. If one variable (gene) of a point (genotype) in the search space can "turn off" or affect the expression of other genes, modifying this gene will lead to a large change in the features of the phenotype. Hence, the *causality* will be weakened and *ruggedness* ensues in the fitness landscape. It also becomes harder to define search operations with an exploitive character. Moreover, subsequent changes to the "deactivated" genes may have no influence on the phenotype at all, which would then increase the degree of *neutrality* in the search space. Representations and genotypes with low pleiotropy often lead to better and more robust solutions [218].

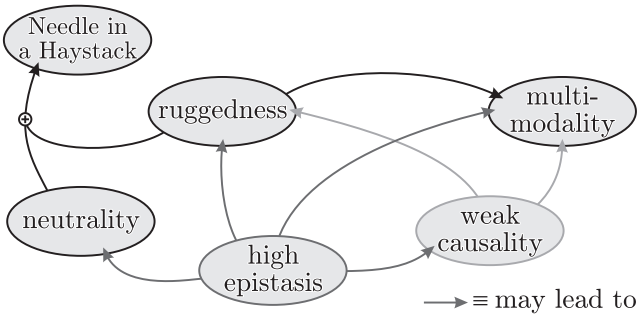


Figure 12: The influence of epistasis on the fitness landscape.

6.2 Countermeasures

Epistasis is a root cause for multiple related issues in optimization tasks. The symptoms of epistasis can be mitigated with the same methods that increase the chance of finding good solutions in the presence of ruggedness or neutrality. Other methods are discussed in the following.

6.2.1 *Choice of the Representation*

Epistasis itself is again an issue resulting from the choice of the search space structure, the search operations, the genotype-phenotype mapping, and the structure of the problem space. Avoiding epistatic effects should be a major concern during the design phase. Choosing the solution space and the genotype-phenotype mapping correctly can lead to great improvements in the quality of the solutions produced by the optimization process [145, 146, 199, 219]. Introducing specialized search operations can achieve similar effects [220].

6.2.2 *Adjusting Selection Pressure*

Using larger populations and favoring explorative search operations could be helpful in epistatic problems, since these are ways to increase diversity. On the other hand, applying *1)* higher selection pressure, i.e., increasing the chance of picking the best candidate solutions for further investigation instead of the weaker ones, and *2) extinctive selection*, i.e., only working with the newest produced set of candidate solutions while discarding their parents, can also increase the reliability of an optimizer to find good solutions [220]. These two concepts are slightly contradicting, so careful adjustment of the algorithm settings appears to be vital in epistatic environments. Higher selection pressure also leads to earlier convergence [220], a fact we already discussed in Section 2.

6.2.3 Linkage and Interaction Learning

According to [221], *linkage* is "the tendency for alleles of different genes to be passed together from one generation to the next" in genetics. This usually indicates that these genes are closely located in the same chromosome. In the context of EAs, this notation is not useful since identifying spatially close elements inside the genotypes is trivial. Instead, we are interested in different genes that have a joint effect on the fitness [222, 223].

Identifying these linked genes, i.e., learning their epistatic interaction, is very helpful for the optimization process. Such knowledge can be used to protect building blocks from being destroyed by the search operations (such as crossover in GAs), for instance. Finding approaches for *linkage learning* for binary [222, 224, 225] and real-valued [226] genomes has become a popular research area. Two important methods derived from this research are the *messy GA* (mGA) [227] and the *Bayesian Optimization Algorithm* (BOA) [228, 229].

Module acquisition [230] may be considered as such an effort too. Here, an additional reproduction operation can group connected components of a genotype together into an atomic group, which becomes immune to modification by other reproduction operators. In GP, this is similar to adding a new automatically defined function that represents a subtree of the program individual.

Especially promising in numerical optimization is the Variable Interaction Learning (VIL) technique [231] that can detect which genes have non-separable relations. These are then grouped together and the resulting division of the genotypes can be optimized separately in a cooperative-coevolution approach [231–233], see Section 9.2.5.

7 Noise and Robustness

Noise is an undesired and unpredictable random disturbance to a signal. In the context of optimization, three types of noise can be distinguished [234]. The first form is noise in the objective functions or in the training data used [235]. In many applications of Machine Learning or optimization where a model for a given system is to be learned, data samples including the input of the system and its measured response are used for training. Besides inexactnesses and fluctuations in the input data of the optimization process, perturbations are also likely to occur during the application of its results, which takes place after the optimization has finished. This category subsumes the other two types of noise: perturbations that may arise from inaccuracies in the process of realizing the solutions *and* environmentally induced perturbations during the applications of the products. The effects of noise in optimization have been the subject of many studies [236–239]. Many optimization algorithms and theoretical results have been proposed to deal with noise. Some of them are, for instance, specialized GAs [240–245], Evolution Strategies (ESs) [246–248], and PSO algorithms [249, 250].

7.1 The Issue: Need for Robustness

The goal of optimization is to find the global optima of the objective functions. While this is fully true from a theoretical point of view, it may not suffice in practice. Optimization problems are normally used to find good parameters or designs for components or plans to be put into action by human beings or machines. As we have discussed, there will always be noise and perturbations in practical realizations of the results of optimization. Designs, plans, and procedures must address the fact that no process is perfect. As a result, practitioners may desire a relatively good and yet predictable solution that can tolerate a certain degree of imprecision during its application in lieu of a less predictable but globally optimal solution.

A system in engineering or biology is *robust* if it is able to function properly in the face of genetic or environmental perturbations [164, 166, 210]. A local optimum (or even a non-optimal element) for which slight disturbances only lead to gentle performance degenerations is usually favored over a global optimum located in a highly rugged area of the fitness landscape [251]. In other words, local optima in regions of the fitness landscape with strong causality are sometimes better than global optima with weak causality. Of course, the level of this acceptability is application-dependent. Figure 13 illustrates the issue of local optima which are robust vs. global optima which are not.

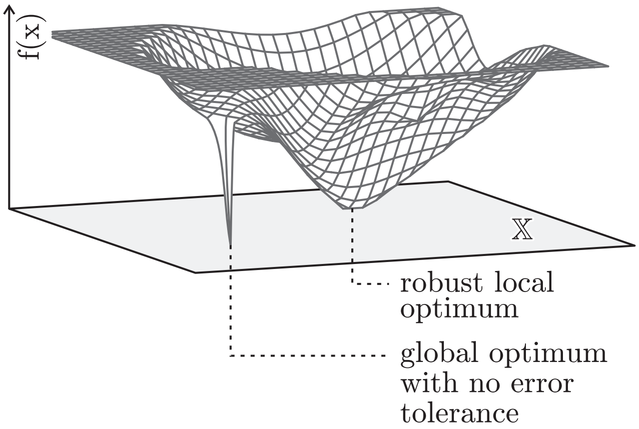


Figure 13: A robust local optimum vs. an "unstable" global optimum.

7.2 Countermeasures

For the special case where the solution space is a real vector space, several approaches for dealing with the need for robustness have been developed. Inspired by Taguchi Methods [252], possible disturbances are represented by a vector $\vec{\delta}$ in the method suggested in [253, 254]. The objective function can be rewritten as $\tilde{f}\left(\vec{x}, \vec{\delta}\right)$ [255] if $\vec{\delta}$ follows a stochastic distribution with known (measured, approximated) parameters. The probability distribution of $\vec{\delta}$ then can be sampled a number of t times and the mean values of $\tilde{f}\left(\vec{x}, \vec{\delta}\right)$ are used during the optimization process [255].

This method turns the optimization algorithm into something like a maximum likelihood estimator and also corresponds to using multiple, different training scenarios during the objective function evaluation. By adding random noise and artificial perturbations to the training cases, the chance of obtaining robust solutions that are stable when applied or realized under noisy conditions can be higher.

8 Dimensionality

Many engineering or scheduling problems involve multiple, often conflicting, optimization criteria. In logistic planning tasks [87, 88], for instance, the goals are *1)* to fulfill as many transportation orders within their respective time windows as possible, *2)* at the lowest possible cost, and *3)* with as little CO_2 emissions as possible. We refer to the number m of objective functions of an optimization problem as its *dimension* (or dimensionality). Later in this article, we will discuss issues arising from a high number of decision variables, which we put under the heading *scalability* in Section 9.)

The most common way to define optima in multi-objective problems (MOPs) is to use the Pareto domination relation. A candidate solution x_1 is said to dominate another candidate solution x_2 ($x_1 \prec x_2$) in an m-objective optimization problem if and only if its corresponding vector of objective values $\vec{f}(x_1)$ is (partially) less than the one of x_2, i.e., $i \in 1..m \Rightarrow f_i(x_1) \leq f_i(x_2)$ and $\exists i \in 1..m : f_i(x_1) < f_i(x_2)$, in minimization problems. More precisely, this is called *weak* dominance; *strong* dominance requires x_1 to be strictly better than x_2 in all objectives. However, the latter notion is usually not applied in the optimization domain. The solutions in the Pareto optimal set (also called *Pareto set* or *Pareto efficient frontier*) are not (weakly) dominated by any other solution in the problem space, i.e., globally optimal with respect to the dominance relation [256–258]. These are the elements we would like to find, or at least approximate as closely as possible, with optimization (see Figure 5 in Section 2.1.1).

Many studies in the literature consider mainly bi-objective problems [259, 260]. Consequently, many algorithms have been designed to deal with that kind of problems. However, MOPs having a higher number of objective functions are common in practice – sometimes the number of objectives reaches double figures [261] – leading to the so-called *many-objective optimization* [48, 259, 262–264]. This term has been coined by the Operations Research community to denote problems with more than two or three objective functions [265, 266].

8.1 The Issue: Many-Objective Optimization

When the dimension of MOPs increases, the majority of the candidate solutions become non-dominated. Traditional multi-objective EAs (MOEAs), however, assign fitness mainly based on information about the Pareto domination relation in the population, usually combined with some diversity metric. Examples include the NSGA-II [267] (Pareto rank combined with the crowding distance in the objective space), SPEA-2 [86] (Pareto-domination based strength together with distance to the k nearest neighbor in the objective space), and PESA [268] (Pareto domination and number of other individuals in the same hyper-box in a grid defined over the search space). It thus can be assumed that Pareto-based optimization approaches (maybe extended with diversity preservation methods) will not perform well in problems with four or more objectives [269, 270]. Results from the application of such algorithms to two or three objectives cannot simply be extrapolated to higher numbers of optimization criteria [259]. In [271], Pareto optimality is considered as unfair and imperfect in many-objective problems and [270] indicated that:

1. an optimizer that produces an entire Pareto set in one run is better than generating the Pareto set through many single-objective optimizations using an aggregation approach if the number of objective function evaluations is fixed, and that
2. optimizers that use Pareto ranking based methods to sort the population will be

very effective for small numbers of objectives, but not perform as effectively for many-objective optimization in comparison with methods based on other approaches.

The results in [259, 262, 272, 273] further demonstrated the degeneration of the performance of traditional multi-objective metaheuristics in many-objective problems in comparison with single-objective approaches. Various elements distant from the true Pareto frontier may survive as hardly-dominated solutions and lead to a decrease in the probability of producing new candidate solutions dominating the existing ones [274]. This phenomenon is called *dominance resistance*. The problem of *redundant solutions* is recognized and demonstrated with an example function (provided as part of a test function suite for continuous multi-objective optimization) in [275].

In addition to these algorithm-sided limitations, [276] suggested that a human mind [277] will not be able to make efficient decisions if more than a dozen of objectives are involved. Visualizing the solutions in a human-understandable way becomes more complex with the rising number of dimensions too [278].

The number of non-dominated elements in random samples increases quickly with the dimension [279]. The hyper-surface of the Pareto frontier may increase exponentially with the number of objective functions [278]. Like in [278], we would like to illustrate this issue with an experiment.

Assume that a population-based optimization approach is used to solve a many-objective problem. The algorithm will fill the initial population with n randomly created individuals. The distribution of the probability $P(\#dom = 0|m, n)$ that a randomly selected individual from this initial population is non-dominated (in this population) depends on the population size n and the number of objective functions m. We have *approximated* this probability distribution using experiments with n m-dimensional vectors where each element is drawn from the same uniform distribution for several values of m spanning from $m = 2$ to $m = 50$ and with $n = 3$ to $n = 3600$.

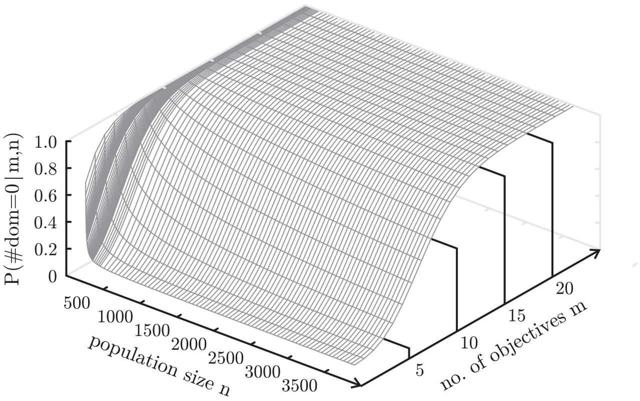


Figure 14: The proportion $P(\#dom = 0|m, n)$ of non-dominated candidate solutions for several population sizes n and dimensionalities m.

The fraction of non-dominated elements in the random populations is illustrated in Figure 14, based on the arithmetic means of 100 000 runs for each configuration. It rises (roughly) exponentially with m, whereas the population size n seems to have only an approximately logarithmically positive influence. If we list the population sizes required to keep the fraction of non-dominated candidate solutions at the same level as in the case of $n = 5$ and $m = 2$ (at around 0.457), we find that for $m = 3 \Rightarrow n \approx 12$, for $m = 4 \Rightarrow n \approx 35$, for $m = 5 \Rightarrow n \approx 90$, for $m = 6 \Rightarrow n \approx 250$, for $m = 7 \Rightarrow n \approx 650$, and for $m = 8 \Rightarrow n \approx 1800$. An extremely coarse rule of thumb here would hence be that around $0.6e^m$ individuals are required in the population to hold the proportion of non-dominated candidate solutions at around 46% in this experiment.

The increasing dimensionality of the objective space leads to three main problems [278]:

1. The performance of traditional ap-

proaches based solely on Pareto comparisons deteriorates.

2. The utility of the solutions cannot be understood by the human operator anymore.
3. The number of possible Pareto-optimal solutions may increase exponentially.

8.2 Countermeasures

Various countermeasures have been proposed against the problem of dimensionality. Surveys on current approaches to many-objective optimization with EC methods, to the difficulties arising in many-objective and on benchmark problems, have been provided in [260, 278, 280]. In the following we list a number of approaches for many-objective optimization, some of which are based on the information provided in [260, 278].

8.2.1 Increasing the Population Size

The most trivial measure is to increase the population size. This, however, works only for a few objective functions and we have to “throw” in exponentially more individuals in order to neutralize the influence of many objectives (as can be seen in Figure 14). Hence, increasing the population size will not get us far.

8.2.2 Multi-Archive Approaches

On large numbers of objectives, traditional MOEAs usually exhibit *either* convergence close to the Pareto front *or* a good spread alongside it [259, 262]. One possible solution is to use two archives [281, 282] in the algorithms: one for diversity and one for convergence, with the goal to combine the two positive features.

8.2.3 Increasing the Selection Pressure

The way multi-objective approaches scale with increasing dimensionality can be improved by increasing the selection pressure into the direction of the Pareto frontier. Ishibuchi *et al.* [278] distinguished approaches that modify the definition of *domination* in order to reduce the number of non-dominated candidate solutions in the population [283] and methods that assign different ranks to non-dominated solutions [284–289]. Relying on fuzzy Pareto methods instead of pure Pareto comparisons is proposed in [266].

8.2.4 Indicator Function-based Approaches

Fitness assignment methods not based on Pareto dominance can also be applied [278]. One approach is to use indicator functions such as those involving hypervolume metrics [290, 291]. Hypervolume metrics have been shown to be able to approximate the Pareto frontier [292].

8.2.5 Scalarizing Approaches

Another possible countermeasure is to use scalarizing functions [278] for fitness assignment in order to treat many-objective problems with single-objective style methods. Several studies [270, 290, 293] showed that this method can produce better results than applying traditional MOEAs such as NSGA-II [267] or SPEA 2 [86], but also refuted the idea that Pareto-based algorithms cannot cope with their performance in general. Other scalarizing methods can be found in [272, 294–296].

8.2.6 Limiting the Search Area in the Objective Space

Furthermore, we can limit the search area in the objective space [260]. This leads to a decrease in the number of non-dominated points [278] and can be achieved by either incorporating preference information [297–299] or by reducing the dimensionality [300–304].

8.2.7 Visualization Methods

Approaches for visualizing solutions of many-objective problems in order to make them more comprehensible have been provided in [305–308].

9 Scalability

An increasing number of objective functions can threaten the performance of optimization algorithms. We referred to this as the *dimensionality* problem, i.e., the dimension of the objective space. There is another space-related issue – the *"curse of dimensionality"* of the search space, i.e., the exponential increase of its volume with the number of genes (or decision variables) [309, 310]. To better distinguish between the dimensionality of the objective space and the search space, we will refer to the latter as *scale*.

As an example, we illustrate small-scale versus large-scale problems using discrete or continuous vector-based search spaces. If we search, for instance, on one gene having values in the natural interval 1..10, there are ten points that could be the optimal solution. When the search space is composed of two such genes, i.e., $(1..10)^2$, there exist one hundred possible results and for $(1..10)^3$, it is already one thousand. In other words, the number of elements that could be a solution to an optimization problem grows exponentially with the number of genes.

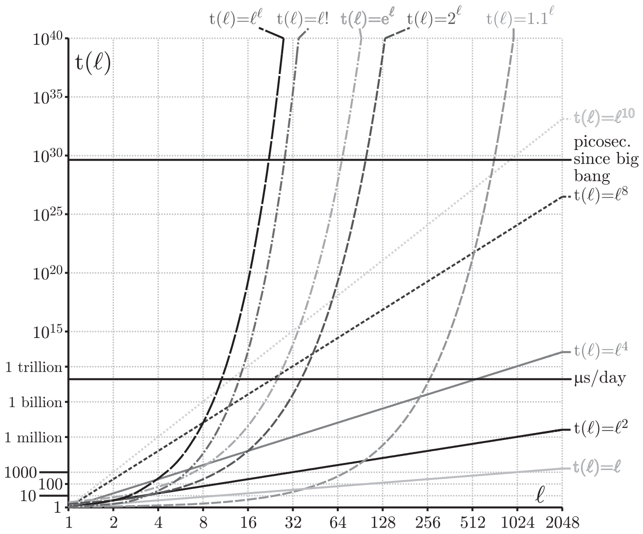


Figure 15: Illustration of the rising speed of some functions, inspired by [311].

9.1 The Issue

The issue of scale has already been introduced in Section 1.3, where we discussed the *computational complexity* as a measure of how many algorithm steps are needed to solve a problem consisting of $\ell_{\mathbb{X}}$ decision variables. As can be seen in Figure 15, if the number $t(\ell_{\mathbb{X}})$ of algorithm steps, i.e., the runtime, needed to solve a problem grows exponentially with the problem size $\ell_{\mathbb{X}}$, it quickly exceeds any feasible bound. However, in Section 1.3, the issue was considered from the perspective of deterministic algorithms, which are supposed to solve a problem to optimality. As a remedy for the infeasible runtime of these algorithms, we then suggested to apply stochastic optimization methods. Although these may be able to solve problems with a several magnitudes higher scale in a close-to-optimal way, their performance deteriorates with rising scales too.

[8] Actually, only a sub-linear speed-up can be achieved [312].

9.2 Countermeasures

9.2.1 Parallelization and Distribution

When facing problems of large scale, the main "obstacle" is the high runtime requirement. Thus, any measure of using as much computational power as available can be a remedy. Obviously, there (currently) exists no way to solve large-scale $\mathcal{NP}$-hard problems exactly within feasible time. With more computers, cores, or hardware, only a linear improvement[8] of runtime can be achieved at most. However, for problems residing in the grey area between feasible and infeasible, distributed computing [313] may be the method of choice.

There is a long tradition of parallelizing and distributing the computational workload in EC [314–317]. The basic ways to parallelize an EA are:

1. *Local Parallelization* To parallelize the execution by using hardware with multiple CPUs [318] in a single computer, or, as is the current trend,

2. by utilizing modern Graphics Processing Units (GPUs) [319–322] to evaluate and process the individuals in a population in parallel.

3. *Parallel Restarts* It is also possible to run different instances of the same algorithm on multiple CPUs or computers in a network at the same time, which would be a parallel version of the restarting strategy.

4. *Master/Slave Approach* [323] If the evaluation of a candidate solution is very time consuming, this step can be parallelized to several workers (threads or computers in a network), which receive their task from a single central server maintaining a global population.

5. *Island Model* [324] Alternatively, each node (or thread) may maintain an own population and, from time to time, exchange promising candidate solutions with neighboring nodes in the topology.

6. Of course, any combination of the above is possible [317].

9.2.2 Generative Representations

Another way, possibly the best way, to tackle a large-scale problem is to "solve" it as a small-scale problem. For some optimization tasks, it is possible to choose a search space $\mathbb{G}$ having a smaller size (e.g., a small number $\ell_{\mathbb{G}}$ of genes) than the problem space $\mathbb{X}$ (e.g., having $\ell_{\mathbb{X}} > \ell_{\mathbb{G}}$ decision variables). Indirect genotype-phenotype mappings can link the spaces together.

Here, one option is the generative mapping, which step-by-step constructs a complex phenotype by translating a genotype according to some static rules. Grammatical Evolution [325, 326], for instance, unfolds a start symbol according to a grammar with rules identified in a genotype. This recursive process can basically lead to arbitrarily complex phenotypes.

9.2.3 Developmental Representations

Applying a developmental, ontogenic mapping [20, 327] that uses feedback from simulations or objective functions in the process of building a candidate solution is another possible countermeasure. If, for instance, a rigid truss composed of $\ell_{\mathbb{X}} = 600$ beams is to be found, instead of optimizing the volumes of each of the beams directly, the goal would be to find a suitable function that receives as a parameter the mechanical stress on a given beam and returns how much the cross section of the beam should be increased.

Beginning with a basic beam structure, the mechanical stress is evaluated and the function is applied to each of the beams. The updated truss is simulated again and the process is repeated a couple of times. The resulting structure would be the phenotype. The genotype can be an Artificial Neural Network representing the function, encoded as real vectors containing the neural weights, thus having much fewer variables (e.g., $\ell_{\mathbb{G}} = 12$). Moreover, $\ell_{\mathbb{G}}$ is independent from $\ell_{\mathbb{X}}$, and therefore, much larger problem spaces can become tangible and excellent results may be obtained in reasonable time, likely with better quality and faster than using generative mappings [20].

9.2.4 Adaptive Encodings

Somewhat in between purely generative and a developmental approach is the Dynamic Parameter Encoding (DPE) method [23], which is basically a dynamic genotype-phenotype mapping for binary-encoded real vectors. Traditionally, the number of bits in each gene is fixed and corresponds to the desired precision. In DPE, the interval in the problem space represented by each gene is assigned dynamically, iteratively shrinking down from the full range: If the GA used for optimization has converged to some values of the bits of a gene, a *zooming* operation changes the meaning of that gene to now represent the corresponding sub-interval only. This way, the number of bits needed to achieve a given solution precision can be significantly reduced. In other words, although it still needs one gene in the genotype per decision variable in the phenotype, the genes themselves only consist of a few bits (e.g., three) and are thus much more compact than in fixed genotype-phenotype mappings.

9.2.5 Exploiting Separability

If a large-scale problem cannot be solved as a single small-scale problem, solving it as multiple small-scale problems may be another option for saving runtime. Sometimes, parts of candidate solutions are independent from each other and can be optimized more or less separately. In such a case (low epistasis, see Section 6), a large-scale problem can be divided into several components of smaller-scale to be optimized separately. If solving a problem of scale $\ell_{\mathbb{X}}$ takes $2^{\ell_{\mathbb{X}}}$ algorithm steps, solving two problems of scale $0.5\ell_{\mathbb{X}}$ will clearly lead to a great runtime reduction. Such a reduction may even be worth the sacrifice of some solution quality. If the optimization problem at hand exhibits low epistasis or is separable, such a sacrifice may even be avoided.

Coevolution has shown to be an efficient approach in combinatorial optimization [328]. If extended with a cooperative component (i.e., to Cooperative Coevolution [231–233]), it can efficiently exploit separability in numerical problems and lead to better results [231, 329].

9.2.6 Combination of Techniques

Generally speaking, it can be a good idea to concurrently use different sets of algorithms [330] or portfolios [331] to work on the same or different populations. This way, the strengths of different optimization methods can be combined. In the beginning, for instance, an algorithm with good convergence speed may be granted more runtime. Later, the focus can shift towards methods that can retain diversity and are not prone to premature convergence. Alternatively, a sequential approach can be performed, which starts with one algorithm and switches to another one when no further improvements can be found [144]. By doing this, an interesting area in the search space can first be discovered, and then be investigated more thoroughly.

10 The No Free Lunch Theorem

So far, we have discussed various difficulties that could arise when applying an optimization algorithm to a given problem. The fact that not a single optimization method is likely to be able to outperform all other methods on all problems can easily be accepted. Instead, we see a variety of optimization methods specialized in solving different types of problems. There are also algorithms that may deliver good results for many different problem classes, but could be outperformed by highly specialized methods in each of them. These facts have been formalized by Wolpert and Macready [332] in their *No Free Lunch Theorems* for search and optimization algorithms.

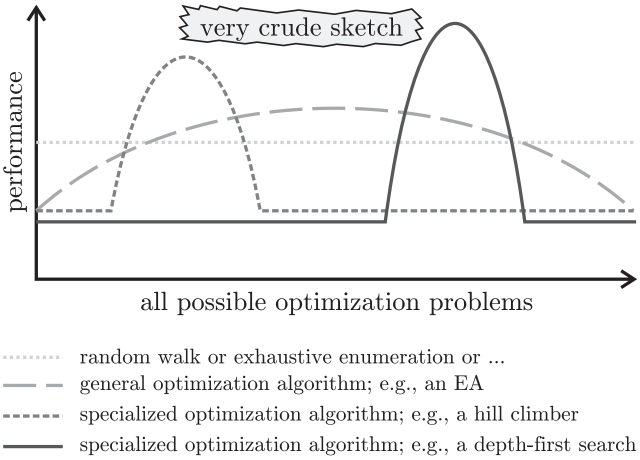


Figure 16: A visualization of the No Free Lunch Theorem.

The performance of an algorithm a executed for p steps on an optimization problem can be defined as the conditional probability of finding a particular sample (such as the global optimum). Wolpert and Macready [332] proved that the sum of such probabilities over all possible optimization problems on *finite*[9] domains is always identical for all optimization algorithms. This means that the average performance over all finite problems is independent of the algorithm applied. From this theorem, we can immediately follow that, in order to outperform algorithm a_1 in one optimization problem, algorithm a_2 will necessarily perform worse in another problem, as sketched in Figure 16. This implies that it is impossible for any optimization algorithm to always outperform non-repeating random walks or exhaustive enumerations.

In practice, an optimizer is not applied to all possible problems but to only some, restricted classes. In terms of these classes, it is well possible to perform comparisons and to make statements regarding which algorithms perform the best (which, by the way, is often the topic of challenges and competitions [212, 214]).

Another interpretation of the No Free Lunch Theorem is that every useful optimization algorithm utilizes some form of problem-specific knowledge. In [335], it is stated that without such knowledge, search algorithms cannot exceed the performance of simple enumerations. Incorporating knowledge starts with relying on simple assumptions like causality (see Section 3.1). The more problem specific knowledge is integrated into the algorithm structure, the better the algorithm can perform [20].

11 Concluding Remarks

The subject of this article is to address questions about issues that make optimization problems difficult to solve, with a particular focus on evolutionary optimization. We have discussed a variety of scenarios that can influence/affect the optimization process and lead to disappointing results.

If an optimization process has converged prematurely, it is said to be trapped in a non-optimal region of the search space from which it cannot “escape” anymore (Section 2). Ruggedness (Section 3) and deceptiveness (Section 4) in the fitness landscape, often caused by epistatic effects (Section 6), can misguide

[9] Recently, it was shown that the No Free Lunch Theorem holds only in a weaker form for countable infinite and not for continuous domains [333, 334].

the search into such a region. Neutrality and redundancy (Section 5) may either slow down optimization or contribute positively. Noise is present in virtually all practical optimization problems. The solutions that are derived for them should thus be robust (Section 7). Also, many practical problems are multi-objective in nature, i.e., involve the optimization of more than one criterion at a time (see Section 8).

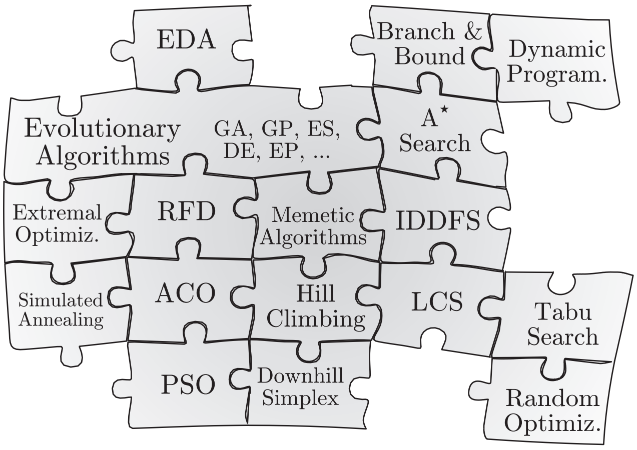


Figure 17: The puzzle of optimization algorithms.

The No Free Lunch Theorem argues that it is not possible to develop a *universal* optimization algorithm, the problem-solving machine that can provide us with near-optimal solutions in short time for every possible optimization task in finite domains. Such a statement may sound depressing for those who are new to this subject.

Actually, quite the opposite is the case, at least from the point of view of a researcher. The No Free Lunch Theorem means that there will always be new ideas, new approaches that will lead to better optimization algorithms to solve a given problem. Instead of being doomed to obsolescence, it is far more likely that most of the currently known optimization methods have at least one niche, one area where they could excel in. This fact has contributed to the emergence of memetic, hybrid and the new area of portfolio-type algorithms [331], which combine different optimization methods.

It is most likely that the "puzzle"[10] of optimization algorithms as sketched in Figure 17 will never be completed. There will always be a chance that an inspiring moment, an observation in nature, for instance, may lead to the invention of a new optimization algorithm that performs better in some problem areas than all the currently known ones.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grants U0835002, 61175065, and 61150110488, by the National Natural Science Foundation of Anhui Province under Grant 1108085J16, by the European Union 7th Framework Program under Grant 247619, by the Chinese Academy of Sciences Fellowship for Young International Scientists CX05040000001, and by a Special Financial Grant from the China Postdoctoral Science Foundation (number 201104329).

References

1. Thomas Weise, Raymond Chiong, and Kē Táng. Evolutionary Optimization: Pitfalls and Booby Traps. *Journal of Computer Science and Technology (JCST)*, 27(5):907–936, September 2012. doi: 10.1007/s11390-012-1274-4. Special Issue on Evolutionary Computation, edited by Xin Yao and Pietro S. Oliveto.
2. Christian Blum, Raymond Chiong, Maurice Clerc, Kenneth Alan De Jong, Zbigniew Michalewicz, Ferrante Neri, and Thomas Weise. Evolutionary Optimization. In Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz, editors, *Variants of Evolutionary Algorithms for Real-World Applications*, chapter 1, pages 1–29. Springer-Verlag: Berlin/Heidelberg, 2011. doi: 10.1007/978-3-642-23424-8_1.
3. Thomas Weise. *Global Optimization Algorithms – Theory and Application.* it-weise.de (self-published): Germany, 2009. URL `http://www.it-weise.de/projects/book.pdf`.

[10] So far undefined abbreviations: RFD. . . River Formation Dynamics, IDDFS. . . Iteratively Deepening Depth-First Search, ACO. . . Ant Colony Optimization, LCS. . . Learning Classifier Systems

4. Ágoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer New York: New York, NY, USA, 1st edition, November 2003. ISBN ISBN: 3540401849.
5. Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz, editors. *Variants of Evolutionary Algorithms for Real-World Applications*. Springer-Verlag: Berlin/Heidelberg, 2011. doi: 10.1007/978-3-642-23424-8.
6. L. Darrell Whitley. A Genetic Algorithm Tutorial. *Statistics and Computing*, 4(2):65–85, June 1994. doi: 10.1007/BF00175354. URL `http://samizdat.mines.edu/ga_tutorial/ga_tutorial.ps`.
7. Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag GmbH: Berlin, Germany, 1996. ISBN ISBN: 3-540-58090-5 and ISBN: 3-540-60676-9.
8. Marek Obitko. Introduction to Genetic Algorithms, 1998.
9. Carlos Artemio Coello Coello. A Short Tutorial on Evolutionary Multiobjective Optimization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos Artemio Coello Coello, and David Wolfe Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*, volume 1993/2001 of *Lecture Notes in Computer Science (LNCS)*, pages 21–40. Springer-Verlag GmbH: Berlin, Germany, 2001. URL `http://www.cs.cinvestav.mx/~emooworkgroup/tutorial-slides-coello.pdf`.
10. Carlos Artemio Coello Coello. Theoretical and Numerical Constraint-Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12): 1245–1287, January 14, 2002. doi: 10.1016/S0045-7825(01)00323-1.
11. Krzysztof Trojanowski and Zbigniew Michalewicz. Evolutionary Algorithms and the Problem-Specific Knowledge. In *Proceedings of the Second National Conference on Evolutionary Computation and Global Optimization (Materiały II Krajowej Konferencji Algorytmy Ewolucyjne i Optymalizacja Globalna)*, pages 281–292, 1997.
12. Raymond Chiong, editor. *Nature-Inspired Algorithms for Optimisation*, volume 193/2009 of *Studies in Computational Intelligence*. Springer-Verlag: Berlin/Heidelberg, April 30, 2009. ISBN ISBN: 3-642-00266-8 and ISBN: 3-642-00267-6. doi: 10.1007/978-3-642-00267-0.
13. Raymond Chiong and Sandeep Dhakal, editors. *Natural Intelligence for Scheduling, Planning and Packing Problems*, volume 250 of *Studies in Computational Intelligence*. Springer-Verlag: Berlin/Heidelberg, October 2009. ISBN ISBN: 3-642-04038-1. doi: 10.1007/978-3-642-04039-9.
14. Raymond Chiong, editor. *Nature-Inspired Informatics for Intelligent Applications and Knowledge Discovery: Implications in Business, Science and Engineering*. Information Science Reference: Hershey, PA, USA, 2009. ISBN ISBN: 1605667056. doi: 10.4018/978-1-60566-705-8.
15. Tianshi Chen, Kē Táng, Guóliáng Chén, and Xīn Yáo. Analysis of Computational Time of Simple Estimation of Distribution Algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 14(1):1–22, February 2010. doi: 10.1109/TEVC.2009.2040019. URL `http://staff.ustc.edu.cn/~ketang/papers/ChenTangChenYao_TEVC10.pdf`.
16. Tianshi Chen, Jun He, Guóliáng Chén, and Xīn Yáo. Choosing Selection Pressure for Wide-Gap Problems. *Theoretical Computer Science*, 411(6): 926–934, February 6, 2010. doi: 10.1016/j.tcs.2009.12.014.
17. Jun He and Xīn Yáo. Towards an Analytic Framework for Analysing the Computation Time of Evolutionary Algorithms. *Artificial Intelligence*, 145(1-2):59–97, April 2003. doi: 10.1016/S0004-3702(02)00381-8. URL `http://www.cs.bham.ac.uk/~xin/papers/published_aij_03.pdf`.
18. Jun He, Colin R. Reeves, Carsten Witt, and Xīn Yáo. A Note on Problem Difficulty Measures in Black-Box Optimization: Classification, Realizations and Predictability. *Evolutionary Computation*, 15(4):435–443, 2007. doi: 10.1162/evco.2007.15.4.435.
19. Darrell F. Lochtefeld and Frank William Ciarallo. A Diversity Classification Scheme for Genetic Algorithms. In Eileen M. Van Aken and Toni L. Doolen, editors, *Industrial Engineering Research Session of the 61st Annual IIE Conference and Expo (IERC'11)*. Institute of Industrial Engineers (IIE): Norcross, GA, USA, 2011.
20. Alexandre Devert, Thomas Weise, and Kē Táng. A Study on Scalable Representations for Evolutionary Optimization of Ground Structures. *Evolutionary Computation*, 20(3): 453–472, 2012. doi: 10.1162/EVCO_a_00054. URL `http://www.marmakoide.org/download/publications/devweita-ecj-preprint.pdf`.

21. Kenneth Alan De Jong. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan: Ann Arbor, MI, USA, August 1975. URL `http://cs.gmu.edu/~eclab/kdj_thesis.html`.
22. Alden H. Wright. Genetic Algorithms for Real Parameter Optimization. In Bruce M. Spatz and Gregory J. E. Rawlins, editors, *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)*, pages 205–218. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990.
23. Nicol N. Schraudolph and Richard K. Belew. Dynamic Parameter Encoding for Genetic Algorithms. *Machine Learning*, 9(1):9–21, June 1992. doi: 10.1023/A:1022624728869. URL `http://cnl.salk.edu/~schraudo/pubs/SchBel92.pdf`.
24. David Edward Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, 1989. ISBN ISBN: 0-201-15767-5.
25. Franz Rothlauf. *Representations for Genetic and Evolutionary Algorithms*, volume 104 of *Studies in Fuzziness and Soft Computing.* Physica-Verlag GmbH & Co.: Heidelberg, Germany, 2nd edition, 2002. ISBN ISBN: 3-540-25059-X and ISBN: 3790814962. Foreword by David E. Goldberg.
26. John J. Grefenstette. Deception Considered Harmful. In L. Darrell Whitley, editor, *Proceedings of the Second Workshop on Foundations of Genetic Algorithms (FOGA'92)*, pages 75–91. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1992.
27. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* MIT Electrical Engineering and Computer Science. MIT Press: Cambridge, MA, USA and McGraw-Hill Ltd.: Maidenhead, UK, England, 2nd edition, 1990. ISBN ISBN: 0070131430, ISBN: 0070131449, ISBN: 0-2620-3141-8, ISBN: 0-2625-3196-8, and ISBN: 8120321413.
28. Donald Ervin Knuth. *Fundamental Algorithms*, volume 1 of *The Art of Computer Programming (TAOCP).* Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA, third edition, 1997. ISBN ISBN: 0-201-89683-4.
29. Paul Gustav Heinrich Bachmann. *Die Analytische Zahlentheorie / Dargestellt von Paul Bachmann*, volume Zweiter Theil of *Zahlentheorie: Versuch einer Gesamtdarstellung dieser Wissenschaft in ihren Haupttheilen.* University of Michigan Library, Scholarly Publishing Office: Ann Arbor, MI, USA, 1894. ISBN ISBN: 1418169633, ISBN: 141818540X, and ISBN: 978-1418169633. URL `http://gallica.bnf.fr/ark:/12148/bpt6k994750`.
30. Edmund Landau. *Handbuch der Lehre von der Verteilung der Primzahlen.* B. G. Teubner: Leipzig, Germany and AMS Chelsea Publishing: Providence, RI, USA, 1909. ISBN ISBN: 0821826506.
31. Alan Mathison Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937. doi: 10.1112/plms/s2-42.1.230. URL `http://www.abelard.org/turpap2/tp2-ie.asp`.
32. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Series of Books in the Mathematical Sciences. W. H. Freeman and Company: New York, NY, USA, 1979. ISBN ISBN: 0-7167-1044-7 and ISBN: 0-7167-1045-5.
33. Oded Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press: Cambridge, UK, 2008.
34. Egon Börger. *Computability, Complexity, Logic*, volume 128 of *Studies in Logic and the Foundations of Mathematics.* Elsevier Science Publishing Company, Inc.: New York, NY, USA, 1989. ISBN ISBN: 0-444-87406-2.
35. Benoît Leblanc and Evelyne Lutton. Bitwise Regularity and GA-Hardness. In Patrick K. Simpson, editor, *The 1998 IEEE International Conference on Evolutionary Computation (CEC'98), 1998 IEEE World Congress on Computation Intelligence (WCCI'98)*, pages 517–522. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 1998. doi: 10.1109/ICEC.1998.700082.
36. Bart Naudts and Leila Kallel. Some Facts About So Called GA-hardness Measures. Rapport Interne (R.I.) 379, Centre de Mathématiques APpliquées (CMAP): Palaiseau Cedex, France, January 20, 1998. URL `ftp://cmapx.polytechnique.fr/pub/RI/1998/naudts_kallel_379.mars.ps.gz`.
37. Yossi Borenstein and Riccardo Poli. Fitness Distributions and GA Hardness. In Xīn Yáo, Edmund K. Burke, José Antonio Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiño, Ata Kabán, and Hans-Paul Schwefel, editors, *Proceed-*

ings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII), volume 3242/2004 of *Lecture Notes in Computer Science (LNCS)*, pages 11–20. Springer-Verlag GmbH: Berlin, Germany, 2008. doi: 10.1007/978-3-540-30217-9_2. URL http://privatewww.essex.ac.uk/~yboren/ppsn163.pdf.

38. Haipeng Guo and William H. Hsu. GA-Hardness Revisited. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell K. Standish, Graham Kendall, Stewart W. Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitchell A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasa Jonoska, and Julian Francis Miller, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, Part I (GECCO'03)*, volume 2723/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 1584–1585. Springer-Verlag GmbH: Berlin, Germany, 2003. doi: 10.1007/3-540-45110-2_36.

39. Pietro S. Oliveto, Jun He, and Xīn Yáo. Time Complexity of Evolutionary Algorithms for Combinatorial Optimization: A Decade of Results. *International Journal of Automation and Computing*, 4(3):281–293, July 2007. doi: 10.1007/s11633-007-0281-3. URL http://www.cs.bham.ac.uk/~pkl/teaching/2009/ec/articles/oliveto2007review.pdf.

40. Pietro S. Oliveto, Jun He, and Xīn Yáo. Analysis of the (1 + 1)-EA for Finding Approximate Solutions to Vertex Cover Problems. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 13(5):1006–1029, October 2009. doi: 10.1109/TEVC.2009.2014362.

41. Jeffrey Horn and David Edward Goldberg. Genetic Algorithm Difficulty and the Modality of the Fitness Landscape. In L. Darrell Whitley and Michael D. Vose, editors, *Proceedings of the Third Workshop on Foundations of Genetic Algorithms (FOGA 3)*, pages 243–269. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1994.

42. Soraya B. Rana. *Examining the Role of Local Optima and Schema Processing in Genetic Search.* PhD thesis, Colorado State University, Department of Computer Science, GENITOR Research Group in Genetic Algorithms and Evolutionary Computation: Fort Collins, CO, USA, July 1, 1999. URL http://www.cs.colostate.edu/~genitor/dissertations/rana.ps.gz.

43. Rasmus K. Ursem. *Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization.* PhD thesis, University of Aarhus, Department of Computer Science: Århus, Denmark, April 1, 2003. URL http://www.daimi.au.dk/~ursem/publications/RKU_thesis_2003.pdf.

44. James David Schaffer, Larry J. Eshelman, and Daniel Offutt. Spurious Correlations and Premature Convergence in Genetic Algorithms. In Bruce M. Spatz and Gregory J. E. Rawlins, editors, *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)*, pages 102–112. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990.

45. Kalyanmoy Deb. Genetic Algorithms for Multimodal Function Optimization. Master's thesis, Clearinghouse for Genetic Algorithms, University of Alabama: Tuscaloosa, 1989. TCGA Report No. 89002.

46. Gulshan Singh and Kalyanmoy Deb. Comparison of Multi-Modal Optimization Algorithms based on Evolutionary Algorithms. In Maarten Keijzer and Mike Cattolico, editors, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pages 1305–1312. ACM Press: New York, NY, USA, 2006. doi: 10.1145/1143997.1144200. URL https://ceprofs.civil.tamu.edu/ezechman/CVEN689/References/singh_deb_2006.pdf. Genetic algorithms session.

47. Catalin Stoean, Mike Preuß, Ruxandra Stoean, and Dumitru (Dan) Dumitrescu. Multimodal Optimization by Means of a Topological Species Conservation Algorithm. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 14(6):842–864, December 2010. doi: 10.1109/TEVC.2010.2041668. URL http://inf.ucv.ro/~rstoean/papers/tec2010paper.pdf.

48. Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio Jesús Nebro Urbaneja. Why is optimization difficult? In Raymond Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, volume 193/2009 of *Studies in Computational Intelligence*, chapter 1, pages 1–50. Springer-Verlag: Berlin/Heidelberg, 2009. doi: 10.1007/978-3-642-00267-0_1. URL http://www.it-weise.de/research/publications/WZCN2009WIOD/WZCN2009WIOD.pdf.

49. William Michael Rudnick. *Genetic Algorithms and Fitness Variance with an Application to the Automated Design of Artificial Neural Networks.* PhD thesis, Oregon Graduate Institute of Science & Technology: Beaverton, OR, USA, 1992.

50. Dirk Thierens, David Edward Goldberg, and

Ângela Guimarães Pereira. Domino Convergence, Drift, and the Temporal-Salience Structure of Problems. In Patrick K. Simpson, editor, *The 1998 IEEE International Conference on Evolutionary Computation (CEC'98), 1998 IEEE World Congress on Computation Intelligence (WCCI'98)*, pages 535–540. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 1998. doi: 10.1109/ICEC.1998.700085.

51. Melanie Mitchell, Stephanie Forrest, and John Henry Holland. The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance. In Francisco J. Varela and Paul Bourgine, editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (ECAL'91)*, Bradford Books, pages 245–254. MIT Press: Cambridge, MA, USA, 1991. URL `http://web.cecs.pdx.edu/~mm/ecal92.pdf`.
52. Ingo Paenke, Jürgen Branke, and Yàochū Jīn. On the Influence of Phenotype Plasticity on Genotype Diversity. In Jerry M. Mendel, Takashi Omari, and Xīn Yáo, editors, *The First IEEE Symposium on Foundations of Computational Intelligence (FOCI'07)*, pages 33–40. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/FOCI.2007.372144. URL `http://www.honda-ri.org/intern/Publications/PN\%2052-06`. Best student paper.
53. Anne E. Magurran. Biological Diversity. *Current Biology*, 15(4):R116–R118, February 22, 2005. doi: 10.1016/j.cub.2005.02.006. URL `http://dx.doi.org/10.1016/j.cub.2005.02.006`.
54. John Henry Holland. Genetic Algorithms – Computer programs that "evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand. *Scientific American*, 267(1):44–50, July 1992. URL `http://members.fortunecity.com/templarseries/algo.html`.
55. Fernando G. Lobo, Cláudio F. Lima, and Zbigniew Michalewicz, editors. *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*. Springer-Verlag: Berlin/Heidelberg, March 2007. ISBN ISBN: 3-540-69431-5 and ISBN: 3-540-69432-3. doi: 10.1007/978-3-540-69432-8.
56. John Henry Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press: Ann Arbor, MI, USA, 1975. ISBN ISBN: 0-472-08460-7.
57. Ágoston E. Eiben and C. A. Schippers. On Evolutionary Exploration and Exploitation. *Fundamenta Informaticae – Annales Societatis Mathematicae Polonae, Series IV*, 35(1-2):35–50, July–August 1998. URL `http://www.cs.vu.nl/~gusz/papers/FunInf98-Eiben-Schippers.ps`.
58. Fred Glover. Tabu Search – Part II. *ORSA Journal on Computing*, 2(1):190–206, 1990. doi: 10.1287/ijoc.2.1.4. URL `http://leeds-faculty.colorado.edu/glover/TS\%20-\%20Part\%20II-ORSA-aw.pdf`.
59. Fred Glover, Éric D. Taillard, and Dominique de Werra. A User's Guide to Tabu Search. *Annals of Operations Research*, 41(1):3–28, March 1993. doi: 10.1007/BF02078647. Special issue on Tabu search.
60. Andreas Nolte and Rainer Schrader. A Note on the Finite Time Behaviour of Simulated Annealing. *Mathematics of Operations Research (MOR)*, 25(3):476–484, August 2000. doi: 10.1287/moor.25.3.476.12211. URL `http://www.zaik.de/~paper/unzip.html?file=zaik1999-347.ps`. Revised version from March 1999.
61. Lester Ingber. Simulated Annealing: Practice versus Theory. *Mathematical and Computer Modelling*, 18(11):29–57, November 1993. doi: 10.1016/0895-7177(93)90204-C. URL `http://www.ingber.com/asa93_sapvt.pdf`.
62. Per Kristian Lehre and Xīn Yáo. On the Impact of Mutation-Selection Balance on the Runtime of Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 2012. URL `ftp://ftp.cs.bham.ac.uk/pub/tech-reports/2009/CSR-09-07.pdf`.
63. Pietro S. Oliveto, Per Kristian Lehre, and Frank Neumann. Theoretical Analysis of Rank-based Mutation – Combining Exploration and Exploitation. In *10th IEEE Congress on Evolutionary Computation (CEC'09)*, pages 1455–1462. IEEE Computer Society: Piscataway, NJ, USA, 2009. doi: 10.1109/CEC.2009.4983114. URL `http://cs.adelaide.edu.au/~frank/papers/cec09.pdf`.
64. Ioan Cristian Trelea. The Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection. *Information Processing Letters*, 85(6):317–325, March 31, 2003. doi: 10.1016/S0020-0190(02)00447-7.
65. Günter Rudolph. *Convergence Properties of Evolutionary Algorithms*. PhD thesis, Universität Dortmund: Dortmund, North Rhine-Westphalia,

Germany, 1996. Published 1997.

66. Larry J. Eshelman, Richard A. Caruana, and James David Schaffer. Biases in the Crossover Landscape. In James David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pages 10–19. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1989.

67. Kalyanmoy Deb. Genetic Algorithms for Optimization. KanGAL Report 2001002, Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT): Kanpur, Uttar Pradesh, India, 2001. URL `http://www.iitk.ac.in/kangal/papers/isna.ps.gz`.

68. Nitin Muttil and Shie-Yui Liong. Superior Exploration–Exploitation Balance in Shuffled Complex Evolution. *Journal of Hydraulic Engineering*, 130(12):1202–1205, December 2004. doi: 10.1061/(ASCE)0733-9429(2004)130:12(1202).

69. Heni Ben Amor and Achim Rettinger. Intelligent Exploration for Genetic Algorithms: Using Self-Organizing Maps in Evolutionary Computation. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 1531–1538. ACM Press: New York, NY, USA, 2005. doi: 10.1145/1068009.1068250.

70. Quang Huy Nguyen, Yew-Soon Ong, Meng Hot Lim, and Natalio Krasnogor. Adaptive Cellular Memetic Algorithms. *Evolutionary Computation*, 17(2):231–256, 2009. doi: 10.1162/evco.2009.17.2.231. URL `http://www.cs.nott.ac.uk/~nxk/PAPERS/CMA.PDF`.

71. Domingo Ortiz-Boyer, César Hervás-Martínez, and Carlos A. Reyes García. CIXL2: A Crossover Operator for Evolutionary Algorithms Based on Population Features. *Journal of Artificial Intelligence Research (JAIR)*, 24:1–48, July 2005. URL `http://www.aaai.org/Papers/JAIR/Vol24/JAIR-2401.pdf`.

72. Chang-Yong Lee and Xīn Yáo. Evolutionary Programming using the Mutations based on the Lévy Probability Distribution. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 8(1):1–13, February 2004. doi: 10.1109/TEVC.2003.816583.

73. Xīn Yáo. An Empirical Study of Genetic Operators in Genetic Algorithms. *Microprocessing and Microprogramming*, 38(1-5):707–714, September 1993. doi: 10.1016/0165-6074(93)90215-7. URL `http://www.cs.bham.ac.uk/~xin/papers/euro93_final.pdf`.

74. Paola Festa and Mauricio G.C. Resende. An Annotated Bibliography of GRASP. AT&T Labs Research Technical Report TD-5WYSEW, AT&T Labs: Florham Park, NJ, USA, February 29, 2004. URL `http://www.research.att.com/~mgcr/grasp/gannbib/gannbib.html`.

75. Thomas A. Feo and Mauricio G.C. Resende. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization*, 6(2):109–133, March 1995. doi: 10.1007/BF01096763. URL `http://www.research.att.com/~mgcr/doc/gtut.ps.Z`.

76. Erik van Nimwegen and James P. Crutchfield. Optimizing Epochal Evolutionary Search: Population-Size Dependent Theory. *Machine Learning*, 45(1):77–114, October 2001. doi: 10.1023/A:1012497308906.

77. Tianshi Chen, Kē Táng, Guóliáng Chén, and Xīn Yáo. A Large Population Size Can Be Unhelpful in Evolutionary Algorithms. *Theoretical Computer Science*, 2011. doi: 10.1016/j.tcs.2011.02.016.

78. Jun He and Xīn Yáo. From an Individual to a Population: An Analysis of the First Hitting Time of Population-based Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 6(5):495–511, October 2002. doi: 10.1109/TEVC.2002.800886.

79. Simon Ronald, John Asenstorfer, and Millist Vincent. Representational Redundancy in Evolutionary Algorithms. In *Second IEEE International Conference on Evolutionary Computation (CEC'95)*, volume 2, pages 631–637. IEEE Computer Society Press: Los Alamitos, CA, USA, 1995. doi: 10.1109/ICEC.1995.487457.

80. David Edward Goldberg and Jon T. Richardson. Genetic Algorithms with Sharing for Multimodal Function Optimization. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms and their Applications (ICGA'87)*, pages 41–49. Lawrence Erlbaum Associates, Inc. (LEA): Mahwah, NJ, USA, 1987.

81. Kalyanmoy Deb and David Edward Goldberg. An Investigation of Niche and Species Formation in Genetic Function Optimization. In James David Schaffer, editor, *Proceedings of the Third In-*

ternational Conference on Genetic Algorithms (ICGA'89), pages 42–50. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1989.

82. Bruno Sareni and Laurent Krähenbühl. Fitness Sharing and Niching Methods Revisited. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 2(3):97–106, September 1998. doi: 10.1109/4235.735432. URL `http://hal.archives-ouvertes.fr/hal-00359799/en/`.
83. Brad L. Miller and Michael J. Shaw. Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization. IlliGAL Report 95010, Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Computer Science, Department of General Engineering, University of Illinois at Urbana-Champaign: Urbana-Champaign, IL, USA, December 1, 1995.
84. David Edward Goldberg, Kalyanmoy Deb, and Jeffrey Horn. Massive Multimodality, Deception, and Genetic Algorithms. In Reinhard Männer and Bernard Manderick, editors, *Proceedings of Parallel Problem Solving from Nature 2 (PPSN II)*, pages 37–48. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands and North-Holland Scientific Publishers Ltd.: Amsterdam, The Netherlands, 1992.
85. Eckart Zitzler and Lothar Thiele. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. TIK-Report 43, Eidgenössische Technische Hochschule (ETH) Zürich, Department of Electrical Engineering, Computer Engineering and Networks Laboratory (TIK): Zürich, Switzerland, May 1998. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/zt1998a.pdf`.
86. Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 101, Eidgenössische Technische Hochschule (ETH) Zürich, Department of Electrical Engineering, Computer Engineering and Networks Laboratory (TIK): Zürich, Switzerland, May 2001. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/zlt2001a.pdf`. Errata added 2001-09-27.
87. Thomas Weise, Alexander Podlich, and Christian Gorldt. Solving Real-World Vehicle Routing Problems with Evolutionary Algorithms. In Raymond Chiong and Sandeep Dhakal, editors, *Natural Intelligence for Scheduling, Planning and Packing Problems*, volume 250 of *Studies in Computational Intelligence*, chapter 2, pages 29–53. Springer-Verlag: Berlin/Heidelberg, 2009. doi: 10.1007/978-3-642-04039-9_2. URL `http://www.it-weise.de/research/publications/WPG2009SRWVRPWEA/WPG2009SRWVRPWEA.pdf`.
88. Thomas Weise, Alexander Podlich, Kai Reinhard, Christian Gorldt, and Kurt Geihs. Evolutionary Freight Transportation Planning. In Mario Giacobini, Penousal Machado, Anthony Brabazon, Jon McCormack, Stefano Cagnoni, Michael O'Neill, Gianni A. Di Caro, Ferrante Neri, Anikó Ekárt, Mike Preuß, Anna Isabel Esparcia-Alcázar, Franz Rothlauf, Muddassar Farooq, Ernesto Tarantino, Andreas Fink, and Shengxiang Yang, editors, *Applications of Evolutionary Computing – Proceedings of EvoWorkshops 2009: EvoCOMNET, EvoENVIRONMENT, EvoFIN, EvoGAMES, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, EvoNUM, EvoSTOC, EvoTRANSLOG (EvoWorkshops'09)*, volume 5484/2009 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 768–777. Springer-Verlag GmbH: Berlin, Germany, 2009. doi: 10.1007/978-3-642-01129-0_87. URL `http://www.it-weise.de/research/publications/WPRGG2009EFTP/WPRGG2009EFTP.pdf`.
89. Alan Pétrowski. A Clearing Procedure as a Niching Method for Genetic Algorithms. In Keisoku Jidō and Seigyo Gakkai, editors, *Proceedings of IEEE International Conference on Evolutionary Computation (CEC'96)*, pages 798–803. IEEE Computer Society Press: Los Alamitos, CA, USA, 1996. doi: 10.1109/ICEC.1996.542703. URL `http://sci2s.ugr.es/docencia/cursoMieres/clearing-96.pdf`.
90. Alan Pétrowski. An Efficient Hierarchical Clustering Technique for Speciation. Technical report, Institut National des Télécommunications: Evry Cedex, France, 1997.
91. Paul J. Darwen and Xīn Yáo. Every Niching Method has its Niche: Fitness Sharing and Implicit Sharing Compared. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, volume 1141/1996 of *Lecture Notes in Computer Science (LNCS)*, pages 398–407. Springer-Verlag GmbH: Berlin, Germany, 1996. doi: 10.1007/3-540-61723-X_1004. URL `sclab.yonsei.ac.kr/courses/03EC/darwen96every.pdf`.
92. Thomas Weise, Stefan Niemczyk, Raymond Chiong, and Míngxù Wàn. A Framework for Multi-Model EDAs with Model Recombination.

In *Proceedings of the 4th European Event on Bio-Inspired Algorithms for Continuous Parameter Optimisation (EvoNUM'11), Applications of Evolutionary Computation – Proceedings of EvoApplications 2011: EvoCOMPLEX, EvoGAMES, EvoIASP, EvoINTELLIGENCE, EvoNUM, and EvoSTOC, Part 1 (EvoAPPLICATIONS'11)*, volume 6624 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 304–313. Springer-Verlag GmbH: Berlin, Germany, 2011. doi: 10.1007/978-3-642-20525-5_31. URL http://www.it-weise.de/research/publications/WNCW2011AFFMMEWMR/WNCW2011AFFMMEWMR.pdf.

93. Stefan Niemczyk and Thomas Weise. A General Framework for Multi-Model Estimation of Distribution Algorithms. Technical report, University of Kassel, Fachbereich 16: Elektrotechnik/Informatik, Distributed Systems Group: Kassel, Hesse, Germany, March 10, 2010. URL http://www.it-weise.de/research/publications/NW2010AGFFMMEODA/NW2010AGFFMMEODA.pdf.

94. David Wallin and Conor Ryan. Maintaining Diversity in EDAs for Real-Valued Optimisation Problems. In *Frontiers in the Convergence of Bioscience and Information Technologies (FBIT'07)*, pages 795–800. IEEE (Institute of Electrical and Electronics Engineers): Piscataway, NJ, USA, 2007. doi: 10.1109/FBIT.2007.132.

95. Teresa Miquélez, Endika Bengoetxea, and Pedro Larrañaga. Evolutionary Computation based on Bayesian Classifiers. *International Journal of Applied Mathematics and Computer Science (AMCS)*, 14(3):335–349, September 2004. URL http://matwbn.icm.edu.pl/ksiazki/amc/amc14/amc1434.pdf.

96. Quiang Lu and Xīn Yáo. Clustering and Learning Gaussian Distribution for Continuous Optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 35(2):195–204, May 2005. doi: 10.1109/TSMCC.2004.841914. URL http://www.cs.bham.ac.uk/~xin/papers/published_IEEETSMC_LuYa05.pdf.

97. Michaël Defoin Platel, Stefan Schliebs, and Nikola Kasabov. Quantum-Inspired Evolutionary Algorithm: A Multimodel EDA. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 13(6):1218–1232, December 2009. doi: 10.1109/TEVC.2008.2003010.

98. Marcus Gallagher, Marcus R. Frean, and Tom Downs. Real-Valued Evolutionary Optimization using a Flexible Probability Density Estimator. In Wolfgang Banzhaf, Jason M. Daida, Ágoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark J. Jakiela, and Robert Elliott Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, pages 840–846. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999.

99. Shumeet Baluja. Population-Based Incremental Learning – A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Carnegy Mellon University (CMU), School of Computer Science, Computer Science Department: Pittsburgh, PA, USA, June 2, 1994. URL http://www.ri.cmu.edu/pub_files/pub1/baluja_shumeet_1994_2/baluja_shumeet_1994_2.pdf.

100. Chang Wook Ahn and Rudrapatna S. Ramakrishna. Clustering-Based Probabilistic Model Fitting in Estimation of Distribution Algorithms. *IEICE Transactions on Information and Systems, Oxford Journals*, E89-D(1):381–383, January 2006. doi: 10.1093/ietisy/e89-d.1.381.

101. Aizeng Cao, Yueting Chen, Jun Wei, and Jinping Li. A Hybrid Evolutionary Algorithm Based on EDAs and Clustering Analysis. In Dàizhǎn Chéng and Mǐn Wú, editors, *Proceedings of the 26th Chinese Control Conference (CCC'07)*, pages 754–758. IEEE (Institute of Electrical and Electronics Engineers): Piscataway, NJ, USA, 2007. doi: 10.1109/CHICC.2006.4347236.

102. Tatsuya Okabe, Yàochū Jīn, Bernhard Sendhoff, and Markus Olhofer. Voronoi-based Estimation of Distribution Algorithm for Multi-objective Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'04)*, volume 2, pages 1594–1601. IEEE Computer Society Press: Los Alamitos, CA, USA, 2004. doi: 10.1109/CEC.2004.1331086. URL http://www.honda-ri.org/intern/hriliterature/PN\%2008-04.

103. Kumara Sastry and David Edward Goldberg. Multiobjective hBOA, Clustering, and Scalability. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 663–670. ACM Press: New York, NY, USA, 2005. doi: 10.1145/1068009.1068122.

104. Günter Rudolph. Self-Adaptation and Global Convergence: A Counter-Example. In Peter John Angeline, Zbigniew Michalewicz, Marc Schoenauer, Xīn Yáo, and Ali M. S. Zalzala, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'99)*, volume 1 and 1-3, pages 646–651. IEEE Computer Society: Piscataway, NJ, USA, 1999. doi: 10.1109/CEC.1999.781994. URL `http://ls11-www.cs.uni-dortmund.de/people/rudolph/publications/papers/CEC99.pdf`.

105. Günter Rudolph. Self-Adaptive Mutations may lead to Premature Convergence. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 5(4):410–414, 2001. doi: 10.1109/4235.942534.

106. Darrell F. Lochtefeld and Frank William Ciarallo. Helper-Objective Optimization Strategies for the Job-Shop Scheduling Problem. *Applied Soft Computing*, 11(6):4161–4174, September 2011. doi: 10.1016/j.asoc.2011.03.007.

107. Joshua D. Knowles, Richard A. Watson, and David Wolfe Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos Artemio Coello Coello, and David Wolfe Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*, volume 1993/2001 of *Lecture Notes in Computer Science (LNCS)*, pages 269–283. Springer-Verlag GmbH: Berlin, Germany, 2001. doi: 10.1007/3-540-44719-9_19. URL `http://www.macs.hw.ac.uk/~dwcorne/rlo.pdf`.

108. Mikkel T. Jensen. Guiding Single-Objective Optimization Using Multi-Objective Methods. In Günther R. Raidl, Jean-Arcady Meyer, Martin Middendorf, Stefano Cagnoni, Juan Jesús Romero Cardalda, David Wolfe Corne, Jens Gottlieb, Agnès Guillot, Emma Hart, Colin G. Johnson, and Elena Marchiori, editors, *Applications of Evolutionary Computing, Proceedings of EvoWorkshop 2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, and EvoSTIM (EvoWorkshop'03)*, volume 2611/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 91–98. Springer-Verlag GmbH: Berlin, Germany, 2003. doi: 10.1007/3-540-36605-9_25.

109. Mikkel T. Jensen. Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation. *Journal of Mathematical Modelling and Algorithms*, 3(4):323–347, December 2004. doi: 10.1023/B:JMMA.0000049378.57591.c6.

110. Frank Neumann and Ingo Wegener. Can Single-Objective Optimization Profit from Multiobjective Optimization? In *Multiobjective Problem Solving from Nature – From Concepts to Applications*, Natural Computing Series, pages 115–130. Springer New York: New York, NY, USA, 2008. doi: 10.1007/978-3-540-72964-8_6.

111. Martin Jähne, Xiǎodōng Lǐ, and Jürgen Branke. Evolutionary Algorithms and Multi-Objectivization for the Travelling Salesman Problem. In Franz Rothlauf, Günther R. Raidl, Anna Isabel Esparcia-Alcázar, Ying-Ping Chen, Gabriela Ochoa, Ender Ozcan, Marc Schoenauer, Anne Auger, Hans-Georg Beyer, Nikolaus Hansen, Steffen Finck, Raymond Ros, L. Darrell Whitley, Garnett Wilson, Simon Harding, William B. Langdon, Man Leung Wong, Laurence D. Merkle, Frank W. Moore, Sevan G. Ficici, William Rand, Rick L. Riolo, Nawwaf Kharma, William R. Buckley, Julian Francis Miller, Kenneth Owen Stanley, Jaume Bacardit i Peñarroya, Will N. Browne, Jan Drugowitsch, Nicola Beume, Mike Preuß, Stephen Frederick Smith, Stefano Cagnoni, Alexandru Floares, Aaron Baughman, Steven Matt Gustafson, Maarten Keijzer, Arthur Kordon, and Clare Bates Congdon, editors, *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*, pages 595–602. Association for Computing Machinery (ACM): New York, NY, USA, 2009. doi: 10.1145/1569901.1569984. URL `http://goanna.cs.rmit.edu.au/~xiaodong/publications/multi-objectivization-jahne-gecco09.pdf`.

112. Darrell F. Lochtefeld and Frank William Ciarallo. Deterministic Helper Objective Sequence Applied to the Job-Shop Scheduling Problem. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'10)*, pages 431–438. ACM Press: New York, NY, USA, 2010. doi: 10.1145/1830483.1830566.

113. Darrell F. Lochtefeld and Frank William Ciarallo. Multiobjectivization via Helper-Objectives with the Tunable Objectives Problem. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 16(3):373–395, June 2012. doi: 10.1109/TEVC.2011.2136345.

114. Dimo Brockhoff, Tobias Friedrich, Nils Hebbinghaus, Christian Klein, Frank Neumann, and Eckart Zitzler. Do additional objectives make a problem harder? In Dirk Thierens, Hans-Georg Beyer, Josh C. Bongard, Jürgen Branke, John Andrew Clark, Dave Cliff, Clare Bates Con-

gdon, Kalyanmoy Deb, Benjamin Doerr, Tim Kovacs, Sanjeev P. Kumar, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Riccardo Poli, Kumara Sastry, Kenneth Owen Stanley, Thomas Stützle, Richard A. Watson, and Ingo Wegener, editors, *Proceedings of 9th Genetic and Evolutionary Computation Conference (GECCO'07)*, pages 765–772. ACM Press: New York, NY, USA, 2007. doi: 10.1145/1276958.1277114.

115. Julia Handl, Simon C. Lovell, and Joshua D. Knowles. Multiobjectivization by Decomposition of Scalar Cost Functions. In *Proceedings of 10th International Conference on Parallel Problem Solving from Nature (PPSN X)*, volume 5199/2008 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 31–40. Springer-Verlag GmbH: Berlin, Germany, 2008. doi: 10.1007/978-3-540-87700-4_4. URL http://dbkgroup.org/handl/ppsn_decomp.pdf.

116. Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* PhD thesis, Technische Universität Berlin: Berlin, Germany, 1971.

117. Ingo Rechenberg. *Evolutionsstrategie '94*, volume 1 of *Werkstatt Bionik und Evolutionstechnik.* Frommann-Holzboog Verlag: Bad Cannstadt, Stuttgart, Baden-Württemberg, Germany, 1994. ISBN ISBN: 3-7728-1642-8.

118. Justinian P. Rosca and Dana H. Ballard. Causality in Genetic Programming. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 256–263. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995.

119. Krasimir Kolarov. Landscape Ruggedness in Evolutionary Algorithms. In Thomas Bäck, Zbigniew Michalewicz, and Xīn Yáo, editors, *IEEE International Conference on Evolutionary Computation (CEC'97)*, pages 19–24. IEEE Computer Society: Piscataway, NJ, USA, 1997. doi: 10.1109/ICEC.1997.592261.

120. Jean-Baptiste Pierre Antoine de Monet, Chevalier de Lamarck. *Philosophie zoologique – ou Exposition des considérations relatives à l'histoire naturelle des Animaux; à la diversité de leur organisation et des facultés qu'ils en obtiennent; ....* Dentu: Paris, France and J. B. Baillière Liberaire: Paris, France, 1809. ISBN ISBN: 1412116465. URL http://www.lamarck.cnrs.fr/ice/ice_book_detail.php?type=text&bdd=lamarck&table=ouvrages_lamarck&bookId=29. Philosophie zoologique – ou Exposition des considérations relatives à l'histoire naturelle des Animaux; à la diversité de leur organisation et des facultés qu'ils en obtiennent; aux causes physiques qui maintiennent en eux la vie et donnent lieu aux mouvements qu'ils exécutant; enfin, à celles qui produisent les unes le sentiment, et les autres l'intelligence de ceux qui en sont doués.

121. L. Darrell Whitley, V. Scott Gordon, and Keith E. Mathias. Lamarckian Evolution, The Baldwin Effect and Function Optimization. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 5–15. Springer-Verlag GmbH: Berlin, Germany, 1994. doi: 10.1007/3-540-58484-6_245.

122. James Mark Baldwin. A New Factor in Evolution. *The American Naturalist*, 30(354):441–451, June 1896. URL http://www.brocku.ca/MeadProject/Baldwin/Baldwin_1896_h.html.

123. Geoffrey E. Hinton and Steven J. Nowlan. How Learning Can Guide Evolution. *Complex Systems*, 1(3):495–502, 1987. URL http://htpprints.yorku.ca/archive/00000172/.

124. Geoffrey E. Hinton and Steven J. Nowlan. How Learning Can Guide Evolution. In Richard K. Belew and Melanie Mitchell, editors, *Adaptive Individuals in Evolving Populations: Models and Algorithms*, volume 26 of *Santa Fe Institue Studies in the Sciences of Complexity*, pages 447–454. Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA and Westview Press: Boulder, CO, USA, 1996.

125. Pablo Moscato. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Caltech Concurrent Computation Program C3P 826, California Institute of Technology (Caltech), Caltech Concurrent Computation Program (C3P): Pasadena, CA, USA, 1989. URL http://www.each.usp.br/sarajane/SubPaginas/arquivos_aulas_IA/memetic.pdf.

126. Michael G. Norman and Pablo Moscato. A Competitive and Cooperative Approach to Complex Combinatorial Search. In *Proceedings of the 20th Informatics and Operations Research Meeting (20th Jornadas Argentinas e Informática e Investigación Operativa) (JAIIO'91)*, pages 3.15–3.29, 1991. Also published as Technical Re-

port Caltech Concurrent Computation Program, Report. 790, California Institute of Technology, Pasadena, California, USA, 1989.

127. Diana Holstein and Pablo Moscato. Memetic Algorithms using Guided Local Search: A Case Study. In David Wolfe Corne, Marco Dorigo, Fred Glover, Dipankar Dasgupta, Pablo Moscato, Riccardo Poli, and Kenneth V. Price, editors, *New Ideas in Optimization*, McGraw-Hill's Advanced Topics In Computer Science Series, pages 235–244. McGraw-Hill Ltd.: Maidenhead, UK, England, 1999.

128. Luciana Buriol, Paulo M. França, and Pablo Moscato. A New Memetic Algorithm for the Asymmetric Traveling Salesman Problem. *Journal of Heuristics*, 10(5):483–506, September 2004. doi: 10.1023/B:HEUR.0000045321.59202.52. URL `http://citeseer.ist.psu.edu/544761.html`.

129. Maria J. Blesa, Pablo Moscato, and Fatos Xhafa. A Memetic Algorithm for the Minimum Weighted k-Cardinality Tree Subgraph Problem. In Mauricio G.C. Resende, Jorge Pinho de Sousa, and Ana Viana, editors, *4th Metaheuristics International Conference – Metaheuristics: Computer Decision-Making (MIC'01)*, volume 86 of *Applied Optimization*, pages 85–90. Springer-Verlag GmbH: Berlin, Germany, 2001.

130. Pablo Moscato and Carlos Cotta. A Gentle Introduction to Memetic Algorithms. In Fred Glover and Gary A. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, chapter 5, pages 105–144. Kluwer Academic Publishers: Norwell, MA, USA and Springer Netherlands: Dordrecht, Netherlands, 2003. doi: 10.1007/0-306-48056-5_5. URL `http://www.lcc.uma.es/~ccottap/papers/handbook03memetic.pdf`.

131. Nicholas J. Radcliffe and Patrick David Surry. Formal Memetic Algorithms. In Terence Claus Fogarty, editor, *Proceedings of the Workshop on Artificial Intelligence and Simulation of Behaviour, International Workshop on Evolutionary Computing, Selected Papers (AISB'94)*, volume 865/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 1–16. Society for the Study of Artificial Intelligence and the Simulation of Behaviour (SSAISB): Chichester, West Sussex, UK, Springer-Verlag GmbH: Berlin, Germany, 1994. doi: 10.1007/3-540-58483-8_1.

132. Jason Digalakis and Konstantinos Margaritis. A Parallel Memetic Algorithm for Solving Optimization Problems. In Mauricio G.C. Resende, Jorge Pinho de Sousa, and Ana Viana, editors, *4th Metaheuristics International Conference – Metaheuristics: Computer Decision-Making (MIC'01)*, volume 86 of *Applied Optimization*, pages 121–125. Springer-Verlag GmbH: Berlin, Germany, 2001. URL `http://citeseer.ist.psu.edu/digalakis01parallel.html`.

133. Jason Digalakis and Konstantinos Margaritis. Performance Comparison of Memetic Algorithms. *Journal of Applied Mathematics and Computation*, 158:237–252, October 2004. doi: 10.1016/j.amc.2003.08.115. URL `http://www.complexity.org.au/ci/draft/draft/digala02/digala02s.pdf`.

134. Natalio Krasnogor and James E. Smith. A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 9(5):474–488, October 2005. doi: 10.1109/TEVC.2005.850260. URL `http://www.cs.nott.ac.uk/~nxk/PAPERS/IEEE-TEC-lastVersion.pdf`.

135. David H. Ackley. *A Connectionist Machine for Genetic Hillclimbing*. PhD thesis, Carnegy Mellon University (CMU): Pittsburgh, PA, USA, August 31, 1987.

136. Martina Gorges-Schleuter. ASPARAGOS: An Asynchronous Parallel Genetic Optimization Strategy. In James David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pages 422–427. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1989.

137. Heinz Mühlenbein. Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization. In Jörg D. Becker, Ignaz Eisele, and F. W. Mündemann, editors, *Parallelism, Learning, Evolution: Workshop on Evolutionary Models and Strategies (Neubiberg, Germany, 1989-03-10/11) and Workshop on Parallel Processing: Logic, Organization, and Technology (Wildbad Kreuth, Germany, 1989-07-24 to 28) (WOPPLOT'89)*, volume 565/1991 of *Lecture Notes in Artificial Intelligence (LNAI, SL 7), Lecture Notes in Computer Science (LNCS)*, pages 398–406. Springer-Verlag GmbH: Berlin, Germany, 1989. doi: 10.1007/3-540-55027-5_23.

138. Heinz Mühlenbein. Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization. In James David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pages 416–421. Morgan Kaufmann Publishers Inc.: San Fran-

cisco, CA, USA, 1989.

139. Heinz Mühlenbein. How Genetic Algorithms Really Work – I. Mutation and Hillclimbing. In Reinhard Männer and Bernard Manderick, editors, *Proceedings of Parallel Problem Solving from Nature 2 (PPSN II)*, pages 15–26. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands and North-Holland Scientific Publishers Ltd.: Amsterdam, The Netherlands, 1992. URL `http://muehlenbein.org/mut92.pdf`.

140. Donald E. Brown, Christopher L. Huntley, and Andrew R. Spillane. A Parallel Genetic Heuristic for the Quadratic Assignment Problem. In James David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms (ICGA'89)*, pages 406–415. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1989.

141. Lawrence Davis, editor. *Handbook of Genetic Algorithms*. VNR Computer Library. Thomson Publishing Group, Inc.: Stamford, CT, USA, January 1991. ISBN ISBN: 0-442-00173-8 and ISBN: 1850328250.

142. Frédéric Gruau and L. Darrell Whitley. Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect. *Evolutionary Computation*, 1(3):213–233, 1993. doi: 10.1162/evco.1993.1.3.213.

143. Ko-Hsin Liang, Xīn Yáo, and Charles S. Newton. Evolutionary Search of Approximated N-Dimensional Landscapes. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 4(3):172–183, June 2000. URL `http://sclab.yonsei.ac.kr/courses/03EC/liang00evolutionary.pdf`.

144. Yú Wáng, Bīn Lǐ, and Thomas Weise. Estimation of Distribution and Differential Evolution Cooperation for Large Scale Economic Load Dispatch Optimization of Power Systems. *Information Sciences – Informatics and Computer Science Intelligent Systems Applications: An International Journal*, 180(12):2405–2420, June 2010. doi: 10.1016/j.ins.2010.02.015. URL `http://www.it-weise.de/research/publications/WLW2010EODADECFLSELDOOPS/WLW2010EODADECFLSELDOOPS.pdf`.

145. Thomas Weise and Kē Táng. Evolving Distributed Algorithms with Genetic Programming. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 16(2):242–265, April 2012. doi: 10.1109/TEVC.2011.2112666.

146. Thomas Weise. *Evolving Distributed Algorithms with Genetic Programming*. PhD thesis, University of Kassel, Fachbereich 16: Elektrotechnik/Informatik, Distributed Systems Group: Kassel, Hesse, Germany, May 4, 2009. URL `http://www.it-weise.de/research/publications/W2009EDAWGP/W2009DISS.pdf`. Won the Dissertation Award of The Association of German Engineers (Verein Deutscher Ingenieure, VDI).

147. Albert Donally Bethke. *Genetic Algorithms as Function Optimizers*. PhD thesis, University of Michigan: Ann Arbor, MI, USA, 1980. National Science Foundation Grant No. MCS76-04297.

148. David Edward Goldberg. Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction. *Complex Systems*, 3(2):129–152, 1989. URL `http://www.complex-systems.com/pdf/03-2-2.pdf`.

149. David Edward Goldberg. Genetic Algorithms and Walsh Functions: Part II, Deception and Its Analysis. *Complex Systems*, 3(2):153–171, 1989. URL `http://www.complex-systems.com/pdf/03-2-3.pdf`.

150. Gunar E. Liepins and Michael D. Vose. Deceptiveness and Genetic Algorithm Dynamics. In Bruce M. Spatz and Gregory J. E. Rawlins, editors, *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)*, pages 36–50. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990. URL `http://www.osti.gov/bridge/servlets/purl/6445602-CfqU6M/`.

151. Thomas Weise, Stefan Niemczyk, Hendrik Skubch, Roland Reichle, and Kurt Geihs. A Tunable Model for Multi-Objective, Epistatic, Rugged, and Neutral Fitness Landscapes. In Maarten Keijzer, Giuliano Antoniol, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Nikolaus Hansen, John H. Holmes, Gregory S. Hornby, Daniel Howard, James Kennedy, Sanjeev P. Kumar, Fernando G. Lobo, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Jordan B. Pollack, Kumara Sastry, Kenneth Owen Stanley, Adrian Stoica, El-Ghazali Talbi, and Ingo Wegener, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'08)*, pages 795–802. ACM Press: New York, NY, USA, 2008. doi: 10.1145/1389095.1389252. URL `http://www.it-weise.de/research/publications/WNSRG2008ATMFMOERANFL/WNSRG2008ATMFMOERANFL.pdf`.

152. Thorsten Schnier and Xīn Yáo. Using Multiple Representations in Evolutionary Algorithms.

In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'00)*, volume 1, pages 479–486. IEEE Computer Society: Piscataway, NJ, USA, 2000. doi: 10.1109/CEC.2000.870335. URL `http://www.cs.bham.ac.uk/~axk/multiple.pdf`.

153. Xiǎodōng Lǐ. Niching without Niching Parameters: Particle Swarm Optimization Using a Ring Topology. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 14(1):150–169, February 2010. doi: 10.1109/TEVC.2009.2026270. URL `http://goanna.cs.rmit.edu.au/~xiaodong/publications/rpso-ieee-tec.pdf`.

154. Shane Legg, Marcus Hutter, and Akshat Kumar. Tournament versus Fitness Uniform Selection. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'04)*, pages 2144–2151. IEEE Computer Society Press: Los Alamitos, CA, USA, 2004. doi: 10.1109/CEC.2004.1331162. URL `http://www.hutter1.net/ai/fussexp.pdf`.

155. Marcus Hutter. Fitness Uniform Selection to Preserve Genetic Diversity. In David B. Fogel, Mohamed A. El-Sharkawi, Xīn Yáo, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1-2, pages 783–788. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society Press: Los Alamitos, CA, USA, 2002. doi: 10.1109/CEC.2002.1007025. URL `ftp://ftp.idsia.ch/pub/techrep/IDSIA-01-01.ps.gz`. Also: Technical Report IDSIA-01-01, 17 January 2001.

156. Marcus Hutter and Shane Legg. Fitness Uniform Optimization. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 10 (5):568–589, October 2006. doi: 10.1109/TEVC.2005.863127. URL `http://www.idsia.ch/idsiareport/IDSIA-16-06.pdf`.

157. Shane Legg and Marcus Hutter. Fitness Uniform Deletion: A Simple Way to Preserve Diversity. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 1271–1278. ACM Press: New York, NY, USA, 2005. doi: 10.1145/1068009.1068216. URL `http://www.hutter1.net/ai/fuds.pdf`.

158. Joel Lehman and Kenneth Owen Stanley. Exploiting Open-Endedness to Solve Problems Through the Search for Novelty. In Seth Bullock, Jason Noble, Richard A. Watson, and Mark A. Bedau, editors, *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems (Artificial Life XI)*, pages 329–336. MIT Press: Cambridge, MA, USA, 2008. URL `http://alifexi.alife.org/papers/ALIFExi_pp329-336.pdf`.

159. Joel Lehman and Kenneth Owen Stanley. Abandoning Objectives: Evolution Through the Search for Novelty Alone. *Evolutionary Computation*, 19(2):189–223, 2011. doi: 10.1162/EVCO_a_00025. URL `http://eplex.cs.ucf.edu/papers/lehman_ecj10.pdf`.

160. Joel Lehman and Kenneth Owen Stanley. Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'11)*, pages 211–218, 2011. doi: 10.1145/2001576.2001606. URL `http://eplex.cs.ucf.edu/papers/lehman_gecco11.pdf`.

161. Míngxù Wàn, Thomas Weise, and Kē Táng. Novel Loop Structures and the Evolution of Mathematical Algorithms. In Sara Silva, James A. Foster, Miguel Nicolau, Penousal Machado, and Mario Giacobini, editors, *Proceedings of the 14th European Conference on Genetic Programming (EuroGP'11)*, volume 6621/2011 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 49–60. Springer-Verlag GmbH: Berlin, Germany, 2011. doi: 10.1007/978-3-642-20407-4_5. URL `http://www.it-weise.de/research/publications/WWT2011NLSATEOMA/WWT2011NLSATEOMA.pdf`.

162. Christian M. Reidys and Peter F. Stadler. Neutrality in Fitness Landscapes. *Journal of Applied Mathematics and Computation*, 117(2–3): 321–350, January 25, 2001. doi: 10.1016/S0096-3003(99)00166-6.

163. Lionel Barnett. Ruggedness and Neutrality – The NKp Family of Fitness Landscapes. In Christoph Adami, Richard K. Belew, Hiroaki Kitano, and Charles E. Taylor, editors, *Proceedings of the Sixth International Conference on Artificial Life (Artificial Life VI)*, volume 6 of *Complex Adaptive Systems, Bradford Books*, pages 18–27. MIT Press: Cambridge, MA, USA, 1998. URL `http://www.cogs.susx.ac.uk/users/lionelb/`

downloads/publications/alife6_paper.ps.gz.

164. Ting Hu and Wolfgang Banzhaf. Evolvability and Speed of Evolutionary Algorithms in the Light of Recent Developments in Biology. *Journal of Artificial Evolution and Applications*, 2010. doi: 10.1155/2010/568375. URL http://www.cs.mun.ca/~tingh/papers/JAEA10.pdf.

165. Richard Dawkins. The Evolution of Evolvability. In Christopher Gale Langdon, editor, *The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems (Artificial Life'87)*, volume 6 of *Santa Fe Institue Studies in the Sciences of Complexity*, pages 201–220. Addison-Wesley Longman Publishing Co., Inc.: Boston, MA, USA and Westview Press: Boulder, CO, USA, 1987.

166. Andreas Wagner. *Robustness and Evolvability in Living Systems*, volume 9 of *Princeton Studies in Complexity*. Princeton University Press: Princeton, NJ, USA, 2005. ISBN ISBN: 0691122407 and ISBN: 0691134049.

167. Andreas Wagner. Robustness, Evolvability, and Neutrality. *FEBS Letters*, 579(8):1772–1778, March 21, 2005. doi: 10.1016/j.febslet.2005.01.063.

168. Marc Kirschner and John Gerhart. Evolvability. *Proceedings of the National Academy of Science of the United States of America (PNAS)*, 95 (15):8420–8427, July 15, 1998. URL http://www.pnas.org/cgi/content/full/95/15/8420.

169. Hans-Georg Beyer. Toward a Theory of Evolution Strategies: The (μ, λ)-Theory. *Evolutionary Computation*, 2(4):381–407, 1994. doi: 10.1162/evco.1994.2.4.381. URL http://ls11-www.cs.uni-dortmund.de/people/beyer/coll/Bey95a/Bey95a.ps.

170. Lee Altenberg. Fitness Distance Correlation Analysis: An Instructive Counterexample. In Thomas Bäck, editor, *Proceedings of The Seventh International Conference on Genetic Algorithms (ICGA'97)*, pages 57–64. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1997. URL http://dynamics.org/Altenberg/PAPERS/FDCAAIC/.

171. Lee Altenberg. The Schema Theorem and Price's Theorem. In L. Darrell Whitley and Michael D. Vose, editors, *Proceedings of the Third Workshop on Foundations of Genetic Algorithms (FOGA 3)*, pages 23–49. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1994. URL http://citeseer.ist.psu.edu/old/700230.html.

172. Gwoing Tina Yu. Program Evolvability Under Environmental Variations and Neutrality. In Fernando Almeida e Costa, Luís Mateus Rocha, Ernesto Jorge Fernandes Costa, Inman Harvey, and António Coutinho, editors, *Proceedings of the 9th European Conference on Advances in Artificial Life (ECAL'07)*, volume 4648/2007 of *Lecture Notes in Artificial Intelligence (LNAI, SL 7), Lecture Notes in Computer Science (LNCS)*, pages 835–844. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-74913-4_84.

173. William Beaudoin, Sébastien Vérel, Philippe Collard, and Cathy Escazut. Deceptiveness and Neutrality the ND Family of Fitness Landscapes. In Maarten Keijzer and Mike Cattolico, editors, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pages 507–514. ACM Press: New York, NY, USA, 2006. doi: 10.1145/1143997.1144091.

174. Motoo Kimura. *The Neutral Theory of Molecular Evolution*. Cambridge University Press: Cambridge, UK, January 1985. ISBN ISBN: 0521317932. reprint edition, 1985, February.

175. Christian V. Forst, Christian M. Reidys, and Jacqueline Weber. Evolutionary Dynamics and Optimization: Neutral Networks as Model-Landscapes for RNA Secondary-Structure Folding-Landscapes. In Federico Morán, Alvaro Moreno, Juan Julián Merelo-Guervós, and Pablo Chacón, editors, *Advances in Artificial Life – Proceedings of the Third European Conference on Artificial Life (ECAL'95)*, volume 929/1995 of *Lecture Notes in Artificial Intelligence (LNAI, SL 7), Lecture Notes in Computer Science (LNCS)*, pages 128–147. Springer-Verlag GmbH: Berlin, Germany, 1995. doi: 10.1007/3-540-59496-5_294.

176. Martijn A. Huynen, Peter F. Stadler, and Walter Fontana. Smoothness within Ruggedness: The Role of Neutrality in Adaptation. *Proceedings of the National Academy of Science of the United States of America (PNAS)*, 93(1):397–401, January 9, 1996. URL http://fontana.med.harvard.edu/www/Documents/WF/Papers/neutrality.pdf. Communicated by Hans Frauenfelder, Los Alamos National Laboratory, Los Alamos, NM, September 20, 1995 (received for review June 29, 1995).

177. Marc Toussaint and Christian Igel. Neutrality: A Necessity for Self-Adaptation. In David B. Fogel, Mohamed A. El-Sharkawi, Xīn Yáo, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress*

on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02), volume 1-2, pages 1354–1359. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society Press: Los Alamitos, CA, USA, 2002. doi: 10.1109/CEC.2002.1004440.

178. Stephen Jay Gould and Niles Eldredge. Punctuated Equilibrium Comes of Age. *Nature*, 366 (6452):223–227, November 18, 1993. doi: 10.1038/366223a0. URL `http://evolucion.fcien.edu.uy/Lecturas/GouldyEldredge1993.pdf`.

179. Niles Eldredge and Stephen Jay Gould. Punctuated Equilibria: An Alternative to Phyletic Gradualism. In Thomas J. M. Schopf, editor, *Models in Paleobiology*, chapter 5, pages 82–115. Freeman, Cooper & Co: San Francisco, CA, USA and DoubleDay: New York, NY, USA, 1972. URL `http://www.blackwellpublishing.com/ridley/classictexts/eldredge.pdf`.

180. Niles Eldredge and Stephen Jay Gould. Punctuated Equilibria: The Tempo and Mode of Evolution Reconsidered. *Paleobiology*, 3(2): 115–151, April 1, 1977. URL `http://www.nileseldredge.com/pdf_files/Punctuated_Equilibria_Gould_Eldredge_1977.pdf`.

181. Per Bak and Kim Sneppen. Punctuated Equilibrium and Criticality in a Simple Model of Evolution. *Physical Review Letters*, 71(24): 4083–4086, December 13, 1993. doi: 10.1103/PhysRevLett.71.4083. URL `http://prola.aps.org/abstract/PRL/v71/i24/p4083_1`.

182. Erik van Nimwegen, James P. Crutchfield, and Melanie Mitchell. Statistical Dynamics of the Royal Road Genetic Algorithm. *Theoretical Computer Science*, 229(1–2):41–102, November 6, 1999. doi: 10.1016/S0304-3975(99)00119-X.

183. James P. Cohoon, S. U. Hegde, Worthy Neil Martin, and D. Richards. Punctuated Equilibria: A Parallel Genetic Algorithm. In John J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms and their Applications (ICGA'87)*, pages 148–154. Lawrence Erlbaum Associates, Inc. (LEA): Mahwah, NJ, USA, 1987.

184. Worthy Neil Martin, Jens Lienig, and James P. Cohoon. Island (Migration) Models: Evolutionary Algorithms Based on Punctuated Equilibria. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, Computational Intelligence Library, chapter C6.3, pages 448–463. Oxford University Press, Inc.: New York, NY, USA, Institute of Physics Publishing Ltd. (IOP): Dirac House, Temple Back, Bristol, UK, and CRC Press, Inc.: Boca Raton, FL, USA, 1997.

185. Gerald M. Edelman and Joseph A. Gally. Degeneracy and Complexity in Biological Systems. *Proceedings of the National Academy of Science of the United States of America (PNAS)*, 98(24): 13763–13768, November 20, 2001. doi: 10.1073/pnas.231499798.

186. James M. Whitacre, Philipp Rohlfshagen, Axel Bender, and Xīn Yáo. The Role of Degenerate Robustness in the Evolvability of Multi-Agent Systems in Dynamic Environments. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Proceedings of the 11th International Conference on Parallel Problem Solving From Nature, Part 1 (PPSN XI-1)*, volume 6238 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 284–293. Springer-Verlag GmbH: Berlin, Germany, 2010. doi: 10.1007/978-3-642-15844-5_29.

187. Tom Smith, Phil Husbands, Paul Layzell, and Michael O'Shea. Fitness Landscapes and Evolvability. *Evolutionary Computation*, 10(1):1–34, 2002. doi: 10.1162/106365602317301754. URL `http://wotan.liu.edu/docis/dbl/evocom/2002_10_1_1_FLAE.html`.

188. Mike Preuß, Lutz Schönemann, and Michael T.M. Emmerich. Counteracting Genetic Drift and Disruptive Recombination in $(\mu \overset{+}{,} \lambda)$-EA on Multimodal Fitness Landscapes. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantú-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llorà, Spiros Mancoridis, Martin Pelikan, Günther R. Raidl, Terence Soule, Jean-Paul Watson, and Eckart Zitzler, editors, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO'05)*, pages 865–872. ACM Press: New York, NY, USA, 2005. doi: 10.1145/1068009.1068155. URL `http://www.liacs.nl/~oshir/NichingES/Preuss_Emmerich_GECCO05.pdf`.

189. Karsten Weicker and Nicole Weicker. Burden and Benefits of Redundancy. In *Proceedings of the Sixth Workshop on Foundations of Genetic Algorithms (FOGA'00)*, pages 313–333. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2001. URL `http://www.imn.htwk-leipzig.de/~weicker/publications/redundancy.pdf`.

190. Rob Shipman, Mark Shackleton, Marc Ebner, and Richard A. Watson. Neutral Search Spaces

for Artificial Evolution: A Lesson from Life. In Mark A. Bedau, John S. McCaskill, Norman H. Packard, and Steen Rasmussen, editors, *Proceedings of the Seventh International Conference on Artificial Life (Artificial Life VII)*, Complex Adaptive Systems, Bradford Books, pages 162–167. MIT Press: Cambridge, MA, USA, 2000. URL `http://homepages.feis.herts.ac.uk/~nehaniv/al7ev/shipmanUH.ps`.

191. Mark Shackleton, Rob Shipman, and Marc Ebner. An Investigation of Redundant Genotype-Phenotype Mappings and their Role in Evolutionary Search. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'00)*, volume 1, pages 493–500. IEEE Computer Society: Piscataway, NJ, USA, 2000. doi: 10.1109/CEC.2000.870337. URL `http://citeseer.ist.psu.edu/old/409243.html`.

192. Stuart Alan Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Inc.: New York, NY, USA, May 1993. ISBN ISBN: 0195058119 and ISBN: 0195079515.

193. Rob Shipman, Mark Shackleton, and Inman Harvey. The Use of Neutral Genotype-Phenotype Mappings for Improved Evolutionary Search. *BT Technology Journal*, 18(4):103–111, October 2000. doi: 10.1023/A:1026714927227.

194. Rob Shipman. Genetic Redundancy: Desirable or Problematic for Evolutionary Adaptation? In Andrej Dobnikar, Nigel C. Steele, David W. Pearson, and Rudolf F. Albrecht, editors, *Proceedings of the 4th International Conference on Artificial Neural Nets and Genetic Algorithms (ICANNGA'99)*, pages 1–11. Birkhäuser Verlag: Basel, Switzerland and Springer-Verlag GmbH: Berlin, Germany, 1999.

195. Vesselin K. Vassilev and Julian Francis Miller. The Advantages of Landscape Neutrality in Digital Circuit Evolution. In Julian Francis Miller, Adrian Thompson, Peter Thomson, and Terence Claus Fogarty, editors, *Proceedings of Third International Conference on Evolvable Systems – From Biology to Hardware (ICES'00)*, volume 1801/2000 of *Lecture Notes in Computer Science (LNCS)*, pages 252–263. Springer-Verlag GmbH: Berlin, Germany, 2000. doi: 10.1007/3-540-46406-9_25.

196. Gwoing Tina Yu and Julian Francis Miller. Finding Needles in Haystacks Is Not Hard with Neutrality. In James A. Foster, Evelyne Lutton, Julian Francis Miller, Conor Ryan, and Andrea G. B. Tettamanzi, editors, *Proceedings of the 5th European Conference on Genetic Programming (EuroGP'02)*, volume 2278/2002 of *Lecture Notes in Computer Science (LNCS)*, pages 46–54. Springer-Verlag GmbH: Berlin, Germany, 2002. doi: 10.1007/3-540-45984-7_2. URL `http://www.cs.mun.ca/~tinayu/index_files/addr/public_html/EuroGP2002.pdf`.

197. David Edward Goldberg. Making Genetic Algorithm Fly: A Lesson from the Wright Brothers. *Advanced Technology For Developers*, 2:1–8, February 1993.

198. Christian F. Tschudin. Fraglets – A Metabolistic Execution Model for Communication Protocols. In *The Second Annual Symposium on Autonomous Intelligent Networks and Systems (AINS'03)*. Center for Autonomous Intelligent Networks and Systems (CAINS), University of California (UCLA): Los Angeles, CA, USA, 2003. URL `http://cn.cs.unibas.ch/pub/doc/2003-ains.pdf`.

199. Thomas Weise and Michael Zapf. Evolving Distributed Algorithms with Genetic Programming: Election. In Lihong Xu, Erik D. Goodman, and Yongsheng Ding, editors, *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC'09)*, pages 577–584. ACM Press: New York, NY, USA, 2009. doi: 10.1145/1543834.1543913. URL `http://www.it-weise.de/research/publications/WZ2009EDAWGPE/WZ2009EDAWGPE.pdf`.

200. Simon Ronald. Robust Encodings in Genetic Algorithms: A Survey of Encoding Issues. In Thomas Bäck, Zbigniew Michalewicz, and Xīn Yáo, editors, *IEEE International Conference on Evolutionary Computation (CEC'97)*, pages 43–48. IEEE Computer Society: Piscataway, NJ, USA, 1997. doi: 10.1109/ICEC.1997.592265.

201. Patrick C. Phillips. The Language of Gene Interaction. *Genetics*, 149(3):1167–1171, July 1998. URL `http://www.genetics.org/cgi/reprint/149/3/1167.pdf`.

202. William Bateson. *Mendel's Principles of Heredity*. Kessinger Publishing's® Rare Reprints. Cambridge University Press: Cambridge, UK, 1909. ISBN ISBN: 1428648194.

203. Sir Ronald Aylmer Fisher. The Correlations between Relatives on the Supposition of Mendelian Inheritance. *Philosophical Transactions of the Royal Society of Edinburgh*, 52:399–433, October 1, 1918. URL `http://www.library.adelaide.edu.au/digitised/fisher/9.pdf`.

204. Jay L. Lush. Progeny Test and Individual Performance as Indicators of an Animal's Breeding

Value. *Journal of Dairy Science (JDS)*, 18(1):1–19, January 1935. URL `http://jds.fass.org/cgi/reprint/18/1/1`.

205. Yuval Davidor. Epistasis Variance: A Viewpoint on GA-Hardness. In Bruce M. Spatz and Gregory J. E. Rawlins, editors, *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)*, pages 23–35. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990.
206. Lee Altenberg. NK Fitness Landscapes. In Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors, *Handbook of Evolutionary Computation*, Computational Intelligence Library, chapter B2.7.2. Oxford University Press, Inc.: New York, NY, USA, Institute of Physics Publishing Ltd. (IOP): Dirac House, Temple Back, Bristol, UK, and CRC Press, Inc.: Boca Raton, FL, USA, 1997. URL `http://dynamics.org/Altenberg/FILES/LeeNKFL.pdf`.
207. Bart Naudts and Alain Verschoren. Epistasis on Finite and Infinite Spaces. In *8th International Conference on Systems Research, Informatics and Cybernetics (InterSymp'96)*, pages 19–23. International Institute for Advanced Studies in Systems Research and Cybernetic (IIAS): Tecumseh, ON, Canada, 1996.
208. Colin R. Reeves and Christine C. Wright. Epistasis in Genetic Algorithms: An Experimental Design Perspective. In Larry J. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA'95)*, pages 217–224. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995.
209. Bruce M. Spatz and Gregory J. E. Rawlins, editors. *Proceedings of the First Workshop on Foundations of Genetic Algorithms (FOGA'90)*, July 15–18, 1990. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA. ISBN ISBN: 1-55860-170-8. Published July 1, 1991.
210. Ting Hu and Wolfgang Banzhaf. Evolvability and Acceleration in Evolutionary Computation. Technical Report MUN-CS-2008-04, Memorial University, Department of Computer Science: St. John's, Canada, October 2008. URL `http://www.cs.mun.ca/~tingh/papers/TR08.pdf`.
211. George C. Williams. Pleiotropy, Natural Selection, and the Evolution of Senescence. *Evolution – International Journal of Organic Evolution*, 11(4):398–411, December 1957. doi: 10.2307/2406060.
212. Kē Táng, Xīn Yáo, Ponnuthurai Nagaratnam Suganthan, Cara MacNish, Ying-Ping Chen, Chih-Ming Chen, and Zhènyǔ Yáng. Benchmark Functions for the CEC'2008 Special Session and Competition on Large Scale Global Optimization. Technical report, University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Héféi, Ānhuī, China, 2007. URL `http://nical.ustc.edu.cn/cec08ss.php`.
213. George Hadley. *Nonlinear and Dynamics Programming*. World Student. Addison-Wesley Professional: Reading, MA, USA, December 1964. ISBN ISBN: 0201026643.
214. Kē Táng, Xiǎodōng Lǐ, Ponnuthurai Nagaratnam Suganthan, Zhènyǔ Yáng, and Thomas Weise. Benchmark Functions for the CEC'2010 Special Session and Competition on Large-Scale Global Optimization. Technical report, University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL): Héféi, Ānhuī, China, January 8, 2010. URL `http://www.it-weise.de/research/publications/TLSYW2010BFFTCSSACOLSGO/TLSYW2010BFFTCSSACOLSGO.pdf`.
215. Anne Auger, Nikolaus Hansen, Nikolas Mauny, Raymond Ros, and Marc Schoenauer. Bio-Inspired Continuous Optimization: The Coming of Age. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07)*. IEEE Computer Society: Piscataway, NJ, USA, 2007. URL `http://www.lri.fr/~marc/schoenauerCEC07.pdf`. Invited Talk at CEC'2007.
216. James V. Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, and Anne Auger. PSO Facing Non-Separable and Ill-Conditioned Problems. Technical Report 6447, Institut National de Recherche en Informatique et en Automatique (INRIA), February 11, 2008. URL `http://hal.archives-ouvertes.fr/docs/00/25/01/60/PDF/RR-6447.pdf`.
217. Andrew M. Sutton, Monte Lunacek, and L. Darrell Whitley. Differential Evolution and Non-Separability: Using Selective Pressure to Focus Search. In Dirk Thierens, Hans-Georg Beyer, Josh C. Bongard, Jürgen Branke, John Andrew Clark, Dave Cliff, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Tim Kovacs, Sanjeev P. Kumar, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Riccardo Poli, Kumara Sastry, Kenneth Owen Stanley, Thomas Stützle, Richard A. Watson, and

Ingo Wegener, editors, *Proceedings of 9th Genetic and Evolutionary Computation Conference (GECCO'07)*, pages 1428–1435. ACM Press: New York, NY, USA, 2007. doi: 10.1145/1276958.1277221. URL `http://www.cs.colostate.edu/~sutton/pubs/Sutton2007de.ps.gz`.

218. Chris P. Bowers. Simulating Evolution with a Computational Model of Embryogeny: Obtaining Robustness from Evolved Individuals. In Mathieu S. Capcarrère, Alex Alves Freitas, Peter J. Bentley, Colin G. Johnson, and Jonathan Timmis, editors, *Proceedings of the 8th European Conference on Advances in Artificial Life (ECAL'05)*, volume 3630 of *Lecture Notes in Artificial Intelligence (LNAI, SL 7), Lecture Notes in Computer Science (LNCS)*, pages 149–158. Springer-Verlag GmbH: Berlin, Germany, 2005. URL `http://www.cs.bham.ac.uk/~cpb/publications/ecal05_bowers.pdf`.

219. Thomas Weise, Michael Zapf, and Kurt Geihs. Rule-based Genetic Programming. In *Proceedings of the 2nd International Conference on Bio-Inspired Models of Network, Information, and Computing Systems (BIONETICS'07)*, pages 8–15. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/BIMNICS.2007.4610073. URL `http://www.it-weise.de/research/publications/WZG2007RBGP/WZG2007RBGP.pdf`.

220. Masaya Shinkai, Arturo Hernández Aguirre, and Kiyoshi Tanaka. Mutation Strategy Improves GAs Performance on Epistatic Problems. In David B. Fogel, Mohamed A. El-Sharkawi, Xīn Yáo, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1 and 1-2, pages 968–973. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society Press: Los Alamitos, CA, USA, 2002. doi: 10.1109/CEC.2002.1007056.

221. Paul C. Winter, G. Ivor Hickey, and Hugh L. Fletcher. *Instant Notes in Genetics*. BIOS Scientific Publishers Ltd.: Oxford, UK, Taylor and Francis LLC: London, UK, and Science Press: Běijīng, China, 1st edition, 1998. ISBN ISBN: 0-387-91562-1, ISBN: 041537619X, ISBN: 1-85996-166-5, ISBN: 1-85996-262-9, and ISBN: 703007307X.

222. Masaharu Munetomo and David Edward Goldberg. Linkage Identification by Non-monotonicity Detection for Overlapping Functions. *Evolutionary Computation*, 7(4):377–398, 1999. doi: 10.1162/evco.1999.7.4.377.

223. Masaharu Munetomo and David Edward Goldberg. Linkage Identification by Non-monotonicity Detection for Overlapping Functions. IlliGAL Report 99005, Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Computer Science, Department of General Engineering, University of Illinois at Urbana-Champaign: Urbana-Champaign, IL, USA, January 1999. URL `http://www.illigal.uiuc.edu/pub/papers/IlliGALs/99005.ps.Z`.

224. Georges Raif Harik. *Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty using Genetic Algorithms*. PhD thesis, University of Michigan: Ann Arbor, MI, USA, 1997.

225. Ying-Ping Chen. *Extending the Scalability of Linkage Learning Genetic Algorithms – Theory & Practice*, volume 190/2006 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag GmbH: Berlin, Germany, 2004. ISBN ISBN: 3-540-28459-1. doi: 10.1007/b102053.

226. Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-Objective Test Problems, Linkages, and Evolutionary Methodologies. In Maarten Keijzer and Mike Cattolico, editors, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pages 1141–1148. ACM Press: New York, NY, USA, 2006. doi: 10.1145/1143997.1144179.

227. David Edward Goldberg, Kalyanmoy Deb, and Bradley Korb. Messy Genetic Algorithms: Motivation, Analysis, and First Results. *Complex Systems*, 3(5):493–530, 1989. URL `http://www.complex-systems.com/pdf/03-5-5.pdf`.

228. Martin Pelikan, David Edward Goldberg, and Erick Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In Wolfgang Banzhaf, Jason M. Daida, Ágoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark J. Jakiela, and Robert Elliott Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99)*, pages 525–532. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1999. URL `https://eprints.kfupm.edu.sa/28537/`.

229. Erick Cantú-Paz, Martin Pelikan, and David Edward Goldberg. Linkage Problem, Distribution Estimation, and Bayesian Networks. *Evolutionary Computation*, 8(3):311–340, 2000. doi: 10.1162/106365600750078808.

230. Peter John Angeline and Jordan B. Pollack. Evolutionary Module Acquisition. In David B. Fo-

gel and Wirt Atmar, editors, *Proceedings of the 2nd Annual Conference on Evolutionary Programming (EP'93)*, pages 154–163. Evolutionary Programming Society: San Diego, CA, USA, 1993. URL `http://www.demo.cs.brandeis.edu/papers/ep93.pdf`.

231. Wénxiáng Chén, Thomas Weise, Zhènyǔ Yáng, and Kē Táng. Large-Scale Global Optimization Using Cooperative Coevolution with Variable Interaction Learning. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Proceedings of the 11th International Conference on Parallel Problem Solving From Nature, Part 2 (PPSN'10-2)*, volume 6239 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 300–309. Springer-Verlag GmbH: Berlin, Germany, 2010. doi: 10.1007/978-3-642-15871-1_31. URL `http://mail.ustc.edu.cn/~chenwx/ppsn10Chen.pdf`.
232. Mitchell A. Potter and Kenneth Alan De Jong. A Cooperative Coevolutionary Approach to Function Optimization. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 249–257. Springer-Verlag GmbH: Berlin, Germany, 1994. doi: 10.1007/3-540-58484-6_269.
233. Mitchell A. Potter and Kenneth Alan De Jong. Cooperative Coevolution: An Architecture for Evolving Coadapted Subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000. doi: 10.1162/106365600568086. URL `http://mitpress.mit.edu/journals/EVCO/Potter.pdf`.
234. Yàochū Jīn and Jürgen Branke. Evolutionary Optimization in Uncertain Environments – A Survey. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 9(3):303–317, June 2005. doi: 10.1109/TEVC.2005.846356. URL `http://www.cs.le.ac.uk/people/sy11/ECiDUE/JinBranke-TEVC05.pdf`.
235. Shengxiang Yang, Yew-Soon Ong, and Yàochū Jīn, editors. *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51/2007 of *Studies in Computational Intelligence*. Springer-Verlag: Berlin/Heidelberg, 2007. ISBN ISBN: 3-540-49772-2 and ISBN: 3-540-49774-9.
236. Brad L. Miller and David Edward Goldberg. Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. *Evolutionary Computation*, 4(2):113–131, 1996. doi: 10.1162/evco.1996.4.2.113.
237. Brad L. Miller and David Edward Goldberg. Genetic Algorithms, Tournament Selection, and the Effects of Noise. IlliGAL Report 95006, Illinois Genetic Algorithms Laboratory (IlliGAL), Department of Computer Science, Department of General Engineering, University of Illinois at Urbana-Champaign: Urbana-Champaign, IL, USA, July 1995. URL `http://www.illigal.uiuc.edu/pub/papers/IlliGALs/95006.ps.Z`.
238. Jack Yiu-Bun Lee and P. C. Wong. The Effect of Function Noise on GP Efficiency. In Xīn Yáo, editor, *Progress in Evolutionary Computation, AI'93 (Melbourne, Victoria, Australia, 1993-11-16) and AI'94 Workshops (Armidale, NSW, Australia, 1994-11-22/23) on Evolutionary Computation, Selected Papers*, volume 956/1995 of *Lecture Notes in Computer Science (LNCS)*, pages 1–16. Springer-Verlag GmbH: Berlin, Germany, 1993. doi: 10.1007/3-540-60154-6_43.
239. L. S. Gurin and Leonard A. Rastrigin. Convergence of the Random Search Method in the Presence of Noise. *Automation and Remote Control*, 26:1505–1511, 1965.
240. J. Michael Fitzpatrick and John J. Grefenstette. Genetic Algorithms in Noisy Environments. *Machine Learning*, 3(2–3):101–120, October 1998. doi: 10.1007/BF00113893. URL `http://www.springerlink.com/content/n6w328441t63q374/fulltext.pdf`.
241. Shigeyoshi Tsutsui, Ashish Ghosh, and Yoshiji Fujimoto. A Robust Solution Searching Scheme in Genetic Search. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, volume 1141/1996 of *Lecture Notes in Computer Science (LNCS)*, pages 543–552. Springer-Verlag GmbH: Berlin, Germany, 1996. doi: 10.1007/3-540-61723-X_1018. URL `http://www.hannan-u.ac.jp/~tsutsui/ps/ppsn-iv.pdf`.
242. Shigeyoshi Tsutsui and Ashish Ghosh. Genetic Algorithms with a Robust Solution Searching Scheme. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 1(3):201–208, September 1997. doi: 10.1109/4235.661550.
243. Yasuhito Sano and Hajime Kita. Optimization of Noisy Fitness Functions by Means of Genetic Algorithms Using History of Search. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xīn Yáo, Evelyne Lutton, Juan Julián Merelo-

Guervós, and Hans-Paul Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917/2000 of *Lecture Notes in Computer Science (LNCS)*, pages 571–580. Springer-Verlag GmbH: Berlin, Germany, 2000. doi: 10.1007/3-540-45356-3_56.

244. Yasuhito Sano and Hajime Kita. Optimization of Noisy Fitness Functions by means of Genetic Algorithms using History of Search with Test of Estimation. In David B. Fogel, Mohamed A. El-Sharkawi, Xīn Yáo, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1-2, pages 360–365. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society Press: Los Alamitos, CA, USA, 2002. doi: 10.1109/CEC.2002.1006261.

245. Hajime Kita and Yasuhito Sano. Genetic Algorithms for Optimization of Noisy Fitness Functions and Adaptation to Changing Environments. In Masuo Suzuki, H. Nishimori, Yoshiyuki Kabashima, Kazuyuki Tanaka, and T. Tanaka, editors, *2003 Joint Workshop of Hayashibara Foundation and Hayashibara Forum – Physics and Information – and Workshop on Statistical Mechanical Approach to Probabilistic Information Processing (SMAPIP)*, 2003. URL `http://www.smapip.is.tohoku.ac.jp/~smapip/2003/hayashibara/proceedings/HajimeKita.pdf`.

246. Hans-Georg Beyer. Toward a Theory of Evolution Strategies: Some Asymptotical Results from the $(1, +\lambda)$-Theory. *Evolutionary Computation*, 1(2):165–188, 1993. URL `http://ls11-www.cs.uni-dortmund.de/~beyer/coll/Bey93a/Bey93a.ps`.

247. Thomas Bäck and Ulrich Hammel. Evolution Strategies Applied to Perturbed Objective Functions. In Zbigniew Michalewicz, James David Schaffer, Hans-Paul Schwefel, David B. Fogel, and Hiroaki Kitano, editors, *Proceedings of the First IEEE Conference on Evolutionary Computation (CEC'94), 1994 IEEE World Congress on Computation Intelligence (WCCI'94)*, volume 1, pages 40–45. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 1994. doi: 10.1109/ICEC.1994.350045. URL `http://citeseer.ist.psu.edu/16973.html`.

248. Ulrich Hammel and Thomas Bäck. Evolution Strategies on Noisy Functions: How to Improve Convergence Properties. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 159–168. Springer-Verlag GmbH: Berlin, Germany, 1994. doi: 10.1007/3-540-58484-6_260.

249. Hui Pan, Ling Wang, and Bo Liu. Particle Swarm Optimization for Function Optimization in Noisy Environment. *Journal of Applied Mathematics and Computation*, 181(2):908–919, October 15, 2006. doi: 10.1016/j.amc.2006.01.066.

250. Lin Han and Xingshi He. A Novel Opposition-Based Particle Swarm Optimization for Noisy Problems. In Jingsheng Lei, JingTao Yao, and Qingfu Zhang, editors, *Proceedings of the Third International Conference on Advances in Natural Computation (ICNC'07)*, volume 3, pages 624–629. IEEE Computer Society Press: Los Alamitos, CA, USA, 2007. doi: 10.1109/ICNC.2007.119.

251. Jürgen Branke. Creating Robust Solutions by Means of Evolutionary Algorithms. In Ágoston E. Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498/1998 of *Lecture Notes in Computer Science (LNCS)*, pages 119–128. Springer-Verlag GmbH: Berlin, Germany, 1998. doi: 10.1007/BFb0056855.

252. Genichi Taguchi. *Introduction to Quality Engineering: Designing Quality into Products and Processes.* Asian Productivity Organization (APO): Chiyoda-ku, Tōkyō, Japan and Kraus International Publications: Millwood, NY, USA, January 1986. ISBN ISBN: 9283310837 and ISBN: 9283310845. Translation of "Sekkeisha no tame no hinshitsu kanri".

253. Horst Greiner. Robust Filter Design by Stochastic Optimization. In Florin Abelès, editor, *Optical Interference Coatings*, volume 2253 of *The Proceedings of SPIE*, pages 150–160. Society of Photographic Instrumentation Engineers (SPIE): Bellingham, WA, USA, 1994. doi: 10.1117/12.192107.

254. Horst Greiner. Robust Optical Coating Design with Evolutionary Strategies. *Applied Optics*, 35 (28):5477–5483, October 1, 1996. doi: 10.1364/AO.35.005477.

255. Dirk Wiesmann, Ulrich Hammel, and Thomas Bäck. Robust Design of Multilayer Optical

Coatings by Means of Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 2(4):162–167, November 1998. doi: 10.1109/4235.738986.

256. Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, 2000. doi: 10.1162/106365600568202. URL `http://sci2s.ugr.es/docencia/cursoMieres/EC-2000-Comparison.pdf`.

257. Carlos Artemio Coello Coello. Evolutionary Multiobjective Optimization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5):444–447, September–October 2011. doi: 10.1002/widm.43.

258. Günter Rudolph. On a Multi-Objective Evolutionary Algorithm and Its Convergence to the Pareto Set. In Patrick K. Simpson, editor, *The 1998 IEEE International Conference on Evolutionary Computation (CEC'98), 1998 IEEE World Congress on Computation Intelligence (WCCI'98)*, pages 511–515. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 1998. doi: 10.1109/ICEC.1998.700081.

259. Vineet Khare, Xīn Yáo, and Kalyanmoy Deb. Performance Scaling of Multi-Objective Evolutionary Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, volume 2632/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 367–390. Springer-Verlag GmbH: Berlin, Germany, 2003. doi: 10.1007/3-540-36970-8_27.

260. Tomoyuki Hiroyasu, Hiroyuki Ishida, Mitsunori Miki, and Hisatake Yokouchi. Difficulties of Evolutionary Many-Objective Optimization. Intelligent Systems Design Laboratory Research Reports (ISDL Reports) 20081006004, Doshisha University, Department of Knowledge Engineering and Computer Sciences, Intelligent Systems Design Laboratory: Kyoto, Japan, October 13, 2008. URL `http://mikilab.doshisha.ac.jp/dia/research/report/2008/1006/004/report20081006004.html`.

261. Carlos Artemio Coello Coello, Gary B. Lamont, and David A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*, volume 5 of *Genetic and Evolutionary Computation*. Springer US: Boston, MA, USA and Kluwer Academic Publishers: Norwell, MA, USA, 2nd edition, 2002. ISBN ISBN: 0306467623 and ISBN: 0387332545. doi: 10.1007/978-0-387-36797-2.

262. Vineet Khare. Performance Scaling of Multi-Objective Evolutionary Algorithms. Master's thesis, University of Birmingham, School of Computer Science: Birmingham, UK, September 21, 2002. URL `http://www.lania.mx/~ccoello/EMOO/thesis_khare.pdf.gz`.

263. Antonio López Jaimes, Luis Vicente Santana-Quintero, and Carlos Artemio Coello Coello. Ranking Methods in Many-Objective Evolutionary Algorithms. In Raymond Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, volume 193/2009 of *Studies in Computational Intelligence*, pages 413–434. Springer-Verlag: Berlin/Heidelberg, 2009. URL `http://biblioteca.cinvestav.mx/indicadores/texto_completo/cinvestav/2009/165795_1.pdf`.

264. Robin Charles Purshouse. *On the Evolutionary Optimisation of Many Objectives.* PhD thesis, University of Sheffield, Department of Automatic Control and Systems Engineering: Sheffield, UK, September 2003.

265. Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos Artemio Coello Coello, and David Wolfe Corne, editors. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*, volume 1993/2001 of *Lecture Notes in Computer Science (LNCS)*, March 7–9, 2001. Springer-Verlag GmbH: Berlin, Germany. ISBN ISBN: 3-540-41745-1. doi: 10.1007/3-540-44719-9.

266. Marco Farina and Paolo Amato. On the Optimal Solution Definition for Many-Criteria Optimization Problems. In James M. Keller and Olfa Nasraoui, editors, *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS'00)*, pages 233–238. IEEE Computer Society: Piscataway, NJ, USA, 2002. doi: 10.1109/NAFIPS.2002.1018061. URL `http://www.lania.mx/~ccoello/farina02b.pdf.gz`.

267. Kalyanmoy Deb, Amrit Pratab, Samir Agrawal, and T Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 6(2):182–197, April 2002. doi: 10.1109/4235.996017. URL `http://dynamics.org/~altenber/UH_ICS/EC_REFS/`

MULTI_OBJ/DebPratapAgarwalMeyarivan.pdf.

268. David Wolfe Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xīn Yáo, Evelyne Lutton, Juan Julián Merelo-Guervós, and Hans-Paul Schwefel, editors, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN VI)*, volume 1917/2000 of *Lecture Notes in Computer Science (LNCS)*, pages 839–848. Springer-Verlag GmbH: Berlin, Germany, 2000. doi: 10.1007/3-540-45356-3_82. URL http://www.lania.mx/~ccoello/corne00.ps.gz.

269. Robin Charles Purshouse and Peter J. Fleming. Evolutionary Many-Objective Optimisation: An Exploratory Analysis. In Ruhul A. Sarker, Robert G. Reynolds, Hussein A. Abbass, Kay Chen Tan, Robert Ian McKay, Daryl Leslie Essam, and Tom Gedeon, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'03)*, volume 3, pages 2066–2073. IEEE Computer Society: Piscataway, NJ, USA, 2003. doi: 10.1109/CEC.2003.1299927. URL http://www.lania.mx/~ccoello/EMOO/purshouse03b.pdf.gz.

270. Evan J. Hughes. Evolutionary Many-Objective Optimisation: Many Once or One Many? In David Wolfe Corne, Zbigniew Michalewicz, Robert Ian McKay, Ágoston E. Eiben, David B. Fogel, Carlos M. Fonseca, Günther R. Raidl, Kay Chen Tan, and Ali M. S. Zalzala, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'05)*, volume 1, pages 222–227. IEEE Computer Society: Piscataway, NJ, USA, 2005. doi: 10.1109/CEC.2005.1554688. URL http://www.lania.mx/~ccoello/hughes05.pdf.gz.

271. Zhuo Kang, Lishan Kang, Xiufen Zou, Minzhong Liu, Changhe Li, Ming Yang, Yan Li, Yuping Chen, and Sanyou Zeng. A New Evolutionary Decision Theory for Many-Objective Optimization Problems. In Lishan Kang, Yong Liu, and Sanyou Zeng, editors, *Second International Symposium on Advances in Computation and Intelligence (ISICA'07)*, volume 4683/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 1–11. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-74581-5_1.

272. Andrzej Jaszkiewicz. On the Computational Efficiency of Multiple Objective Metaheuristics: The Knapsack Problem Case Study. *European Journal of Operational Research (EJOR)*, 158(2):418–433, October 16, 2004. doi: 10.1016/j.ejor.2003.06.015.

273. Hisao Ishibuchi, Yusuke Nojima, and Tsutomu Doi. Comparison between Single-Objective and Multi-Objective Genetic Algorithms: Performance Comparison and Performance Measures. In Gary G. Yen, Simon M. Lucas, Gary B. Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos Artemio Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06), 2006 IEEE World Congress on Computation Intelligence (WCCI'06)*, pages 3959–3966. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 2006. doi: 10.1109/CEC.2006.1688438.

274. Kokolo Ikeda, Hajime Kita, and Shigenobu Kobayashi. Failure of Pareto-based MOEAs: Does Non-Dominated Really Mean Near to Optimal? In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'01)*, volume 2, pages 957–962. IEEE Computer Society: Piscataway, NJ, USA, 2001. doi: 10.1109/CEC.2001.934293.

275. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In David B. Fogel, Mohamed A. El-Sharkawi, Xīn Yáo, Hitoshi Iba, Paul Marrow, and Mark Shackleton, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'02), 2002 IEEE World Congress on Computation Intelligence (WCCI'02)*, volume 1 and 1-2, pages 825–830. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society Press: Los Alamitos, CA, USA, 2002. doi: 10.1109/CEC.2002.1007032.

276. Denis Bouyssou. Building Criteria: A Prerequisite For MCDA. In Carlos A. Bana e Costa, editor, *Selected Readings from the Third International Summer School on Multicriteria Decision Aid: Methods, Applications, and Software (MCDA'90)*, pages 58–80. Springer-Verlag GmbH: Berlin, Germany, 1998. URL http://www.lamsade.dauphine.fr/~bouyssou/CRITERIA.PDF.

277. George A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, 63(2):81–97, March 1956. URL http://www.psych.utoronto.ca/users/peterson/psy430s2001/Miller\%20GA\%20Magical\%20Seven\%20Psych\%20Review\%201955.pdf.

278. Hisao Ishibuchi, Noritaka Tsukamoto, and

Yusuke Nojima. Evolutionary Many-Objective Optimization: A Short Review. In Zbigniew Michalewicz and Robert G. Reynolds, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Computational Intelligence: Research Frontiers – IEEE World Congress on Computational Intelligence – Plenary/Invited Lectures (WCCI)*, volume 5050/2008 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 2424–2431. IEEE Computer Society: Piscataway, NJ, USA, 2008. doi: 10.1109/CEC.2008.4631121. URL `http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e/research/pdf_file/multiobjective/CEC2008_Many_Objective_Final.pdf`.

279. Kalyanmoy Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley Interscience Series in Systems and Optimization. John Wiley & Sons Ltd.: New York, NY, USA, May 2001. ISBN ISBN: 047187339X.

280. Antonio López Jaimes and Carlos Artemio Coello Coello. Some Techniques to Deal with Many-Objective Problems. In Franz Rothlauf, Günther R. Raidl, Anna Isabel Esparcia-Alcázar, Ying-Ping Chen, Gabriela Ochoa, Ender Ozcan, Marc Schoenauer, Anne Auger, Hans-Georg Beyer, Nikolaus Hansen, Steffen Finck, Raymond Ros, L. Darrell Whitley, Garnett Wilson, Simon Harding, William B. Langdon, Man Leung Wong, Laurence D. Merkle, Frank W. Moore, Sevan G. Ficici, William Rand, Rick L. Riolo, Nawwaf Kharma, William R. Buckley, Julian Francis Miller, Kenneth Owen Stanley, Jaume Bacardit i Peñarroya, Will N. Browne, Jan Drugowitsch, Nicola Beume, Mike Preuß, Stephen L. Smith, Stefano Cagnoni, Alexandru Floares, Aaron Baughman, Steven Matt Gustafson, Maarten Keijzer, Arthur Kordon, and Clare Bates Congdon, editors, *Proceedings of the 11th Annual Conference – Companion on Genetic and Evolutionary Computation Conference (GECCO'09)*, pages 2693–2696. Association for Computing Machinery (ACM): New York, NY, USA, 2009. doi: 10.1145/1570256.1570386.

281. Kata Praditwong and Xīn Yáo. A New Multi-Objective Evolutionary Optimisation Algorithm: The Two-Archive Algorithm. In Yiu-ming Cheung, Yuping Wang, and Hailin Liu, editors, *Proceedings of the 2006 International Conference on Computational Intelligence and Security (CIS'06)*, volume 1, pages 286–291. IEEE (Institute of Electrical and Electronics Engineers): Piscataway, NJ, USA, 2006. doi: 10.1109/ICCIAS.2006.294139. URL `http://www.cs.bham.ac.uk/~xin/papers/CIS06TwoArchive.pdf`.

282. Kata Praditwong, Mark Harman, and Xīn Yáo. Software Module Clustering as a Multi-Objective Search Problem. *IEEE Transactions on Software Engineering*, 37(2):264–282, March–April 2011. doi: 10.1109/TSE.2010.26. URL `http://www.cs.bham.ac.uk/~xin/papers/PraditwongHarmanYao10TSE.pdf`.

283. Hiroyuki Sato, Arturo Hernández Aguirre, and Kiyoshi Tanaka. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO'07)*, volume 4403/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 5–20. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-70928-2_5.

284. Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi-Objective Optimisation Based on Relation *Favour*. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos Artemio Coello Coello, and David Wolfe Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO'01)*, volume 1993/2001 of *Lecture Notes in Computer Science (LNCS)*, pages 154–166. Springer-Verlag GmbH: Berlin, Germany, 2001. doi: 10.1007/3-540-44719-9_11. URL `http://www.lania.mx/~ccoello/EMOO/drechsler01.pdf.gz`.

285. Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi-objective Optimization in Evolutionary Algorithms Using Satisfiability Classes. In Bernd Reusch, editor, *International Conference on Computational Intelligence: Theory and Applications – 6th Fuzzy Days*, volume 1625/1999 of *Lecture Notes in Computer Science (LNCS)*, pages 108–117. International Biometric Society (IBS): Washington, DC, USA, 1996. doi: 10.1007/3-540-48774-3_14. URL `http://citeseer.ist.psu.edu/old/drechsler99multiobjective.html`.

286. André Sülflow, Nicole Drechsler, and Rolf Drechsler. Robust Multi-Objective Optimization in High Dimensional Spaces. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolu-*

tionary Multi-Criterion Optimization (EMO'07), volume 4403/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 715–726. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-70928-2_54. URL `http://www.informatik.uni-bremen.de/agra/doc/konf/emo2007_RobustOptimization.pdf`.

287. Mario Köppen and Kaori Yoshida. Substitute Distance Assignments in NSGA-II for Handling Many-Objective Optimization Problems. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO'07)*, volume 4403/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 727–741. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-70928-2_55.

288. David Wolfe Corne and Joshua D. Knowles. Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others. In Dirk Thierens, Hans-Georg Beyer, Josh C. Bongard, Jürgen Branke, John Andrew Clark, Dave Cliff, Clare Bates Congdon, Kalyanmoy Deb, Benjamin Doerr, Tim Kovacs, Sanjeev P. Kumar, Julian Francis Miller, Jason H. Moore, Frank Neumann, Martin Pelikan, Riccardo Poli, Kumara Sastry, Kenneth Owen Stanley, Thomas Stützle, Richard A. Watson, and Ingo Wegener, editors, *Proceedings of 9th Genetic and Evolutionary Computation Conference (GECCO'07)*, pages 773–780. ACM Press: New York, NY, USA, 2007. URL `http://dbkgroup.org/knowles/pap583s7-corne.pdf`.

289. Saku Kukkonen and Jouni A. Lampinen. Ranking-Dominance and Many-Objective Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07)*, pages 3983–3990. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/CEC.2007.4424990.

290. Tobias Wagner, Nicola Beume, and Boris Naujoks. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO'07)*, volume 4403/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 742–756. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-70928-2_56.

291. Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Iterative Approach to Indicator-based Multiobjective Optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07)*, pages 3967–3974. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/CEC.2007.4424988.

292. Karl Bringmann and Tobias Friedrich. The Maximum Hypervolume Set Yields Near-optimal Approximation. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'10)*, pages 511–518. ACM Press: New York, NY, USA, 2010. doi: 10.1145/1830483.1830576. URL `http://www.mpi-inf.mpg.de/~tfried/paper/2010GECCO_Hyp.pdf`.

293. Tobias Wagner, Nicola Beume, and Boris Naujoks. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. Reihe Computational Intelligence: Design and Management of Complex Technical Processes and Systems by Means of Computational Intelligence Methods CI-217/06, Universität Dortmund, Collaborative Research Center (Sonderforschungsbereich) 531: Dortmund, North Rhine-Westphalia, Germany, September 2006. URL `http://hdl.handle.net/2003/26125`.

294. Hisao Ishibuchi, Tsutomu Doi, and Yusuke Nojima. Incorporation of Scalarizing Fitness Functions into Evolutionary Multiobjective Optimization Algorithms. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund K. Burke, Juan Julián Merelo-Guervós, L. Darrell Whitley, and Xīn Yáo, editors, *Proceedings of 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*, volume 4193/2006 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 493–502. Springer-Verlag GmbH: Berlin, Germany, 2006. doi: 10.1007/11844297_50. URL `http://www.ie.osakafu-u.ac.jp/~hisaoi/ci_lab_e/research/pdf_file/multiobjective/PPSN_2006_EMO_Camera-Ready-HP.pdf`.

295. Hisao Ishibuchi and Yusuke Nojima. Optimization of Scalarizing Functions Through Evolutionary Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimiza-*

tion (EMO'07), volume 4403/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 51–65. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-70928-2_8. URL `http://ksuseer1.ist.psu.edu/viewdoc/summary?doi=10.1.1.78.8014`.

296. Evan J. Hughes. MSOPS-II: A General-Purpose Many-Objective Optimiser. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07)*, pages 3944–3951. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/CEC.2007.4424985.
297. Peter J. Fleming, Robin Charles Purshouse, and Robert J. Lygoe. Many-Objective Optimization: An Engineering Design Perspective. In Carlos Artemio Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO'05)*, volume 3410/2005 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 14–32. Springer-Verlag GmbH: Berlin, Germany, 2005.
298. Kalyanmoy Deb and J. Sundar. Reference Point Based Multi-Objective Optimization using Evolutionary Algorithms. In Maarten Keijzer and Mike Cattolico, editors, *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, pages 635–642. ACM Press: New York, NY, USA, 2006. doi: 10.1145/1143997.1144112. URL `www.lania.mx/~ccoello/deb06b.pdf.gz`.
299. Lothar Thiele, Kaisa Miettinen, Pekka J. Korhonen, and Julian Molina. A Preference-based Interactive Evolutionary Algorithm for Multiobjective Optimization. HSE Working Paper W-412, Helsinki School of Economics (HSE, Helsingin kauppakorkeakoulu): Helsinki, Finland, January 2007. URL `ftp://ftp.tik.ee.ethz.ch/pub/people/thiele/paper/TMKM07.pdf`.
300. Dimo Brockhoff and Eckart Zitzler. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund K. Burke, Juan Julián Merelo-Guervós, L. Darrell Whitley, and Xīn Yáo, editors, *Proceedings of 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*, volume 4193/2006 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 533–542. Springer-Verlag GmbH: Berlin, Germany, 2006. doi: 10.1007/11844297_54. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/bz2006d.pdf`.
301. Dimo Brockhoff and Eckart Zitzler. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In Karl-Heinz Waldmann and Ulrike M. Stocker, editors, *Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Jointly Organized with the Austrian Society of Operations Research (ÖGOR) and the Swiss Society of Operations Research (SVOR)*, volume 2006 of *Operations Research Proceedings*, pages 423–429. Springer-Verlag: Berlin/Heidelberg, 2006. doi: 10.1007/978-3-540-69995-8_68. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/bz2007d.pdf`.
302. Kalyanmoy Deb and Dhish Kumar Saxena. On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. KanGAL Report 2005011, Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur (IIT): Kanpur, Uttar Pradesh, India, December 2005. URL `http://www.iitk.ac.in/kangal/papers/k2005011.pdf`.
303. Dhish Kumar Saxena and Kalyanmoy Deb. Dimensionality Reduction of Objectives and Constraints in Multi-Objective Optimization Problems: A System Design Perspective. In Zbigniew Michalewicz and Robert G. Reynolds, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'08), Computational Intelligence: Research Frontiers – IEEE World Congress on Computational Intelligence – Plenary/Invited Lectures (WCCI)*, volume 5050/2008 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 3204–3211. IEEE Computer Society: Piscataway, NJ, USA, 2008. doi: 10.1109/CEC.2008.4631232.
304. Dimo Brockhoff and Eckart Zitzler. Improving Hypervolume-based Multiobjective Evolutionary Algorithms by using Objective Reduction Methods. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07)*, pages 2086–2093. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/CEC.2007.4424730. URL `http://www.tik.ee.ethz.ch/sop/publicationListFiles/bz2007c.pdf`.
305. Takeshi Furuhashi and Tomohiro Yoshikawa. Vi-

sualization Techniques for Mining of Solutions. In *Proceedings of the 8th International Symposium on Advanced Intelligent Systems (ISIS'07)*, pages 68–71. Korean Fuzzy Logic and Intelligent System Society (KFIS), 2007. URL `http://isis2007.fuzzy.or.kr/submission/upload/A1484.pdf`.

306. Mario Köppen and Kaori Yoshida. Visualization of Pareto-Sets in Evolutionary Multi-Objective Optimization. In Andreas König, Mario Köppen, Ajith Abraham, Christian Igel, and Nikola Kasabov, editors, *Proceedings of the 7th International Conference on Hybrid Intelligent Systems (HIS'07)*, pages 156–161. IEEE Computer Society: Piscataway, NJ, USA, 2007. doi: 10.1109/HIS.2007.62.

307. Shigeru Obayashi and Daisuke Sasaki. Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization (EMO'03)*, volume 2632/2003 of *Lecture Notes in Computer Science (LNCS)*, pages 796–806. Springer-Verlag GmbH: Berlin, Germany, 2003. doi: 10.1007/3-540-36970-8_56. URL `http://www.ifs.tohoku.ac.jp/edge/library/Final\%20EMO2003.pdf`.

308. Kaisa Miettinen. *Nonlinear Multiobjective Optimization*, volume 12 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers: Norwell, MA, USA and Springer Netherlands: Dordrecht, Netherlands, 1998. ISBN ISBN: 0-7923-8278-1.

309. Richard Ernest Bellman. *Dynamic Programming*. Dover Books on Mathematics. Princeton University Press: Princeton, NJ, USA, 1957. ISBN ISBN: 0486428095.

310. Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press: Princeton, NJ, USA, 1961. ISBN ISBN: 0691079013.

311. Ashish Sabharwal. Combinatorial Problems I: Finding Solutions. In Silvio Franz, Matteo Marsili, and Haijun Zhou, editors, *2nd Asian-Pacific School on Statistical Physics and Interdisciplinary Applications*. Abdus Salam International Centre for Theoretical Physics (ICTP): Triest, Italy, Chinese Center of Advanced Science and Technology (CCAST): Běijīng, China, and Chinese Academy of Sciences, Kavli Institute of Theoretical Physics China (KITPC): Běijīng, China, 2008. URL `http://www.cs.cornell.edu/~sabhar/tutorials/kitpc08-combinatorial-problems-I.ppt`.

312. Gene M. Amdahl. Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In *American Federation of Information Processing Societies: Proceedings of the Spring Joint Computer Conference (AFIPS)*, pages 483–485. Association for Computing Machinery (ACM): New York, NY, USA and Academic Press: London, New York, 1967. doi: 10.1145/1465482.1465560. URL `http://www-inst.eecs.berkeley.edu/~n252/paper/Amdahl.pdf`.

313. Pu Liu, Francis C. M. Lau, Michael J. Lewis, and Cho-li Wang. A New Asynchronous Parallel Evolutionary Algorithm for Function Optimization. In Juan Julián Merelo-Guervós, Panagiotis Adamidis, Hans-Georg Beyer, José Luis Fernández-Villacañas Martín, and Hans-Paul Schwefel, editors, *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature (PPSN VII)*, volume 2439/2002 of *Lecture Notes in Computer Science (LNCS)*, pages 401–410. Springer-Verlag GmbH: Berlin, Germany, 2002. doi: 10.1007/3-540-45712-7_39.

314. Erick Cantú-Paz. A Survey of Parallel Genetic Algorithms. *Calculateurs Parallèles, Réseaux et Systèmes Répartis*, 10(2):141–171, 1998. URL `http://www.evolutionaria.com/publications/cparalleles98-survey.ps.gz`.

315. Enrique Alba Torres and José M. Troya. A Survey of Parallel Distributed Genetic Algorithms. *Complexity*, 4(4):31–52, March–April 1999. doi: 10.1002/(SICI)1099-0526(199903/04)4:4⟨31::AID-CPLX5⟩3.0.CO;2-4.

316. Enrique Alba Torres and Marco Tomassini. Parallelism and Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 6(5):443–462, October 2002. doi: 10.1109/TEVC.2002.800880.

317. Thomas Weise and Kurt Geihs. DGPF – An Adaptable Framework for Distributed Multi-Objective Search Algorithms Applied to the Genetic Programming of Sensor Networks. In Bogdan Filipič and Jurij Šilc, editors, *Proceedings of the Second International Conference on Bioinspired Optimization Methods and their Applications (BIOMA'06)*, Informacijska Družba (Information Society), pages 157–166. Jožef Stefan Institute: Ljubljana, Slovenia, 2006. URL `http://www.it-weise.de/research/publications/WG2006DAAFFDMOSAATTGPOSN/WG2006DAAFFDMOSAATTGPOSN.pdf`.

318. Rainer Hauser and Reinhard Männer. Imple-

mentation of Standard Genetic Algorithm on MIMD Machines. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Proceedings of the Third Conference on Parallel Problem Solving from Nature; International Conference on Evolutionary Computation (PPSN III)*, volume 866/1994 of *Lecture Notes in Computer Science (LNCS)*, pages 504–513. Springer-Verlag GmbH: Berlin, Germany, 1994. doi: 10.1007/3-540-58484-6_293.

319. Simon Harding and Wolfgang Banzhaf. Fast Genetic Programming on GPUs. In Marc Ebner, Michael O'Neill, Anikó Ekárt, Leonardo Vanneschi, and Anna Isabel Esparcia-Alcázar, editors, *Proceedings of the 10th European Conference on Genetic Programming (EuroGP'07)*, volume 4445/2007 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 90–101. Springer-Verlag GmbH: Berlin, Germany, 2007. doi: 10.1007/978-3-540-71605-1_9. URL http://www.cs.mun.ca/~banzhaf/papers/eurogp07.pdf.

320. William B. Langdon and Wolfgang Banzhaf. A SIMD Interpreter for Genetic Programming on GPU Graphics Cards. In Michael O'Neill, Leonardo Vanneschi, Steven Matt Gustafson, Anna Isabel Esparcia-Alcázar, Ivanoe de Falco, Antonio Della Cioppa, and Ernesto Tarantino, editors, *Genetic Programming – Proceedings of the 11th European Conference on Genetic Programming (EuroGP'08)*, volume 4971/2008 of *Theoretical Computer Science and General Issues (SL 1), Lecture Notes in Computer Science (LNCS)*, pages 73–85. Springer-Verlag GmbH: Berlin, Germany, 2008. doi: 10.1007/978-3-540-78671-9_7. URL http://www.cs.bham.ac.uk/~wbl/biblio/gp-html/langdon_2008_eurogp.html.

321. Weihang Zhu. Nonlinear Optimization with a Massively Parallel Evolution Strategy-Pattern Search Algorithm on Graphics Hardware. *Applied Soft Computing*, 11(2):1770–1781, March 2011. doi: 10.1016/j.asoc.2010.05.020.

322. Shigeyoshi Tsutsui and Yoshiji Fujimoto. Solving Quadratic Assignment Problems by Genetic Algorithms with GPU Computation: A Case Study. In Franz Rothlauf, Günther R. Raidl, Anna Isabel Esparcia-Alcázar, Ying-Ping Chen, Gabriela Ochoa, Ender Ozcan, Marc Schoenauer, Anne Auger, Hans-Georg Beyer, Nikolaus Hansen, Steffen Finck, Raymond Ros, L. Darrell Whitley, Garnett Wilson, Simon Harding, William B. Langdon, Man Leung Wong, Laurence D. Merkle, Frank W. Moore, Sevan G. Ficici, William Rand, Rick L. Riolo, Nawwaf Kharma, William R. Buckley, Julian Francis Miller, Kenneth Owen Stanley, Jaume Bacardit i Peñarroya, Will N. Browne, Jan Drugowitsch, Nicola Beume, Mike Preuß, Stephen Frederick Smith, Stefano Cagnoni, Alexandru Floares, Aaron Baughman, Steven Matt Gustafson, Maarten Keijzer, Arthur Kordon, and Clare Bates Congdon, editors, *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*, pages 2523–2530. Association for Computing Machinery (ACM): New York, NY, USA, 2009. doi: 10.1145/1570256.1570355. URL http://www2.hannan-u.ac.jp/~tsutsui/ps/gecco/wk3006-tsutsui.pdf.

323. Marc Dubreuil, Christian Gagné, and Marc Parizeau. Analysis of a Master-Slave Architecture for Distributed Evolutionary Computations. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 36(1):229–235, February 2006. doi: 10.1109/TSMCB.2005.856724. URL http://vision.gel.ulaval.ca/~parizeau/publications/smc06.pdf.

324. Shisanu Tongchim and Xīn Yáo. Parallel Evolutionary Programming. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'04)*, volume 2, pages 1362–1367. IEEE Computer Society Press: Los Alamitos, CA, USA, 2004. doi: 10.1109/CEC.2004.1331055. URL http://www.cs.bham.ac.uk/~xin/papers/TongchimYao_CEC04.pdf.

325. Conor Ryan, J. J. Collins, and Michael O'Neill. Grammatical Evolution: Evolving Programs for an Arbitrary Language. In Wolfgang Banzhaf, Riccardo Poli, Marc Schoenauer, and Terence Claus Fogarty, editors, *Proceedings of the First European Workshop on Genetic Programming (EuroGP'98)*, volume 1391/1998 of *Lecture Notes in Computer Science (LNCS)*, pages 83–95. Springer-Verlag GmbH: Berlin, Germany, 1998. URL http://www.grammatical-evolution.org/papers/eurogp98.ps.

326. Michael O'Neill and Conor Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in an Arbitrary Language*, volume 4 of *Genetic Programming Series*. Springer Science+Business Media, Inc.: New York, NY, USA, 2003. ISBN ISBN: 1402074441.

327. Alexandre Devert. *Building Processes Optimization: Toward an Artificial Ontogeny based Approach.* PhD thesis, Université Paris-Sud, Ecole Doctorale d'Informatique: Paris, France and Institut National de Recherche en Informatique et

en Automatique (INRIA), Centre de Recherche Saclay – Île-de-France: Orsay, France, May 2009.

328. Phil Husbands and Frank Mill. Simulated Co-Evolution as the Mechanism for Emergent Planning and Scheduling. In Richard K. Belew and Lashon Bernard Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA'91)*, pages 264–270. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1991. URL `http://www.informatics.sussex.ac.uk/users/philh/pubs/icga91Husbands.pdf`.

329. Zhènyǔ Yáng, Kē Táng, and Xīn Yáo. Large Scale Evolutionary Optimization using Cooperative Coevolution. *Information Sciences – Informatics and Computer Science Intelligent Systems Applications: An International Journal*, 178 (15), August 1, 2008. doi: 10.1016/j.ins.2008.02.017. URL `http://nical.ustc.edu.cn/papers/yangtangyao_ins.pdf`.

330. Antonio LaTorre, José María Peña, Santiago Muelas, and Manuel Zaforas. Hybrid Evolutionary Algorithms for Large Scale Continuous Problems. In Franz Rothlauf, Günther R. Raidl, Anna Isabel Esparcia-Alcázar, Ying-Ping Chen, Gabriela Ochoa, Ender Ozcan, Marc Schoenauer, Anne Auger, Hans-Georg Beyer, Nikolaus Hansen, Steffen Finck, Raymond Ros, L. Darrell Whitley, Garnett Wilson, Simon Harding, William B. Langdon, Man Leung Wong, Laurence D. Merkle, Frank W. Moore, Sevan G. Ficici, William Rand, Rick L. Riolo, Nawwaf Kharma, William R. Buckley, Julian Francis Miller, Kenneth Owen Stanley, Jaume Bacardit i Peñarroya, Will N. Browne, Jan Drugowitsch, Nicola Beume, Mike Preuß, Stephen Frederick Smith, Stefano Cagnoni, Alexandru Floares, Aaron Baughman, Steven Matt Gustafson, Maarten Keijzer, Arthur Kordon, and Clare Bates Congdon, editors, *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation (GECCO'09)*, pages 1863–1864. Association for Computing Machinery (ACM): New York, NY, USA, 2009. doi: 10.1145/1569901.1570205.

331. Fei Peng, Kē Táng, Guóliáng Chén, and Xīn Yáo. Population-based Algorithm Portfolios for Numerical Optimization. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 14(5): 782–800, March 29, 2010. doi: 10.1109/TEVC.2010.2040183.

332. David H. Wolpert and William G. Macready. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 1(1):67–82, April 1997. doi: 10.1109/4235.585893.

333. Anne Auger and Oliver Teytaud. Continuous Lunches are Free Plus the Design of Optimal Optimization Algorithms. Rapports de Recherche inria-00369788, Institut National de Recherche en Informatique et en Automatique (INRIA), March 21, 2009. URL `http://hal.inria.fr/docs/00/36/97/88/PDF/ccflRevisedVersionAugerTeytaud.pdf`.

334. Anne Auger and Oliver Teytaud. Continuous Lunches are Free Plus the Design of Optimal Optimization Algorithms. *Algorithmica*, 57(1):121–146, May 2010. doi: 10.1007/s00453-008-9244-5.

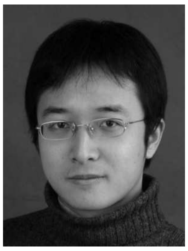
335. Nicholas J. Radcliffe. The Algebra of Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence*, 10(4):339–384, December 1994. doi: 10.1007/BF01531276. URL `http://stochasticsolutions.com/pdf/amai94.pdf`.

Thomas Weise (M'07) received the Diplom Informatiker (equivalent to M.Sc.) degree from the Department of Computer Science, Chemnitz University of Technology, Chemnitz, Germany, in 2005, and the Ph.D. degree at the Distributed Systems Group of the Fachbereich Elektrotechnik and Informatik, University of Kassel, Kassel, Germany in 2009.

Since 2009, he is with the Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China. His major research interests include Evolutionary Computation, Genetic Programming, and real-world applications of optimization algorithms. His experience ranges from applying GP to distributed systems and multi-agent systems, efficient web service composition for Service Oriented Architectures, to solving large-scale real-world vehicle routing problems for multimodal logistics and transportation.

Besides being the author/co-author of over 60 refereed publications, Dr. Weise also authors the electronic book *Global Optimization Algorithms – Theory and Application* which is freely available at his website `http://www.it-weise.de/`.

Raymond Chiong (M'05) is with the Faculty of Higher Education Lilydale, Swinburne University of Technology, Australia. He has been lecturing in Computer Science/Information Systems for many years. His teaching has focused on programming and databases. Besides teaching, he has been actively pursuing research in computational and artificial intelligence, with a particular interest in the applications of nature-inspired computational methodologies to various optimisation, scheduling and planning problems. He is the Editor-in-Chief of the Interdisciplinary Journal of Information, Knowledge, and Management (IJIKM), and Editor of the journal Engineering Applications of Artificial Intelligence (EAAI). He also serves on the review board/program committee of several international journals and conferences. To date, he has more than 70 refereed publications in books, journals and conference proceedings.



Ke Tang (S'05–M'07) received the B.Eng. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2002 and the Ph.D. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2007.

In 2007, he joined the Nature Inspired Computation and Applications Laboratory (NICAL), School of Computer Science and Technology, University of Science and Technology of China, Hefei, China, where he was promoted to Professor in 2011. He is the author/co-author of more than 60 refereed publications. His major research interests include evolutionary computation, machine learning, and their real-world applications.

Dr. Tang is an associate editor of IEEE Computational Intelligence Magazine and the Chair of the IEEE Task Force on Collaborative Learning and Optimization. He served as a program co-chair of 2010 IEEE Congress on Evolutionary Computation, held in Barcelona.

```
@article{WCT2012EOPABT,
  author       = {Thomas Weise and Raymond Chiong and K{\={e}} T{\'{a}}ng},
  title        = {Evolutionary Optimization: Pitfalls and Booby Traps},
  journal      = {Journal of Computer Science and Technology (JCST)},
  number       = {5},
  volume       = {27},
  pages        = {907--936},
  year         = {2012},
  dsdate       = sep # {~} # {2012},
  month        = sep,
  doi          = {10.1007/s11390-012-1274-4},
  note         = {Special Issue on Evolutionary Computation, edited by
                  Xin Yao and Pietro S. Oliveto.},
},
```