Coello Coello Carlos A., 0000-0002-8435-680X
CINVESTAV-IPN, Mexico City, Mexico, ccoello{at}cs.cinvestav.mx
Castillo Tapia Ma Guadalupe, 0000-0001-8371-521X
UAM Azcapotzalco, Mexico City, Mexico, mgct{at}azc.uam.mx

Multi-objective evolutionary algorithms and some of their applications in reliability

Keywords

multi-objective optimization, evolutionary algorithms, metaheuristics

Abstract

Multi-objective optimization has become increasingly important, mainly because many real-world problems are multi-objective in nature. The complexity of many of such problems has made necessary the use of metaheuristics. From them, the use of multi-objective evolutionary algorithms has become very popular mainly because of their ease of use and flexibility. In this chapter, we provide a short review of multi-objective evolutionary algorithms and some of their applications in reliability. In the final part of the chapter, some possible paths for future research in this area are also discussed.

1. Introduction

In many different disciplines, it is necessary to tackle problems having two or more (often conflicting) objectives at the same time. Such problems are called *multi-objective*, and their solution implies finding the best possible trade-offs among the objectives.

For several years, a wide variety of mathematical programming techniques were developed to deal with multi-objective optimization problems (Ehrgott, 2005; Miettinen, 1999). However, their limitations (e.g., most of these techniques can only generate a single solution at a time and have a limited applicability) motivated the development of alternative approaches from which multi-objective evolutionary algorithms have become a popular choice (Coello Coello et al., 2007; Deb, 2001).

Evolutionary algorithms (EAs) are a metaheuristic (Talbi, 2009) inspired on the “survival of the fittest” principle from Darwin's evolutionary theory (Goldberg, 1989). EAs have become very popular as multi-objective optimizers because of their ease of use (and implementation) and flexibility (e.g., EAs are less sensitive than mathematical programming techniques to the initial search points and to the shape and continuity of the Pareto front). Additionally, the fact that EAs are population-based techniques makes it possible to manage, simultaneously, a set of solutions, instead of one at a time, as normally happens with mathematical programming techniques.

The first Multi-Objective Evolutionary Algorithm (MOEA) was called *Vector Evaluated Genetic Algorithm* (VEGA) and was proposed by J. David Schaffer in the mid-1980s (Schaffer, 1984; Schaffer, 1985; Schaffer & Grefenstette, 1985). Something interesting is that there was not much interest in evolutionary multi-objective optimization (EMOO) research for almost a decade. However, in the mid-1990s, this area started to attract a lot of attention from several research groups around the world, and has maintained a high research activity since then. The first author maintains the EMOO repository (Coello Coello, 2006) which currently contains over 12,600 bibliography refer-

Safety and Reliability of Systems and Processes, Summer Safety and Reliability Seminar 2021.

DOI: 10.26408/srsp-2021-04.

ences on evolutionary multi-objective optimization. The EMOO repository is located at: https://emoo.cs.cinvestav.mx.
The remainder of this chapter is organized as follows. In Section 2, we provide some basic multi-objective optimization concepts required to make this chapter self-contained. Section 3 contains a short review of MOEAs from a historical perspective. Section 4 contains a short review of some representative applications of MOEAs in reliability. Section 5 indicates some potential paths for future research in this area. Finally, conclusions are provided in Section 6.

2. Basic concepts

In multi-objective optimization, the aim is to solve problems of the type (without loss of generality, we will assume only minimization problems):

$$Minimize\ \vec{f}(\vec{x}) := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})] \quad (1)$$

Subject to:

$$g_i(\vec{x}) \le 0,\ i = 1,2,\dots,m, \quad (2)$$

$$h_j(\vec{x}) = 0,\ j = 1,2,\dots,p, \quad (3)$$

where $\vec{x} = [x_1, x_2, \dots, x_n]^T$ is the vector of decision variables, $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1,\dots,k$ are the objective functions and $g_i, h_j: \mathbb{R}^n \to \mathbb{R}$, $i = 1,\dots,m$, $j = 1,\dots,p$ are the constraint functions of the problem.
A few additional definitions are required to introduce the notion of optimality used in multi-objective optimization.

Definition 1. Given two vectors $\vec{x}, \vec{y} \in \mathbb{R}^k$, we say that $\vec{x} \le \vec{y}$ if $x_i \le y_i$ for $i = 1,\dots,k$, and that $\vec{x}$ ***dominates*** $\vec{y}$ (denoted by $\vec{x} \prec \vec{y}$) if $\vec{x} \le \vec{y}$ and $\vec{x} \ne \vec{y}$.

Definition 2. We say that a vector of decision variables $\vec{x} \in \mathrm{X} \subset \mathbb{R}^n$ is ***nondominated*** with respect to X, if there does not exist another $\vec{y} \in \mathrm{X}$ such that $\vec{f}(\vec{y}) \prec \vec{f}(\vec{x})$.

Definition 3. We say that a vector of decision variables $\vec{x} \in \mathcal{F} \subset \mathbb{R}^n$ ($\mathcal{F}$ is the feasible region) is *Pareto optimal* if it is nondominated with respect to $\mathcal{F}$.

Definition 4. The *Pareto Optimal Set* P^* is defined by:

$$P^* = \{\vec{x} \in \mathcal{F} | \vec{x} \text{ is Pareto optimal}\}.$$

Definition 5. The *Pareto Front* PF^* is defined by:

$$PF^* = \left\{\vec{f}(\vec{x}) \in \mathbb{R}^k \middle| \vec{x} \in P^*\right\}.$$

Therefore, our aim is is to obtain the Pareto optimal set from the set $\mathcal{F}$ of all the decision variable vectors that satisfy eqs. (1) and (2). Note however that in practice, not all the Pareto optimal set is normally desirable or achievable, and decision makers tend to prefer certain types of solutions or regions of the Pareto front (Branke & Deb, 2005).

3. Review of multi-objective evolutionary algorithms

Although the first reference on the use of EAs for solving multi-objective problems dates back to the late 1960s (Rosenberg, 1967), the first actual implementation was developed in the mid-1980s (Schaffer, 1984; Schaffer, 1985). Next, we will provide a historical review of MOEAs and some additional mechanisms that have been incorporated into them over the years.

3.1. Early days

In their origins, MOEAs were very simple and naive. A good example of this is the Vector Evaluated Genetic Algorithm (VEGA), (Schaffer, 1985) in which the population of a simple genetic algorithm was subdivided into as many sub-populations as the number of objectives of the multi-objective optimization problem (MOP) to be solved (only problems with two objectives were normally considered at that time). Then, solutions in each subpopulation were selected based on their performance on a single objective (i.e., in the first subpopulation, individuals were selected based on their performance on the first objective and in the second subpopulation, individuals were selected based on their performance on the second objective). Then, the individuals of all the subpopulations were shuffled with the aim of recombining solutions that were the best in the first objective with those that were the best in the second objective. When combined with proportional selection, e.g., the roulette-wheel method (Goldberg, 1989), VEGA produced solutions similar to those obtained with the use of a linear aggregating function that combines all the objective functions into a single scalar value (Coello Coello, 1996). In spite of the limitations of VEGA, some researchers eventually found applications in which this sort of scheme could be useful, see for example (Coello Coello, 2000).

Linear aggregating functions were among the most popular approaches adopted in the early days of MOEAs (Hajela & Lin, 1992), but their incapability for dealing with non-convex Pareto fronts was soon pointed out by some researchers, see for example (Das & Dennis, 1997). Nevertheless, linear aggregating functions and other naive approaches, such as lexicographic ordering have survived in the EMO literature for many years (Coello Coello et al., 2007).

3.2. Pareto-based MOEAs

Goldberg proposed in his seminal book on genetic algorithms (Goldberg, 1989) a mechanism called *Pareto ranking* for the selection scheme of a MOEA. The core idea of Pareto ranking is to rank the population of an evolutionary algorithm based on Pareto optimality, such that the nondominated solutions obtain the highest (best) possible rank and are sampled at the same rate (i.e., all nondominated solutions have the same probability of survival). Since Goldberg did not provide a specific algorithm for Pareto ranking, several implementations were developed based on his proposal. From them, the two main ones were those provided in the Multi-Objective Genetic Algorithm (MOGA) of Fonseca and Fleming (Fonseca & Fleming, 1993) and the Nondominated Sorting Genetic Algorithm (NSGA) of Srinivas and Deb (Srinivas & Deb, 1994). In the first (MOGA), the ranking was done in a single pass (by comparing each individual with respect to everybody else, in terms of Pareto optimality), whereas the second required the creation of several layers of solutions, which involved re-ranking the population several times (i.e., NSGA was more computationally expensive than MOGA).

Goldberg (Goldberg, 1989) realized that in MOEAs, diversity would be a key issue if we aimed to generate as many elements of the Pareto optimal set as possible in a single algorithmic execution. This gave rise to the use of a mechanism that was eventually called *density estimator*, whose task is to maintain different (nondominated) solutions in the population, thus avoiding convergence to a single solution (something that eventually happens with any evolutionary algorithm if it is allowed to run for too many generations, because of stochastic noise (Goldberg, 1989). MOGA (Fonseca & Fleming, 1993) and NSGA (Srinivas & Deb, 1994) used fitness sharing (Goldberg & Richardson, 1987) as their density estimator, but a wide variety of other approaches have been proposed since then: clustering (Zitzler & Thiele, 1999), adaptive grids (Knowles & Corne, 2003), crowding (Deb et al., 2002), entropy (Pires et al., 2013) and parallel coordinates (Hernández Gómez et al., 2016), among others.

In the late 1990s, another mechanism was incorporated into MOEAs: elitism. The idea of ***elitism*** is to retain the best solutions obtained by a MOEA so that they are not destroyed by the evolutionary operators (i.e., crossover and mutation). However, since all nondominated solutions are considered equally good (unless we have some preference information), this leads to the generation of a large number of solutions. Zitzler realized this when developing the Strength Pareto Evolutionary Algorithm (SPEA), (Zitzler & Thiele, 1999) and also observed that retaining such a large number of solutions diluted the selection pressure. Thus, he proposed not only to use an external archive to store the nondominated solutions generated by his MOEA, but also proposed to prune such an archive once a certain (user-defined) limit was reached. For this sake, he adopted clustering. Elitism is important not only for practical reasons (it is easier to compare the performance of two MOEAs that produce the same number of nondominated solutions), but also for theoretical reasons, since it has been proved that such a mechanism is required in a MOEA to guarantee convergence (Rudolph & Agapie, 2000). Pareto-based MOEAs were very popular in the mid-1990s, but few of the many approaches that were proposed at that time have been actually used by other researchers. With no doubt, the most popular of the Pareto-based MOEAs has been the Nondominated Sorting Genetic Algorithm-II (NSGA-II), (Deb et al., 2002) which uses a more efficient ranking scheme (called *nondominated sorting*) than its predecessor (NSGA), and adopts a clever mechanism called crowded comparison operator (which does not require any parameters), as its density estimator. NSGA-II is still used today by many researchers, in spite of the well-known limitations of its crowded comparison operator when dealing with MOPs having more than three objectives (the so-called ***many-objective optimization problems*** (Coello Coello et al., 2007). In fact, there is empirical evidence indicating that the crowded comparison operator has difficulties even in MOPs with only 3 objectives, see for example (Kukkonen & Deb, 2006).

3.3. Indicator-based MOEAs

For over 10 years, Pareto-based MOEAs were, by far, the most popular approaches in the specialized literature. In 2004, a different type of algorithmic design was proposed, although it remained underdeveloped for several years: indicator-based selection. The core idea of this sort of MOEA was introduced in the Indicator-Based Evolutionary Algorithm (IBEA), (Zitzler & Künzli, 2004) which consists of an algorithmic framework that allows the incorporation of any performance indicator into the selection mechanism of a MOEA. IBEA was originally tested with the hypervolume (Zitzler, 1999) and the binary indicator (Zitzler & Künzli, 2004).

Indicator-based MOEAs were initially seen as a curiosity in the field, since it was not clear what were their advantages other than providing an alternative mechanism for selecting solutions. However, when the limitations of Pareto-based selection for dealing with many-objective problems became clear, researchers started to get interested in indicator-based MOEAs, which did not seem to have scalability limitations. Much of the interest in this area was produced by the introduction of the S Metric Selection Evolutionary Multiobjective Algorithm (SMS-EMOA) in 2005 (Emmerich et al., 2005). SMS-EMOA randomly generates an initial population and then produces a single solution per iteration (i.e., it uses steady state selection) adopting the crossover and mutation operators from NSGA-II. Then, it applies nondominated sorting (as in NSGA-II). When the last nondominated front has more than one solution, SMS-EMOA uses hypervolume (Zitzler, 1999) to decide which solution should be removed. In other words, SMS-EMOA is a steady state version of NSGA-II in which the hypervolume replaces the crowded comparison operator.

Beume et al. (Beume et al., 2007) proposed a new version of SMS-EMOA in which the hypervolume contribution is not used when, in the nondominated sorting process, we obtain more than one front (clearly, the hypervolume is used as a density estimator). In this case, they use the number of solutions that dominate to a certain individual (i.e., the solution that is dominated by the largest number of solutions is removed). This version of SMS-EMOA is more efficient. However, since this MOEA relies on the use of exact hypervolume contributions, it eventually becomes too computationally expensive as we increase the number of objectives (Beume et al., 2009).

SMS-EMOA started a trend for designing indicator-based MOEAs (several of which rely on the hypervolume indicator) although it is worth indicating that in such approaches, the performance indicator has been mostly used as a density estimator, see for example (Igel et al., 2007). The use of "pure" indicator-based selection mechanisms has been very rare in the specialized literature, see for example (Menchaca-Mendez & Coello Coello, 2017).

At this point, an obvious question is: why is that the ***hypervolume*** is such an attractive choice for indicator-based selection? The hypervolume (which is also known as the $\mathcal{S}$ metric or the Lebesgue Measure) of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively. One of its main advantages are its mathematical properties, since it has been proved that the maximization of this performance measure is equivalent to finding the Pareto optimal set (Fleischer, 2003). Additionally, empirical studies have shown that (for a certain number of points previously determined) maximizing the hypervolume indeed produces subsets of the true Pareto front (Knowles & Corne, 2003; Emmerich et al., 2005).

Additionally, the hypervolume assesses both convergence and, to a certain extent, also the spread of solutions along the Pareto front (although without necessarily enforcing a uniform distribution of solutions). However, there are several issues regarding the use of the hypervolume. First, the computation of this performance indicator depends of a reference point, which can influence the results in a significant manner. Some people have proposed to use the worst objective function values in the current population, but this requires scaling of the objectives. Nevertheless, the most serious limitation of the hypervolume is its high computational cost. The best algorithms known to compute hypervolume have a polynomial complexity on the number of points used, but such complexity grows exponentially on the number of objectives (Beume et al., 2009). This has triggered a significant amount of research regarding algorithms that can reduce the computational cost of computing the hypervolume and the hypervolume contributions, which is what we need for a hypervolume-based MOEA, see for example (Russo & Francisco, 2016; Cox & Whiley, 2016; Lacour et al., 2017; Jaszkiewicz, 2018; Guerreiro & Fonseca, 2018).

An obvious alternative to deal with this issue is to approximate the actual hypervolume contributions. This is the approach adopted by the Hypervolume Estimation Algorithm for Multi-Objective Optimization (HyPE), (Bader & Zitzler, 2011) in which Monte Carlo simulations were used to approximate exact hypervolume values. In spite of the fact that HyPE can efficiently solve MOPs having a very large number of objectives, its results are not as competitive as when using exact hypervolume contributions.

Another possibility is to use a different performance indicator whose computation is relatively inexpensive. Unfortunately, the hypervolume is the only unary indicator which is known to be Pareto compliant (Zitzler et al., 2003), which makes less attractive the use of other performance indicators. Nevertheless, there are some other performance indicators which are weakly Pareto compliant, such as R2 (Brockhoff et al., 2012) and the Inverted Generational Distance plus (IGD+) (Ishibuchi et al., 2015). Although several efficient and effective indicator-based MOEAs have been proposed around these performance indicators, see for example (Hernandez Gomez & Coello Coello, 2015; Brockhoff et al., 2015; Li et al., 2018; Manoatl Lopez & Coello Coello, 2016; Tian et al., 2016; Manoatl Lopez & Coello Coello, 2018), their use has remained relatively rare in the specialized literature.

More recently, some researchers have proposed mechanisms that combine different performance indicators (e.g., using ensembles) with the aim of providing more robust indicator-based MOEAs, see for example (Phan & Junichi, 2011; Falcón-Cardona et al., 2020).

3.4. Decomposition-based MOEAs

In 2007, a different sort of approach was proposed, quickly attracting a lot of interest: the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), (Zhang & Li, 2007). The idea of using decomposition (or scalarization) methods was originally proposed in mathematical programming more than 20 years ago (Das & Dennis, 1998) and it consists in transforming an MOP into several single-objective optimization problems which are then solved to generate the nondominated solutions of the original problem. Unlike linear aggregating functions, the use of scalarization (or decomposition) methods allows the generation of non-convex portions of the Pareto front and works even in disconnected Pareto fronts. MOEA/D presents an important advantage with respect to methods proposed in the mathematical programming literature (such as Normal Boundary Intersection (NBI), (Das & Dennis, 1998): it uses neighborhood search to solve simultaneously all the single-objective optimization problems generated from the transformation. Additionally, MOEA/D is not only effective and efficient, but can also be used for solving problems with more than 3 objectives although in such cases it will require higher population sizes.

Decomposition-based MOEAs became fashionable at around 2010 and have remained as an active research area since then (Santiago et al., 2014). In fact, this sort of approach influenced the development of the Nondominated Sorting Genetic Algorithm-III (NSGA-III), (Deb & Jain, 2014) which adopts both decomposition and reference points to deal with many-objective problems. However, it was recently found that decomposition-based MOEAs do not work properly with certain Pareto front geometries (Ishibuchi et al., 2017). This will certainly trigger a lot of research in the next few years, given the popularity of decomposition-based MOEAs.

4. Applications of MOEAs in reliability

A wide variety of system design and reliability optimization problems involve the incorporation of several conflicting objectives (e.g. cost, reliability and performance, among others). In fact, the use of multi-objective optimization in the design of reliability systems has been reported in the literature since the late 1970s, see for example (Inagaki et al., 1978; Hwang et al., 1979). However, the use of MOEAs in the design of reliability systems is much more recent. Next, we will briefly review some of the many applications reported in the specialized literature.

4.1. Network reliability

Kim and Gen (Kim & Gen, 1999) adopted a genetic algorithm to solve bi-objective network topology design problems of wide-band communication networks connected with fiber optic cable, considering network reliability. In this case, delay and cost are also the objectives considered by the authors, and a weighted linear aggregating function is adopted in combination with an evolutionary algorithm.

Kumar et al. (Kumar et al., 2002) used a multi-objective genetic algorithm to design a communications network subject to reliability and flow constraints. Two objectives were considered: delay and cost. The authors adopted the Pareto Converging Genetic Algorithm (Kumar & Rockett, 1998), which was developed by one of them. The authors showed that using a MOEA offered several advantages, since the network designer could have a range of network costs and packet delays to choose from, knowing that their corresponding topologies were reliable in case of single node failures and that it was guaranteed that the maximum packet load on any link would not exceed the link capacity.

Marseguerra et al. (Marseguerra et al., 2005) used a multi-objective genetic algorithm combined with Monte Carlo simulations to identify optimal network designs considering: the maximization of the network reliability estimate and the minimization of its associated variance when component types, under uncertain reliability, and redundancy levels are the decision variables. The authors seem to adopt an elitist version of the original Nondominated Sorting Genetic Algorithm (NSGA), (Srinivas & Deb, 1994). The Monte Carlo simulation was adopted by the authors for evaluating the two objective functions: the expectation of network reliability estimate and the negative of its variance. This approach was applied to two network design problems, with multiple choices of components' types available and the possibility of allocating redundancy. Design constraints on total cost and weight were also considered. The authors indicated that the obtained results provided a variety of alternatives to the user, which allowed the identification of a risk-averse network design characterized by a high degree of confidence in the actual network reliability.

Zhang et al. (Zhang et al., 2011) model a critical infrastructure as a complex network for which a new metric is defined to understand its reliability. This new metric describes the average reliability between every pair of nodes in a complex network. Then, in an effort to identify the most critical components that impact this metric, a multi-objective optimization problem called “the critical component detection problem” is introduced by the authors. Solving this MOP provides two important insights about the behavior of a complex network: (1) a set of nondominated solutions that identify the most critical components and (2) a quantitative assessment of how these failures affect the entire network. The MOEA adopted in this case is MP-PSDA which was proposed by some of the same authors (Claudio et al., 2009).

4.2. Design of circuits and devices

Deb et al. (Deb et al., 2004) treated the optimal placement of electronic components on a printed circuit board as a bi-objective problem. The objectives considered were: minimizing the overall wire length and minimizing the failure rate of the board arising due to uneven local heat accumulation. The authors adopt a novel representation scheme which enables the use of an easier recombination operator. The MOEA adopted in this case is the NSGA-II (Deb et al., 2002).

Two cases for which they had previous results were adopted by the authors. In both cases, the NSGA-II was able to find much better nondominated solutions which represented very interesting trade-offs. Regarding wire-length minimization, interconnected components were placed in independent clusters. Regarding failure-rate minimization due to temperature effects, the high-risk components (both in terms of high failure-rate and high heat-generation rates) were placed near the uninsulated boundaries, so that a small steady-state temperature was developed on them. In a final example, the authors kept the overall board size as a variable and two different sets of solutions (one with a 18×2 configuration and another one with a 6×6 configuration) emerged. This illustrated the flexibility of this sort of MOEA-based approach.

Zafiropoulos and Dialynas (Zafiropoulos & Dialynas, 2004) adopted both a single-objective and a multi-objective optimization approach for obtaining the optimal system structure of electronic devices while considering constraints on reliability and cost. In both cases, simulated annealing was adopted for performing the optimization (a linear aggregating function is adopted in this case). This approach was applied to a power electronic device for which the component failure rate uncertainty was modeled with two alternative probability distribution functions.

Bolchini et al. (Bolchini et al., 2010) proposed a framework for the design space exploration of reliable FPGA systems based on the use of a MOEA (NSGA-II), (Deb et al., 2002). The authors considered two objectives: (1) the average size of the reconfigurable areas required for implementing the reliable solutions, which is directly proportional to the reconfiguration time and (2) the dimension of the system that represents the effective cost of the application of reliability-oriented techniques due to the introduction of voters and the partitioning of the functional units in reconfigurable areas. An interesting aspect of this work is that the authors compare results with respect to a multi-objective version of simulated annealing called AMOSA (Bandyopadhyay, 2008) in terms of scalability, using three real-world circuits and a set of synthetic problems of different sizes. The authors reported that the NSGA-II was able to clearly outperform AMOSA.

4.3. Systems design

Sinha (Sinha, 2007) provides a methodology for reliability-based multi-objective optimization of large-scale engineering systems. Then, this methodology is applied to the vehicle crashworthiness design optimization for side impact, considering both structural crashworthiness and occupant safety. The author considered as objectives the structural weight and the front door velocity under side impact. The author adopted two first order reliability method-based techniques (i.e., approximate moment and reliability index) for uncertainty quantification. A software called GDOT was adopted. This software uses a multi-objective genetic algorithm for the optimization task. The results reported by the author indicate that the vehicle weight can be significantly reduced with respect to the baseline design, while reducing, at the same time, the door velocity. Something interesting of this work is that the author adopts a decision-making criterion to select a subset from all the nondominated solutions obtained by the multi-objective genetic algorithm.

Taboada et al. (Taboada et al., 2008) developed a tailored MOEA to solve multi-objective multi-state reliability optimization design problems. The authors adopt the multi-objective multi-state genetic algorithm (MOMS-GA). The objectives that they consider are: the maximization of the system availability and the minimization of both the system cost and the weight. MOMS-GA uses the universal moment generating function approach to evaluate the different reliability or availability indices of the system. The components are characterized by having different performance levels, cost, weight, and reliability. The authors present two examples to illustrate their approach. In both cases, MOMS-GA was able to obtain good trade-off solutions.

Deb et al. (Deb et al., 2009) showed how classical reliability-based concepts can be borrowed and modified and integrated into both single-objective and multi-objective evolutionary algorithms. The authors discuss three different optimization tasks in which classical reliability-based optimization procedures usually have difficulties:

- reliability-based optimization problems having multiple local optima,
- finding and revealing reliable solutions for different reliability indices simultaneously by means of a bi-criterion optimization approach, and
- multi-objective optimization with uncertainty and specified system or component reliability values.

Each of these optimization tasks is illustrated by solving a number of test problems and a well-studied automobile design problem. Results are also compared with a classical reliability-based methodology. The MOEA adopted by the authors is the NSGA-II (Deb et al., 2002).

Ardakan and Rezvan (Ardakan & Rezvan, 2018) tackled the reliability-redundancy allocation problem, which involves the selection of components reliability and redundancy levels with the aim of maximizing system reliability. The authors formulate this as a bi-objective problem in which the goal is to maximize system reliability while minimizing the total cost of the system. The MOEA adopted in this case is the NSGA-II (Deb et al., 2002). The authors reported that the NSGA-II had a superior performance than traditional approaches reported in the specialized literature.

Meedeniya et al. (Meedeniya et al., 2011) proposed an approach to automate the optimal deployment of software components to hardware nodes. The main goal is that the reliabilities of individual services implemented at the software level are balanced, which is an issue when the hardware architecture was designed prior to the customized software architecture. The authors adopted Fonseca and Fleming's MOGA (Fonseca & Fleming, 1993) in their proposal and considered the automotive domain. The objectives considered correspond to system services (like the *Antilock Brake System* (ABS), the *Adaptive Cruise Control* (ACC) or the *Airbag* service) in their automotive case study.

4.4. Scheduling

Cui et al. (Cui et al., 2017) conduct a reliability analysis of cloud services by applying a Markov-based method. Then, they formulate the cloud scheduling problem as a multi-objective optimization problem with constraints in terms of reliability, makespan and flowtime. This problem is solved using a genetic algorithm-based chaotic ant swarm (GA-CAS) algorithm. The results show that the GA-CAS algorithm is able to speed up convergence and to outperform other metaheuristics in the problem tackled by the authors.

Ahn and Hur (Ahn & Hur, 2021) provide a mathematical model for cloud manufacturing. In cloud manufacturing, customers register customized requirements, and manufacturers provide appropriate services to complete the task. A cloud manufacturing manager establishes manufacturing schedules that determine the service provision time in a real-time manner as the requirements are registered in real time. In addition, customer satisfaction is affected by various measures such as cost, quality, tardiness, and reliability. So, the authors deal with a real-time and multi-objective task scheduling problem in which the aim is minimizing tardiness, cost, quality and reliability. This model is solved using a multi-objective genetic algorithm. The authors report that their proposed approach is effective and efficient.

Han et al. (Han et al., 2021) proposed a heuristic called *Cost and Makespan Scheduling of Workflows in the Cloud* (CMSWC) to solve the workflow scheduling problem. In this case, the objectives are to minimize the cost and the makespan (execution time) of workflows in cloud computing.

The proposed approach follows a two-phase list scheduling philosophy: ranking and mapping. CMSWC is really a variant of MOHEFT (Durillo et al., 2012), which adopts Shift-based Density Estimation (SDE), (Li et al., 2014) to weaken the density estimator of the multi-objective evolutionary algorithm with the aim of promoting convergence. The experimental results reported by the authors in real-life workflow applications, show that the proposed approach consistently produces solutions with better cost and makespan than those of state-of-the-art approaches in all cases.

5. Future areas of research

There is plenty of room for extending the use of MOEAs in reliability. The following are a few suggestions for possible paths for future research that may be worth exploring.

- *Use of Different Types of MOEAs:* the use of decomposition-based (Santiago et al., 2014) and indicator-based (Falcón-Cardona et al., 2020) MOEAs seems to be fairly limited in reliability. This may be due to the relatively low dimensionality (in objective space) of most of the problems that have been tackled in this area. However, the solution of many-objective problems using alternative types of MOEAs is still relatively rare in this area. There are some recent proposals which already tackle many-objective problems, see for example (Saeedi et al., 2020), but more work in this direction is expected in the next few years.
- *Incorporation of User's Preferences:* most MOEAs are commonly employed under the assumption that the entire Pareto optimal set is needed. However, in most practical applications, not all the solutions are required, since users normally identify regions of interest within the Pareto front and this could be the case in some problems related to reliability as done, for example in (Sinha, 2007). So, the incorporation of user's preferences in the search conducted by a MOEA is certainly an interesting research area within reliability that is worth exploring, see for example (Filatovas et al., 2017; Rachmawati & Srinivasan, 2006; Hu et al., 2017).
- *Use of Domain Knowledge:* the incorporation of knowledge may improve the performance of MOEAs adopted to solve complex problems. Such knowledge may be provided either *a priori* (when available) or can be extracted during the search (Landa Becerra et al., 2008; Liu,

2011). This knowledge may influence the operators of a MOEA in order to conduct a more efficient and/or effective search, or can be used to design heuristic procedures aimed to reduce the size of the search space.

6. Conclusion

In this chapter, we have seen some representative problems related to reliability in which the use of multi-objective optimization models and multi-objective evolutionary algorithms to solve them has shown several relevant advantages.

The use of multi-objective evolutionary algorithms in this area still has a lot of potential and many more applications are expected to occur within the next few years. Also, other mechanisms, which have been traditionally adopted in evolutionary multi-objective optimization, could bring additional benefits to this area as briefly discussed in the final part of the chapter.

Acknowledgment

The first author acknowledges support from CONACyT project no. 2016-01-1920 (*Investigación en Fronteras de la Ciencia 2016*) and from a project from the 2018 SEP-Cinvestav Fund (application no. 4).

References

Ahn, G. & Hur, S. 2021. Multiobjective real-time scheduling of tasks in cloud manufacturing with genetic algorithm. *Mathematical Problems in Engineering* 1305849.

Ardakan, M.A. & Rezvan, M.T. 2018. Multi-objective optimization of reliability-redundancy allocation problem with cold-standby strategy using NSGA-II. *Reliability Engineering & System Safety* 172, 225–238.

Bader, J. & Zitzler, E. 2011. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation* 19(1), 45–76.

Bandyopadhyay, S., Saha, S., Maulik, U. & Deb, K. 2008. A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation* 12(3), 269–283.

Beume, N., Fonseca, C.M., López-Ibáñez, M., Paquete, L. & Vahrenhold, J. 2009. On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation* 13(5), 1075–1082.

Beume, N., Naujoks, B. & Emmerich, M. 2007. SMS-EMOA: multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181(3), 1653–1669.

Bolchini, C., Lanzi, P.L. & Miele, A. 2010. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, IEEE Press, Barcelona, 419–426.

Branke, J. & Deb, K. 2005. Integrating user preferences into evolutionary multi-objective optimization. Y. Jin (Ed.). *Knowledge Incorporation in Evolutionary Computation*, Springer, Berlin Heidelberg, 461–477.

Brockhoff, D., Wagner, T. & Trautmann, H. 2012. On the properties of the R2 indicator. *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)* 465–472.

Brockhoff, D., Wagner, T. & Trautmann, H. R2 Indicator-Based Multiobjective Search. 2015. *Evolutionary Computation* 23(3), 369–395.

Claudio, M., Rocco, S., Ramirez-Marquez, J.E., Daniel, E. & Salazar, A. 2009. A multiple-objective approach for the vulnerability assessment of infrastructure networks. R. Bris et al. (Eds.). *Reliability, Risk, and Safety*. CRC Press, London.

Coello Coello, C.A. 1996. *An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design*, PhD thesis, Department of Computer Science, Tulane University, New Orleans.

Coello Coello, C.A. 2000. Treating constraints as objectives for single-objective evolutionary optimization. *Engineering Optimization* 32(3), 275–308.

Coello Coello, C.A. 2006. The EMOO repository: a resource for doing research in evolutionary multiobjective optimization. *IEEE Computational Intelligence Magazine* 1(1), 37–45.

Coello Coello, C.A., Lamont, G.B. & Van Veldhuizen, D.A. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, 2nd edition.

Cox, W. & While, L. 2016. Improving the IWFG algorithm for calculating incremental hypervolume. *2016 IEEE Congress on Evolutionary Computation (CEC'2016)*, IEEE Press, Vancouver.

Cui, H., Li, Y., Liu, X., Ansari, N. & Yunjie, L. 2017. Cloud service reliability modelling and optimal task scheduling. *IET Communications* 11(2), 161–167.

Das, I. & Dennis, J. 1997. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization* 14(1), 63–69.

Das, I. & Dennis, J.E. 1998. Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization* 8(3), 631–657.

Deb, K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester.

Deb, K. & Jain, H. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Transactions on Evolutionary Computation* 18(4), 577–601.

Deb, K., Gupta, S., Daum, D., Branke, J., Mall, A.K. & Padmanabhan, D. 2009. Reliability-based optimization using evolutionary algorithms. *IEEE Transactions on Evolutionary Computation* 13(5), 1054–1074.

Deb, K., Jain, P., Gupta, N.K. & Maji, H.K. 2004. Multiobjective placement of electronic components using evolutionary algorithms. *IEEE Transactions on Components and Packaging Technologies* 27(3), 480–492.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197.

Durillo, J.J., Fard, H.M. & Prodan, R. 2012. MOHEFT: A multi-objective list-based method for workflow scheduling. *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*. IEEE Computer Society, Taipei, 185–192.

Ehrgott, M. 2005. *Multicriteria Optimization*. Springer, Berlin, 2nd edition.

Emmerich, M., Beume, N. & Naujoks, B. An EMO algorithm using the hypervolume measure as selection criterion. C.A. Coello Coello et al. (Eds.). *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, Springer. *Lecture Notes in Computer Science* 3410, 62–76.

Falcón-Cardona, J.G., Liefooghe, A. & Coello Coello, C.A. 2020. An ensemble indicator-based density estimator for evolutionary multi-objective optimization. T. Bäck et al. (Eds.), *Parallel Problem Solving from Nature – PPSN XVI, 16th International Conference, PPSN 2020. Proceedings, Part II*, Springer. *Lecture Notes in Computer Science* 12270, 201–214.

Filatovas, E., Lancinskas, A., Kurasova, O. & Zilinskas, J. 2017. A preference-based multi-objective evolutionary algorithm R-NSGA-II with stochastic local search. *Central European Journal of Operations Research* 25(4), 859–878.

Fleischer, M. 2003. The measure of Pareto optima. Applications to multi-objective metaheuristics. C.M. Fonseca et al. (Eds.), *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, Springer. *Lecture Notes in Computer Science* 2632, 519–533.

Fonseca, C.M. & Fleming, P.J. 1993. Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. S. Forrest (Ed.). *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, San Mateo, 416–423.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Goldberg, D.E. & Richardson, J. 1987. Genetic algorithm with sharing for multimodal function optimization. J.J. Grefenstette (Ed.). *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*. Lawrence Erlbaum, Hillsdale, New Jersey, 41–49.

Guerreiro, A.P. & Fonseca, C.M. 2018. Computing and updating hypervolume contributions in up to four dimensions. *IEEE Transactions on Evolutionary Computation* 22(3), 449–463.

Hajela, P. & Lin, C.Y. 1992. Genetic search strategies in multicriterion optimal design. *Structural Optimization* 4, 9–107.

Han, P., Du, C., Chen, J., Ling, F. & Du, X. 2021. Cost and makespan scheduling of workflows in clouds using list multiobjective optimization technique. *Journal of Systems Architecture* 112, 101837.

Hernandez Gómez, R. & Coello Coello, C.A. 2015. Improved metaheuristic based on the R2 indicator for many-objective optimization. *2015 Genetic and Evolutionary Computation Conference (GECCO 2015)*. ACM Press, Madrid, 679–686.

Hernández Gómez, R., Coello Coello, C.A. & Alba Torres, E. 2016. A multi-objective evolutionary algorithm based on parallel coordinates. *2016 Genetic and Evolutionary Computation Conference (GECCO'2016)*. ACM Press, Denver, 565–572.

Hu, J., Yu, G., Zheng, J. & Zou, J. 2017. A preference-based multi-objective evolutionary algorithm using preference selection radius. *Soft Computing* 21(17), 5025–5051.

Hwang, C.L., Tillman, F.A., Wei, W.K. & Lie, C.H. 1979. Optimal scheduled-maintenance policy based on multiple-criteria decision-making. *IEEE Transactions on Reliability* 28(5), 394–399.

Igel, C., Hansen, N. & Roth, S. 2007. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation* 15(1), 1–28.

Inagaki, T., Inoue, K. & Akashi, H. 1978. Interactive optimization of system reliability under multiple objectives. *IEEE Transactions on Reliability* 27(4), 264–267.

Ishibuchi, H., Masuda, H., Tanigaki, Y. & Nojima, Y. 2015. Modified distance calculation in generational distance and inverted generational distance. A. Gaspar-Cunha et al. (Eds.). *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015*, Springer. *Lecture Notes in Computer Science* 9019, 110–125.

Ishibuchi, H., Setoguchi, Y., Masuda, H. & Nojima, Y. 2017. Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation* 21(2), 169–190.

Jaszkiewicz, A. 2018. Improved quick hypervolume algorithm. *Computers & Operations Research* 90, 72–83.

Kim, J.R. & Gen, M. 1999. Genetic algorithm for solving bicriteria network topology design problem. *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99)*. IEEE Press, Washington, DC, 2272–2279.

Knowles, J. & Corne, D. 2003. Properties of an adaptive archiving algorithm for storing nondominated vectors. *IEEE Transactions on Evolutionary Computation* 7(2), 100–116.

Kukkonen, S. & Deb, K. 2006. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. IEEE Press, Vancouver, BC, 1164–1171.

Kumar, R., Parida, P.P. & Gupta, M. 2002. Topological design of communication networks using multiobjective genetic optimization. *2002 IEEE Congress on Evolutionary Computation (CEC'2002)*. IEEE Press, Piscataway, 1, 425–430.

Kumar, R. & Rockett, P. 1998. Multiobjective genetic algorithm partitioning for hierarchical learning of high-dimensional pattern spaces: a learning-follows-decomposition strategy. *IEEE Transactions on Neural Networks* 9(5), 822–830.

Lacour, R., Klamroth, K. & Fonseca, C.M. 2017. A box decomposition algorithm to compute the hypervolume indicator. *Computers & Operations Research* 79, 347–360.

Landa Becerra, R., Santana-Quintero, L.V. & Coello Coello, C.A. 2008. Knowledge incorporation in multi-Objective evolutionary algorithms. A. Ghosh et al. (Eds.), *Multi-objective Evolutionary Algorithms for Knowledge Discovery from Data Bases*. Springer, Berlin, 23–46.

Li, F., Cheng, R., Liu, J. & Jin, Y. 2018. A two-stage R2 indicator based evolutionary algorithm for many-objective optimization. *Applied Soft Computing* 67, 245–260.

Li, M., Yang, S. & Liu, X. 2014. Shift-based density estimation for Pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 18(3), 348–365.

Liu, D. *Multi-Objective Cultural Algorithms*. PhD thesis, Wayne State University, Detroit, 2011.

Manoatl Lopez, E. & Coello Coello, C.A. 2016. IGD^+-EMOA: a multi-objective evolutionary algorithm based on IGD^+. *2016 IEEE Congress on Evolutionary Computation (CEC'2016).* IEEE Press, Vancouver, 999–1006.

Manoatl Lopez, E. & Coello Coello, C.A. 2018. An improved version of a reference-based multi-objective evolutionary algorithm based on IGD+. *2018 Genetic and Evolutionary Computation Conference (GECCO'2018).* ACM Press, Kyoto, 713–720.

Marseguerra, M., Zio, E., Podofillini, L. & Coit, D.W. 2005. Optimal design of reliable network systems in presence of uncertainty. *IEEE Transactions on Reliability* 54(2), 243–253.

Meedeniya, I., Buhnova, B., Aleti, A. & Grunske, L. 2011. Reliability-driven deployment optimization for embedded systems. *Journal of Systems and Software* 84(5), 835–846.

Menchaca-Mendez, A. & Coello Coello, C.A. 2017. An alternative hypervolume-based selection mechanism for multi-objective evolutionary algorithms. *Soft Computing* 21(4), 861–884.

Miettinen, K.M. 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston.

Phan, D.H. & Junichi, S. 2011. Boosting indicator-based selection operators for evolutionary multiobjective optimization algorithms. *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence (ICTAI'2011)*, IEEE Press, Boca Raton, 276–281.

Pires, E.J. Solteiro, M., Tenreiro, J.A. & de Moura Oliveira, P.B. 2013. Entropy diversity in multi-objective particle swarm optimization. *Entropy* 15(12), 5475–5491.

Rachmawati, L. & Srinivasan, D. Preference incorporation in multi-objective evolutionary algorithms: a survey. 2006. *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*. IEEE Press, Vancouver, BC, 3385–3391.

Rosenberg, R.S. 1967. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Arbor, Michigan.

Rudolph, G. & Agapie, A. 2000. Convergence properties of some multi-objective evolutionary algorithms. *Proceedings of the 2000 Conference on Evolutionary Computation (CEC'2000)*. IEEE Press, Piscataway, 2, 1010–1016.

Russo, L.M.S. & Francisco, A.P. 2016. Extending quick hypervolume. *Journal of Heuristics* 22(3), 245–271.

Saeedi, S., Khorsand, R., Bigdoli, S.G. & Ramezanpour, M. 2020. Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing. *Computers & Industrial Engineering* 147, 106649.

Santiago, A., Huacuja, H.J.F., Dorronsoro, B., Pecero, J.E., Santillan, C.G., Barbosa, J.J.G. & Monterrubio, J.C.S. 2014. A survey of decomposition methods for multi-objective optimization. O. Castillo et al. (Eds.). *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*. Springer, 453–465.

Schaffer, J.D. 1984. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Tennessee.

Schaffer, J.D. 1985. Multiple objective optimization with vector evaluated genetic algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum, Hillsdale, 93–100.

Schaffer, J.D. & Grefenstette, J.J. 1985. Multi-objective learning via genetic algorithms. *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI-85)*. AAAI, Los Angeles, 593–595.

Sinha, K. 2007. Reliability-based multiobjective optimization for automotive crashworthiness and occupant safety. *Structural and Multidisciplinary Optimization* 33(3), 255–268.

Srinivas, N. & Deb, K. 1994. Multiobjective optimization using nondominated Sorting in Genetic Algorithms. *Evolutionary Computation* 2(3), 221–248.

Taboada, H.A., Espiritu, J.F. & Coit, D.W. 2008. MOMS-GA: A multi-objective multi-state genetic algorithm for system reliability optimization design problems. *IEEE Transactions on Reliability* 57(1), 182–191.

Talbi, El-Ghazali. 2009. *Metaheuristics. From Design to Implementation*. John Wiley & Sons, Inc., Hoboken.

Tian, Y., Zhang, X., Cheng, R. & Jin, Y. 2016. A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. *2016 IEEE Congress on Evolutionary Computation (CEC'2016)*. IEEE Press, Vancouver, 5222–5229.

Zafiropoulos, E.P. & Dialynas, E.N. 2004. Reliability and cost optimization of electronic devices considering the component failure rate uncertainty. *Reliability Engineering and System Safety* 84(3), 271–284.

Zhang, C., Ramirez-Marquez, J.E. & Sanseverino, C.M. 2011. a holistic method for reliability performance assessment and critical components detection in complex networks. *IIE Transactions* 43(9), 661–675.

Zhang, Q. & Li, H. 2007. MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731.

Zitzler, E. 1999. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich.

Zitzler, E. & Künzli, S. 2004. Indicator-based Selection in multiobjective search. Xin Yao et al. (Eds.). *Parallel Problem Solving from Nature – PPSN VIII*, *Lecture Notes in Computer Science*. Springer-Verlag. Birmingham, 3242, 832–842.

Zitzler, E. & Thiele, L. 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation* 3(4), 257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M. & Grunert da Fonseca, V. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7(2), 117–132.