

Evolutionary Many-Objective Optimization: A Quick-Start Guide

Shelvin Chand¹, Markus Wagner²

University of New South Wales, Canberra, Australia¹

University of Adelaide, Adelaide, Australia²

Abstract

Multi-objective optimization problems having more than three objectives are referred to as many-objective optimization problems. Many-objective optimization brings with it a number of challenges that must be addressed, which highlights the need for new and better algorithms that can efficiently handle the growing number of objectives. This article reviews the different challenges associated with many-objective optimization and the work that has been done in the recent-past to alleviate these difficulties. It also highlights how the existing methods and body of knowledge have been used to address the different real world many-objective problems. Finally, it brings focus to some future research opportunities that exist with many-objective optimization.

We report in this article what is commonly used, be it algorithms or test problems, so that the reader knows what are the benchmarks and also what other options are available. We deem this to be especially useful for new researchers and for researchers from other fields who wish to do some work in the area of many-objective optimization.

Keywords: Many-Objective Optimization, Review, Algorithms, Optimization Test Problems, Real World Applications, Theory, Progress, Challenges

1. Introduction

Multi-objective optimization refers to the simultaneous optimization of multiple conflicting objectives. It gives rise to a set of optimal solutions (known as

the Pareto-optimal solutions), instead of a single optimal solution [18]. None of the optimal solutions can claim to be better than any other with respect to all objective functions.

Surveys have highlighted this to be one of the fastest growing fields of research and application among all computational intelligence topics [12]. It is also a field of research that attracts interest from people of different backgrounds including mathematicians, computer scientists, economists and engineers [12].

Evolutionary multi-objective optimization (EMO) methods have shown to be highly successful in finding well-converged and well-diversified non-dominated solutions for optimization problems with two and three objectives[15]. Some of these successful methodologies include Strength Pareto Evolutionary Algorithm (SPEA) [87], SPEA2 [88], Non-dominated Sorting Genetic Algorithm (NSGA) [71], NSGA-II [18] and Pareto Archived Evolution Strategy (PAES) [43].

While all these methodologies have shown good success, it is important to consider that many real world problems have more than three objectives. Scalability tests for these methodologies highlight a number of problems relating to convergence, diversity and computation time [42]. As a result, it is important to come up with new methodologies or to improve existing ones to be able to deal with a higher number of objectives. Multi-objective problems having more than three objectives are referred to as many-objective optimization problems [65, 63]. Many-objective optimization gives rise to a new set of challenges that must be addressed. It also opens doors for new research opportunities which can allow us to solve more complex real world problems.

While many-objective optimization is a fairly new area of research, it is important to take note that some work on this had already begun in the early 1990s. One of the earliest algorithms which has been applied to many-objective problems is MOGA [26]. MOGA was tested on the four objective Pegasus gas turbine engine optimization problem [26]. Since then a number of researchers have attempted to solve different real world and simulated many-objective optimization problems. Majority of the work in this area has taken place within the last decade.

This short and compact review represents an update over existing surveys on this topic, such as the ones done by Wagner et al. [77] (2007, 22 references in total), Ishibuchi et al. [38] (2008, 55 references in total) and the recent one by von Luecken et al. [75] (2014, 112 references in total). We extend those by putting over 60 new articles into the context of many-objective optimization. We highlight some of the current challenges and bring focus to the work that has been done to address these difficulties. We also identify a combination of old and recently developed methods which have shown success with many-objective optimization. There also exists quite a bit of literature on application research. We highlight some of the recent application research done in this area. We conclude by bringing focus to some of the future research opportunities that exist with many-objective optimization.

2. Definitions & Basic Principles

Without loss of generality, a simple multi-objective problem¹ can be formulated as:

$$\begin{aligned} \min \quad & F(x) = (f_1(x), f_2(x), \dots, f_m(x))^T \\ & x \in X \subset \mathbb{R}^n \end{aligned} \tag{1}$$

where $x = (x_1, \dots, x_n)$ is a vector of n decision variables and X is an n -dimensional decision space. m is the number of objectives to be optimized. When $m \geq 4$, the problem becomes a many-objective problem.

In the context of multi-objective optimization, the optimal solutions are also referred to as non-dominated solutions. In a minimization problem, a solution x is considered non-dominated in comparison to another solution x^* when no objective value of x^* is less than x and at least one objective value of x^* is greater than x [71].

¹As above-mentioned, many-objective problems are multi-objective ones with more than three objectives. Thus, the definitions here hold for many-objective problems as well.

Convergence and diversity are the main goals of any multi-objective optimization algorithm. Convergence refers to finding a set of solutions that lie on or close to the true Pareto-optimal front [12]. Diversity refers to finding a set of solutions which are diverse enough to represent the entire range of the Pareto-optimal front [12].

To measure the performance of EMO algorithms, a number of quality indicators have been proposed over the years. Some of the most widely used quality indicators are the inverted generational distance (IGD), hypervolume and the R2 indicator. IGD measures the average distance for all members in the true Pareto-optimal set to their nearest solutions in the obtained solution set (opposite of generational distance (GD)) [67]. The hypervolume of a set of solutions measures the size of the portion of objective space that is dominated by those solutions collectively [82]. The IGD and the hypervolume can be used to measure both the spread of the solutions and convergence to the Pareto-front. The family of R-indicators (R1, R2, R3) can be used to assess and compare Pareto set approximations on the basis of a set of utility functions [89]. In particular the R2-indicator [10] was explored recently because it is weakly monotonic and computationally efficient [22, 11].

A list of other quality indicators are given in Table 1.²

3. Challenges

3.1. *Non-dominated Population*

Most of the EMO algorithms use the concept of Pareto Dominance in order to compare and identify the best solutions [28]. An increase in the number of objectives causes a large portion of a randomly generated population to become non-dominated [28]. Having a population which is largely composed of non-dominated solutions does not give room for creating new solutions in every generation [15, 38]. This slows down the overall search process.

²This is a summary and extension of Table 1 from [49].

Table 1: Quality indicators for multi-objective optimization.

Quality Indicator	Measures [49]
Hypervolume [82]	Convergence, Spread & Uniformity
IGD [67], IGD_p [69]	Convergence, Spread & Uniformity
Hyperarea Ratio[74]	Convergence, Spread & Uniformity
Averaged Hausdorff Distance [69]	Convergence, Spread & Uniformity
G -Metric [50]	Convergence, Spread & Uniformity
Diversity Comparison Indicator (DCI) [49]	Spread & Uniformity
Diversity Measure [16]	Spread & Uniformity
R2 indicator [10]	Spread & Uniformity
Sigma Diversity Metric [56]	Spread & Uniformity
GD [67], GD_p [69]	Convergence
Convergence Measure [47]	Convergence
Error Ratio [74]	Convergence

Some research has been done on tackling this problem and finding alternatives to the Pareto dominance approach. Sato et al. [68] proposed a novel multi-objective evolutionary algorithm that uses Pareto partial dominance. It calculates dominance between solutions using a subset of the objectives which are switched after a fixed number of generations. Their approach was able to give better convergence in comparison to conventional NSGA-II for the many-objective 0/1 knapsack problem. Aguirre and Tanaka [2] proposed a method to search on many-objective problems by partitioning the objective space into sub-spaces and performing one generation of the evolutionary search in each sub-space. Their method showed good performance on the MNK Landscapes with 4 to 10 objectives.

The ϵ -domination principle [46, 15] which is used for approximating the Pareto-front can also be used to address the problem of a large non-dominated set [32]. The use of this principle will make all points within an ϵ -distance from a set of Pareto-optimal points ϵ -dominated. This process will allow for the generation of a finite number of Pareto-optimal points as the target [15]. It will also allow for a more diverse set of solutions. Algorithms based on the ϵ -domination principle include the ϵ -MOEA [20], ϵ -NSGA-II [45], Borg-MOEA [33] and AGE-II [76]. Other domination principles such as fuzzy-dominance [79] can also be used to overcome this problem.

A related problem with high-dimensional Pareto fronts is that larger populations are needed to adequately represent those. Intuitively, it might make sense to increase the population size exponentially with the number of objectives, however, this is not always necessary. For example, Papadimitriou and Yannakakis [61] and Erlebach et al. [24] have shown that under certain assumptions there is always an approximate Pareto set whose size is just polynomial in the length of the encoded input. This can be achieved by placing a hyper-grid in the objective space using the coordinates $1, (1 + \epsilon), (1 + \epsilon)^2, \dots$ for each objective. Since it suffices to have one representative solution in each grid cell and to have only non-dominated cells occupied, it can be seen that for any finite ϵ and any set F with bounded vectors f , i.e. $1 \leq f_i \leq K$ for all $i \in \{1, \dots, m\}$,

there exists a set F_ϵ containing $|F_\epsilon| \leq \left(\frac{\log K}{\log(1+\epsilon)}\right)^{m-1}$ vectors. The proof outline is as follows. First, the objective space is divided into $\left(\frac{\log K}{\log(1+\epsilon)}\right)^m$ boxes, and from each box at most one point can be in the approximation at the same time. Second, there are $\left(\frac{\log K}{\log(1+\epsilon)}\right)^{m-1}$ equivalence classes of boxes where (without loss of generality) in each class the boxes have the same coordinates in all but one dimension. In each class $\frac{\log K}{\log(1+\epsilon)}$ different boxes constitute a chain of dominating boxes. Hence, only one point from each of these classes can be a member of the approximation at the same time. For detailed proof and for more details on this regarding evolutionary multi-objective and many-objective optimization, we refer the interested reader to the article by Laumanns et al. [46].

Another solution would be to focus on finding a partial front corresponding to the preference information given by users [12]. This would also save unnecessary computation time which is spent on finding solutions that are not of interest to the user. A range of different modifications to existing algorithms have been proposed in the past to incorporate user preference. The techniques range from reference points (e.g., [21, 55]) to weights in the objective space (e.g., [59, 27]).

3.2. Computational Efficiency

Another major problem is that many-objective optimization algorithms tend to be computationally expensive. Some performance matrices such as the hypervolume require exponentially more computation power when the number of objectives increase, since high-dimensional points are being compared against one another [15]. To improve on this, faster, accurate and also approximate algorithms for the computation of the hypervolume have been suggested over the years [5, 82, 7].

Diversity measures such as the crowding distance also become computationally expensive when identifying neighbors in a high-dimensional space [15]. To address this, Deb and Jain [15] suggested the use of a reference point based approach in which points corresponding to a set of well-spread reference points can be emphasized to find a set of widely distributed Pareto-optimal points. This approach can also alleviate the problem of a large non-dominated set. The

ϵ -domination principle can also be used here to achieve a set of well spread solutions since its computational complexity increases only linearly with the number of dimensions.

3.3. Recombination

Recombination operators play an important part in EMO. Problems arise when only a handful of solutions are to be found in a large dimensional space as the solutions are more likely to be widely distant from each other [15], if the EMO algorithm focuses on extreme points. Special recombination operators may be necessary for handling this problem. To alleviate this problem, Deb and Jain [15] suggested the use of a special recombination scheme in which near-parent solutions are emphasized. An example of this is the simulated binary crossover (SBX) [13] with a large distribution index. Careful consideration must be put into the selection of the operator and the participating parents.

3.4. Visualization

Visualization is an important part of optimization and decision making. As the number of objectives increase beyond three it becomes difficult for researchers to be able to visualize the objective space. One possible way around this would be to visually alternate between the given objectives. For example, a user can start off by choosing a set of three objectives and view the distribution using a three-dimensional graph. The user can then switch between objectives and view the different trade-offs while giving emphasis to three objectives at one time. This will also be an effective approach in preventing the user from getting overwhelmed with the large number of objectives. Objective reduction methods [17] can also be helpful.

In terms of actual graphing methods, the parallel coordinates plot [37] has proven to be a popular choice in many-objective problem solving research [25]. The parallel coordinates plot is a two-dimensional plot with the x-axis representing the objectives and the y-axis representing the solution values. The geometrical features of a surface in n-dimensional space are preserved in the

parallel coordinates system, however, this approach also has a few shortcomings [25].

Another option is to use heatmaps [62]. In a heatmap, the columns represent objectives and rows represent the solutions with the relative values as heat represented by color [78]. Some researchers [78] are utilizing radial visualization [34] which essentially displays n-dimensional data in an elliptical or circular fashion. Sophisticated methods, such as decision maps [53] or geodesic maps [72] can be helpful as well.

Ultimately, more effort should be put into developing better, more intuitive, graphing and visualization tools to assist researchers and decision makers.

4. Many-Objective Optimization Methods

Different approaches have been explored for many-objective optimization. Some have shown more success than others. Some of these methods are highlighted in this section. It must be noted that some of these were initially designed and tested on two and three objective problems only, but over the years they have been included in various studies for many-objective optimization as well. As a result, we feel that it is necessary to highlight such methods as well. It must also be noted that this list is by no means exhaustive as there are countless methods and extensions which have been developed for many-objective optimization.

4.1. Methods

One of the most popular algorithm in literature is the NSGA-II [18]. It is often used as a baseline algorithm for comparison with new algorithms. The NSGA-II is a computationally fast and elitist MOEA based on a non-dominated sorting approach. It also uses an explicit diversity-preserving mechanism to obtain a set of well-spread Pareto-optimal solutions. The NSGA-II was initially tested on problems with smaller number of objectives, but over the years it has shown to be successful in solving problems with many objectives as well.

A number of NSGA-II improvements have been proposed over the years to make the algorithm more efficient in handling a larger number of objectives. ϵ -NSGA-II [45] combines NSGA-II with an ϵ -dominance archive, adaptive population sizing and time continuation [32]. This algorithm has also been used for many different real world many-objective problems [44, 45].

ϵ -MOEA is a steady-state MOEA that exploits the benefits of an ϵ -dominance archive [20, 32]. The archive contains the ϵ -non-dominated individuals from the main population. The algorithm divides the objective space into a number of hyper-boxes. Diversity is maintained within the archive by allowing only one solution to be present in each hyperbox. The ϵ -dominance procedure helps in reducing the cardinality of the Pareto-optimal region, which makes the ϵ -MOEA useful for many-objective optimization.

Hadka and Reed [32, 33] proposed Borg MOEA which is an algorithm designed for handling many-objective, multimodal problems using an auto-adaptive multi-operator recombination operator which is able to improve search in many different problem domains. It uses many different recombination operators such as the SBX, parent-centric crossover (PCX), simplex crossover (SPX), polynomial mutation (PM) and many others. This allows the algorithm to quickly adapt to the problem’s local characteristics and adjust as required.

Bringmann et al. [8] proposed an algorithm that works with a formal notion of approximation. The proposed algorithm, called Approximation-Guided Evolution (AGE), outperformed state-of-the-art approaches in terms of the desired additive approximation and the covered hypervolume on standard benchmark functions (with many objectives) given a fixed time budget. Despite the good performance on problems with many objectives, AGE was not very efficient for problems with few objectives. Another major issue with AGE was that it stored all non-dominated points seen so far in an archive which greatly affected its runtime.

Wagner and Neumann [76] addressed these problems by proposing a fast approximation-guided evolutionary algorithm called AGE-II. AGE-II approximates the archive in a meticulous manner which controls its size and its influence

on the runtime. This is achieved through the concept of ϵ -dominance. The algorithm once again outperformed current state-of-the-art algorithms in terms of the desired additive approximation for standard many-objective benchmark functions with up to 20 objectives. It also performed well for problems with fewer objectives.

HypE (Hypervolume Estimation Algorithm) [5] is a fast hypervolume indicator-based multi-objective evolutionary algorithm that can be applied to problems with arbitrary number of objective functions. It uses Monte Carlo simulation to approximate the exact hypervolume values. It was tested on a range of many-objective optimization problems and managed to give a competitive set of results in comparison to NSGA-II [18], SPEA2 [88] and IBEA.

MOEA/D [85] is another algorithm which has shown success with many-objective optimization. It uses a decomposition method to decompose the given problem into a number of scalar optimization problems. These sub-problems are then simultaneously optimized using an evolutionary algorithm. The MOEA/D has been used for comparison in various recent studies [15, 33], making it a benchmark algorithm for many-objective optimization.

Giagkiozis et al. [31] designed an improved decomposition method that uses the so-called cross entropy method [66] as an estimation of distribution algorithm due to its theoretical properties. The resulting optimization framework MACE-gD was found to provide the best distributions of solutions on the Pareto front if the geometry of the Pareto front was known a priori.

Asafuddoula et al. [3, 4] proposed a decomposition-based evolutionary algorithm with adaptive epsilon comparison (DBEA-Eps). The algorithm is designed using a steady state form and utilizes reference directions to guide the search. The balance between diversity and convergence is maintained using an adaptive epsilon comparison.

Related to these previous two decomposition-based algorithms are the theoretical studies by Giagkiozis and Fleming [30]. There, the authors investigate whether there is an advantage of using a decomposition-based method over Pareto-based methods. Among others, the results are that (1) Pareto-

dominance methods and the Chebyshev scalarizing function are equivalent, and that (2) optimal scalarizing functions can be found if the geometry of the Pareto front is known in advance.

Deb and Jain [15] recently proposed the NSGA-III which uses a reference point based approach for many-objective optimization. The number of reference points is similar to the population size ensuring that each population member is associated with a reference point. This ensures diversity as the reference points are uniformly distributed across the normalized hyper-plane. The method, which has been developed specifically for many-objective optimization, showed superior performance in comparison to methods such as the MOEA/D [85] and NSGA-II [18]. The performance scaling to 15 objectives was achieved mainly due to the aid in diversity preservation by supplying a set of well-distributed reference points.

A number of NSGA-III extensions and improvements have also been proposed. Jain and Deb [40] proposed an adaptive NSGA-III (A-NSGA-III) approach that was able to adaptively add and delete reference points, depending on the crowding of population members on different parts of the current non-dominated front. Keeping in mind the limitations of A-NSGA-III highlighted in [39], the same authors presented an efficiently adaptive NSGA-III procedure (A²-NSGA-III) with improvements in the reference point relocation strategy. Yuan et al. [84] introduced θ -NSGA-III, which uses a new preference relation, θ -dominance, to achieve balance between convergence and diversity for many-objective optimization.

Another important extension is the U-NSGA-III [70] which is a unified evolutionary algorithm capable of solving single/multi/many-objective problems. The U-NSGA-III was able to give comparable and sometimes better performance in comparison to a real-coded genetic algorithm (for single-objective problems), NSGA-II (for bi-objective problems) and NSGA-III (for many-objective problems).

Wang et al. [80] introduced Two_Arch2 which is an improved two archive algorithm for many-objective optimization. The two archives are named Con-

vergence Archive (CA) and Diversity Archive (DA). The $I_{\epsilon+}$ indicator is used as the selection principle for CA to improve convergence for many-objective problems while Pareto-dominance is used as the selection principle for DA to promote diversity. They also designed a new Lp-norm-based ($p < 1$) diversity maintenance scheme. The proposed method gave competitive performance in comparison to other many-objective optimization algorithms.

A knee point driven evolutionary algorithm (KnEA) [86] has been proposed recently in which the knee points among the non-dominated solutions in the current population are given preference for mating selection and environmental selection. The authors show that preference of knee points can approximately be seen as a bias towards larger hypervolume which assists in achieving good convergence and diversity.

4.2. Comparative Analysis

With so many algorithms available, one may ask the question as to which algorithm is the best. A number of comparative studies have been conducted for state-of-the-art algorithms on many-objective optimization problems [32, 47]. Li et al. [47] compared eight state-of-the-art EMO algorithms on a wide range of continuous and combinatorial many-objective optimization problems. They concluded that there is no clear performance gap between algorithms for all the tested problems. Instead, the authors suggested that the choice of algorithm depends on the problem at hand as certain algorithms are better suited for certain types of problems. Hadka and Reed [32] compared nine state-of-the-art algorithms which included Borg MOEA, ϵ -MOEA, ϵ -NSGA-II, MOEA/D and many others. Their results showed Borg to have a dominant performance on higher dimensional problem instances. Lastly, Wang et al. [81] performed a comprehensive and systematic comparison of six algorithms that included five different classes of MOEAs: the preference-inspired co-evolutionary algorithm called PICEA-g, the Pareto-dominance-based algorithm NSGA-II, the ϵ -dominance-based algorithm ϵ -MOEA, the scalarizing function-based algorithm MOEA/D, and the indicator-based algorithm HypE. Their observation was that PICEA-g

and HypE typically outperformed the others when the number of objectives increased.

As so often, the development and wide use of collections of tools and problems has greatly helped over the last years not only to increase the reproducibility, but also to simply provide reference implementations to the wider community. In addition to this, the publication of an article together with its code also means that the bar has been raised officially, and that comparisons with new methods *can and should* be performed.³ As of this writing, several frameworks such as jMetal [23], the MOEA Framework [1], and the Shark library [36] contain recently developed algorithms. In addition, some algorithm designers maintain project websites where source code and raw results are made available for everybody.

On a related note: researchers should not stop at publishing code, but they should also thoroughly analyze code published by others. Brockhoff [9] recently discovered a bug that negatively influences the performance in the popular EMO algorithm IBEA. This bug has been discovered 10 years after the publication of the original algorithm, and has been spread to many frameworks. Algorithm engineers, especially those of heuristic and randomized algorithms, do not always verify that the implementation is correct with respect to its specification. If testing is done, then it is often stopped when improvements over state-of-the-art approaches are achieved. In software engineering, activities such as comprehensive testing of an implementation and documentation are everyday activities. A more stringent programming and review process might have discovered this bug before it influenced the perception about IBEA's performance.

Coming back to the comparative analysis, multi-objective or many-objective test problems are typically static and without noise. For real-world problems, these assumptions do not necessarily hold, and consequently one might draw wrong conclusions from comparative studies. To close this gap between research

³The *should* is best practice in any discipline, however, the *can* is not always possible nowadays. This is to our surprise, since many EMO researchers work with software.

Table 2: DTLZ benchmark test problems [19] for many-objective optimization.

Problem	Properties [48]
DTLZ1	Linear, Multimodal
DTLZ2	Concave
DTLZ3	Concave, Multimodal
DTLZ4	Concave, Biased
DTLZ5	Concave, Degenerate
DTLZ6	Concave, Degenerate, Biased
DTLZ7	Mixed, Disconnected, Multimodal

and practice, Deb and Gupta [14] and Gaspar-Cunha et al. [29] investigated robustness as a feature and designed benchmark problems. However, it is to date not known whether the methods presented are suitable for many-objective problems as well.

5. Test Problems & Applications

There are many different many-objective optimization problems that researchers have tried to solve. The DTLZ (Table 2) and WFG (Table 3) test problems are continuous optimization problems and they are by far the most popular ones. They are sets of scalable test problems used to establish the strengths and weaknesses of many-objective algorithms. In addition, these problems contain decision variables that can be analyzed for convergence and diversity. Most of the algorithms mentioned in the previous section have been evaluated on these two sets of problems. Tutum and Deb [73] considered variations of the DTLZ functions, such as scaled objectives and the variations with constraints. Another popular choice is the many-objective 0/1 knapsack problem [57].

One of the common criticisms of these functions is that they are artificial, and rightly so. However, in their defence, the properties of these artificial problems vary significantly, and the reasoning is that the well-performing algorithms

Table 3: WFG benchmark test problems [35] for many-objective optimization.

Problem	Properties
WFG1	Convex, Mixed, Biased
WFG2	Convex, Disconnected, Multimodal
WFG3	Linear, Degenerate
WFG4	Concave, Multimodal
WFG5	Concave
WFG6	Concave
WFG7	Concave, Biased
WFG8	Concave, Biased
WFG9	Concave, Multimodal, Biased

are suitable for real-world problems. In addition, while these test problems (and others) are readily available via many optimization frameworks, real-world problems are typically not available. This is due to a variety of reasons that range from confidentiality of data, to technical problems, and to uneasiness of authors to publish code.

Despite these reasons, we can observe that benchmarking on real-world problems with ≥ 4 objectives is getting increasingly common—but it is still far from being widely spread. For example, the following articles consider problems from very different domains:

- Otake et al. [60] used NSGA-II to solve the many-objective simplified nurse scheduling problem, which is one of the few discrete many-objective optimization problem.
- Different many-objective algorithms have been used in the optimization of airfoil design [83, 51].
- NSGA-III [15] and DBEA-Eps [3] were tested on constraint based engineering design problems such as the 5-objective water resource manage-

ment problem and the 10-objective general aviation aircraft design problem.

- Jaimes et al. [52] made an attempt to solve the many-objective space trajectory design problem.
- Mkaouer et al. [54] applied NSGA-III to many-objective software refactoring.
- Narukawa and Rodemann [58] examined the performance of a few state-of-the-art algorithms including NSGA-II and MOEA/D on the hybrid car controller optimization problem with six objectives.
- Justesen and Ursem [41] investigated the novel approach of Many-Objective Distinct Candidates Optimization using Differential Evolution (MODCODE) on three centrifugal design problems having six to nine objectives.
- Bandyopadhyay and Mukherjee [6] compared different approaches (among them a differential evolution variant) on artificial test functions and also on the six-objective design of a factory-shed truss.

It appears that the methods and tools are finally available to tackle problems with many-objectives in the real world.

6. Future Opportunities

Many-objective optimization is a key research area for modern day evolutionary computation. This article highlights some key challenges that exist with relation to many-objective optimization and some recent work that has been done in trying to address these challenges.

The availability of a wide range of methods and tools to deal efficiently with many-objective problems allows us to investigate directions that were previously not accessible, which also includes problems in other domains. In some cases, the technology is not yet fully matured, and we see the following major opportunities for future research in the field of many-objective optimization:

- the development of better visualization tools to assist researchers and decision makers to navigate many-dimensional spaces,
- the integration of existing methods with multi-criteria decision making techniques to assist decision makers [64],
- the conduct of more comprehensive comparative studies that involve the recently developed algorithms, which is hopefully accompanied by an increased availability of code,
- and the utilization of existing and future methods to solve more complex real world problems, including dynamic and noisy problems.

The time has come to push this technology into more applications in different fields. Ultimately, we will be able to measure the success of many-objective evolutionary optimisation by the number of success stories written about it by others.

References

- [1] MOEA Framework - A Free and Open Source Java Framework for Multi-objective Optimization. <http://www.moeaframework.org/>.
- [2] H. Aguirre and K. Tanaka. Many-objective optimization by space partitioning and adaptive ϵ -ranking on mnk-landscapes. In *Evolutionary Multi-Criterion Optimization*, volume 5467 of *LNCS*, pages 407–422. Springer, 2009. ISBN 978-3-642-01019-4.
- [3] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition based evolutionary algorithm for many objective optimization with systematic sampling and adaptive epsilon control. In *Evolutionary Multi-Criterion Optimization*, volume 7811 of *LNCS*, pages 413–427. Springer, 2013. ISBN 978-3-642-37139-4.
- [4] M. Asafuddoula, T. Ray, and R. Sarker. A decomposition-based evolutionary algorithm for many objective optimization. *Evolutionary Computation, IEEE Transactions on*, 19(3):445–460, June 2015. ISSN 1089-778X.

- [5] J. Bader and E. Zitzler. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76, 2011. ISSN 1063-6560.
- [6] S. Bandyopadhyay and A. Mukherjee. An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution. *Evolutionary Computation, IEEE Transactions on*, 19(3):400–413, June 2015. ISSN 1089-778X.
- [7] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *Algorithms and Computation*, volume 5369 of *LNCS*, pages 436–447. Springer, 2008. ISBN 978-3-540-92181-3.
- [8] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner. Approximation-guided evolutionary multi-objective optimization. In *International Joint Conference on Artificial Intelligence (IJCAI) 2011*, pages 1198–1203. AAAI Press, 2011. ISBN 978-1-57735-514-4.
- [9] D. Brockhoff. A bug in the multiobjective optimizer ibea: Salutory lessons for code release and a performance re-assessment. In A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, editors, *Evolutionary Multi-Criterion Optimization*, volume 9018 of *LNCS*, pages 187–201. Springer, 2015. ISBN 978-3-319-15933-1.
- [10] D. Brockhoff, T. Wagner, and H. Trautmann. On the properties of the r2 indicator. In *Genetic and Evolutionary Computation Conference (GECCO) 2012*, pages 465–472. ACM, 2012. Publication status: Published.
- [11] D. Brockhoff, T. Wagner, and H. Trautmann. R2 indicator based multi-objective search. *Evolutionary Computation*, 0, 2014. Publication status: Published.
- [12] K. Deb. Recent developments in evolutionary multi-objective optimization. In *Trends in Multiple Criteria Decision Analysis*, volume 142 of *International Series in Operations Research & Management Science*, pages 339–368. Springer, 2010. ISBN 978-1-4419-5903-4.
- [13] K. Deb and R. B. Agrawal. Simulated binary crossover for continuous

- search space. *Complex Systems*, 9(2):115–148, 1995.
- [14] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evolutionary Computation*, 14(4):463–494, 2006.
- [15] K. Deb and H. Jain. 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, 2014. ISSN 1089-778X.
- [16] K. Deb and S. Jain. Running performance metrics for evolutionary multi-objective optimization. KanGAL report 2002004, Indian Institute of Technology, Kanpur, India, 2002.
- [17] K. Deb and D. K. Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. KanGAL report 2005011, Indian Institute of Technology, Kanpur, India, 2005.
- [18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. ISSN 1089-778X.
- [19] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *IEEE Congress on Evolutionary Computation (CEC) 2002*, volume 1, pages 825–830, 2002.
- [20] K. Deb, M. Mohan, and S. Mishra. A fast multi-objective evolutionary algorithm for finding well-spread pareto-optimal solutions. KanGAL report 2003002, Indian Institute of Technology, Kanpur, India, 2003.
- [21] K. Deb, J. Sundar, U. B. R. N, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. In *International Journal of Computational Intelligence Research*, pages 273–286, 2006.
- [22] A. Diaz-Manriquez, G. Toscano-Pulido, C. Coello, and R. Landa-Becerra. A ranking method based on the R2 indicator for many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC) 2013*, pages 1523–1530, 2013.

- [23] J. J. Durillo and A. J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011. ISSN 0965-9978.
- [24] T. Erlebach, H. Kellerer, and U. Pferschy. Approximating multi-objective knapsack problems. In *Algorithms and Data Structures*, volume 2125 of *LNCS*, pages 210–221. Springer, 2001. ISBN 978-3-540-42423-9.
- [25] P. Fleming, R. Purshouse, and R. Lygoe. Many-objective optimization: An engineering design perspective. In *Evolutionary Multi-Criterion Optimization*, volume 3410 of *LNCS*, pages 14–32. Springer, 2005. ISBN 978-3-540-24983-2.
- [26] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1-55860-299-2.
- [27] T. Friedrich, T. Kroeger, and F. Neumann. Weighted preferences in evolutionary multi-objective optimization. In *Advances in Artificial Intelligence (AI) 2011*, volume 7106 of *LNCS*, pages 291–300. Springer, 2011. ISBN 978-3-642-25831-2.
- [28] M. Garza-Fabre, G. Pulido, and C. Coello. Ranking methods for many-objective optimization. In *MICAI 2009: Advances in Artificial Intelligence*, volume 5845 of *LNCS*, pages 633–645. Springer, 2009. ISBN 978-3-642-05257-6.
- [29] A. Gaspar-Cunha, J. Ferreira, and G. Recio. Evolutionary robustness analysis for multi-objective optimization: benchmark problems. *Structural and Multidisciplinary Optimization*, 49(5):771–793, 2014. ISSN 1615-147X.
- [30] I. Giagkiozis and P. Fleming. Methods for multi-objective optimization: An analysis. *Information Sciences*, 293(0):338 – 350, 2015. ISSN 0020-0255. doi: <http://dx.doi.org/10.1016/j.ins.2014.08.071>. URL <http://www.sciencedirect.com/science/article/pii/S0020025514009074>.
- [31] I. Giagkiozis, R. Purshouse, and P. Fleming. Generalized decomposition

- and cross entropy methods for many-objective optimization. *Information Sciences*, 282(0):363 – 387, 2014. ISSN 0020-0255. doi: <http://dx.doi.org/10.1016/j.ins.2014.05.045>. URL <http://www.sciencedirect.com/science/article/pii/S0020025514006057>.
- [32] D. Hadka and P. Reed. Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization. *Evolutionary Computation*, 20(3):423–452, 2012. ISSN 1063-6560.
- [33] D. Hadka and P. Reed. Borg: An auto-adaptive many-objective evolutionary computing framework. *Evolutionary Computation*, 21(2):231–259, 2013.
- [34] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. Dna visual and analytic data mining. In *Visualization '97., Proceedings*, pages 437–441, Oct 1997.
- [35] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006. ISSN 1089-778X.
- [36] C. Igel, V. Heidrich-Meisner, and T. Glasmachers. Shark. *Machine Learning Research*, 9:993–996, 2008.
- [37] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *IEEE Conference on Visualization 1990*, pages 361–378, 1990.
- [38] H. Ishibuchi, N. Tsukamoto, and Y. Nojima. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2419–2426, June 2008.
- [39] H. Jain and K. Deb. An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 7811 of *LNCS*, pages 307–321. Springer, 2013. ISBN 978-3-642-37139-4.
- [40] H. Jain and K. Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part

- II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, 2014. ISSN 1089-778X.
- [41] P. Justesen and R. Ursem. Many-objective distinct candidates optimization using differential evolution on centrifugal pump design problems. In *IEEE Congress on Evolutionary Computation (CEC) 2010*, pages 1–8, 2010.
- [42] V. Khare, X. Yao, and K. Deb. Performance scaling of multi-objective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, volume 2632 of *LNCS*, pages 376–390. Springer, 2003. ISBN 978-3-540-01869-8.
- [43] J. Knowles and D. Corne. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In *IEEE Congress on Evolutionary Computation (CEC) 1999*, volume 1, pages –105 Vol. 1, 1999.
- [44] J. Kollat and P. Reed. A computational scaling analysis of multiobjective evolutionary algorithms in long-term groundwater monitoring applications. *Advances in Water Resources*, 30(3):408 – 419, 2007.
- [45] J. B. Kollat and P. Reed. Comparison of multi-objective evolutionary algorithms for long-term monitoring design. *Advances in Water Resources*, 29(6):792–807, 2006.
- [46] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002. ISSN 1063-6560.
- [47] M. Li, S. Yang, X. Liu, and R. Shen. A comparative study on evolutionary algorithms for many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, volume 7811 of *LNCS*, pages 261–275. Springer, 2013. ISBN 978-3-642-37139-4.
- [48] M. Li, S. Yang, and X. Liu. Shift-based density estimation for pareto-based algorithms in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 18(3):348–365, 2014. ISSN 1089-778X.
- [49] M. Li, S. Yang, and X. Liu. Diversity comparison of pareto front approxima-

- tions in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12):2568–2584, 2014. ISSN 2168-2267.
- [50] G. Lizarraga-Lizarraga, A. Hernandez-Aguirre, and S. Botello-Rionda. G-metric: An m-ary quality indicator for the evaluation of non-dominated sets. In *Genetic and Evolutionary Computation Conference (GECCO) 2008*, pages 665–672. ACM, 2008. ISBN 978-1-60558-130-9.
- [51] A. Lopez-Jaimes and C. A. C. Coello. Including preferences into a multi-objective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277(0):1 – 20, 2014. ISSN 0020-0255.
- [52] A. Lopez-Jaimes, A. Oyama, and K. Fujii. Space trajectory design: Analysis of a real-world many-objective optimization problem. In *IEEE Congress on Evolutionary Computation (CEC) 2013*, pages 2809–2816, 2013.
- [53] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. Interactive decision maps. 2004.
- [54] M. W. Mkaouer, M. Kessentini, S. Bechikh, K. Deb, and M. Ó Cinnéide. High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III. In *Genetic and Evolutionary Computation Conference (GECCO) 2014*, pages 1263–1270. ACM, 2014. ISBN 978-1-4503-2662-9.
- [55] A. Mohammadi, M. Omidvar, X. Li, and K. Deb. Integrating user preferences and decomposition methods for many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC) 2014*, pages 421–428, 2014.
- [56] S. Mostaghim and J. Teich. A new approach on many objective diversity measurement. In *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.
- [57] T. Murata and A. Taki. Many-objective optimization for knapsack problems using correlation-based weighted sum approach. In *Evolution-*

- ary *Multi-Criterion Optimization*, volume 5467 of *LNCS*, pages 468–480. Springer, 2009. ISBN 978-3-642-01019-4.
- [58] K. Narukawa and T. Rodemann. Examining the performance of evolutionary many-objective optimization algorithms on a real-world application. In *International Conference on Genetic and Evolutionary Computing (ICGEC) 2012*, pages 316–319, 2012.
- [59] A. Nguyen, M. Wagner, and F. Neumann. User preferences for approximation-guided multi-objective evolution. In *Simulated Evolution and Learning (SEAL) 2014*, volume 8886 of *LNCS*, pages 251–262. Springer, 2014. ISBN 978-3-319-13562-5.
- [60] S. Otake, T. Yoshikawa, and T. Furuhashi. Basic study on aggregation of objective functions in many-objective optimization problems. In *World Automation Congress (WAC), 2010*, pages 1–6, 2010.
- [61] C. H. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *41st Annual Symposium on Foundations of Computer Science (FOCS) 2000*, pages 86–92. IEEE Press, 2000.
- [62] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap visualization of population based multi objective algorithms. In *Evolutionary Multi-Criterion Optimization*, volume 4403 of *LNCS*, pages 361–375. Springer, 2007. ISBN 978-3-540-70927-5.
- [63] R. Purshouse and P. Fleming. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784, 2007. ISSN 1089-778X.
- [64] R. Purshouse, K. Deb, M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In *IEEE Congress on Evolutionary Computation (CEC) 2014*, pages 1147–1154, 2014.
- [65] R. C. Purshouse and P. J. Fleming. Evolutionary many-objective optimisation: An exploratory analysis. In *IEEE Congress on Evolutionary Computation (CEC) 2004*, pages 2066–2073. IEEE Press, 2003. ISBN 0-

7803-7804-0.

- [66] R. Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology And Computing In Applied Probability*, 1(2): 127–190, 1999. ISSN 1387-5841.
- [67] H. Sato, H. Aguirre, and K. Tanaka. Local dominance using polar coordinates to enhance multiobjective evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC) 2004*, volume 1, pages 188–195 Vol.1, 2004.
- [68] H. Sato, H. E. Aguirre, and K. Tanaka. Pareto partial dominance MOEA and hybrid archiving strategy included CDAS in many-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC) 2010*, pages 1–8, 2010.
- [69] O. Schutze, X. Esquivel, A. Lara, and C. A. Coello Coello. Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, 2012. ISSN 1089-778X.
- [70] H. Seada and K. Deb. U-nsga-iii: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results. In *Evolutionary Multi-Criterion Optimization*, volume 9019 of *LNCS*, pages 34–49. Springer, 2015. ISBN 978-3-319-15891-4.
- [71] N. Srinivas and K. Deb. Multiobjective function optimization using non-dominated sorting genetic algorithms. *Evolutionary Computation*, 2(3): 221–248, 1995.
- [72] J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [73] C. C. Tutum and K. Deb. A multimodal approach for evolutionary multi-objective optimisation: MEMO. *COIN Report Number 2014018*, pages 993–996, 2014.
- [74] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Graduate School of

- Engineering, Air Force Institute of Technology, Air University, 1999.
- [75] C. von Luecken, B. Baran, and C. Brizuela. A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756, 2014. ISSN 0926-6003.
 - [76] M. Wagner, K. Bringmann, T. Friedrich, and F. Neumann. Efficient optimization of many objectives by approximation-guided evolution. *European Journal of Operational Research*, page in print, 2014. ISSN 0377-2217.
 - [77] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Evolutionary Multi-Criterion Optimization*, volume 4403 of *LNCS*, pages 742–756. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-70927-5.
 - [78] D. Walker, R. Everson, and J. Fieldsend. Visualizing mutually nondominating solution sets in many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, 17(2):165–184, April 2013. ISSN 1089-778X.
 - [79] G. Wang and H. Jiang. Fuzzy-dominance and its application in evolutionary many objective optimization. In *Computational Intelligence and Security Workshops (CISW) 2007*, pages 195–198, 2007.
 - [80] H. Wang, L. Jiao, and X. Yao. An improved two-archive algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1, 2014. ISSN 1089-778X.
 - [81] R. Wang, R. Purshouse, and P. Fleming. Preference-inspired coevolutionary algorithms for many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, 17(4):474–494, Aug 2013. ISSN 1089-778X.
 - [82] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, 2006. ISSN 1089-778X.
 - [83] U. Wickramasinghe, R. Carrese, and X. Li. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *IEEE Congress on Evolutionary Computation (CEC) 2010*, pages 1–8, 2010.

- [84] Y. Yuan, H. Xu, and B. Wang. An improved NSGA-III procedure for evolutionary many-objective optimization. In *Genetic and Evolutionary Computation Conference (GECCO) 2014*, pages 661–668. ACM, 2014. ISBN 978-1-4503-2662-9.
- [85] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. ISSN 1089-778X.
- [86] X. Zhang, Y. Tian, and Y. Jin. A knee point driven evolutionary algorithm for many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, PP(99):1–1, 2014. ISSN 1089-778X.
- [87] E. Zitzler. Evolutionary algorithms for multiobjective optimization: Methods and applications. *Doctoral dissertation ETH 13398, Swiss Federal Institute of Technology (ETH)*, 1999.
- [88] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, 2001.
- [89] E. Zitzler, J. Knowles, and L. Thiele. Quality assessment of pareto set approximations. In *Multiobjective Optimization*, volume 5252 of *LNCS*, pages 373–404. Springer, 2008. ISBN 978-3-540-88907-6.