# Effects of Noisy Multiobjective Test Functions Applied to Evolutionary Optimization Algorithms

Remo Ryter, Thomas Hanne, and Rolf Dornberger
Institute for Information Systems, School of Business, FHNW University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland
Email: remo.ryter@gmail.com, {thomas.hanne, rolf.dornberger}@fhnw.ch

*Abstract*—**In this paper we study the effects of noise in multiobjective optimization problems. We consider a test function, which may be affected by noise with different strength and frequency of occurrence. To simplify the analysis, the noise is applied to only one of the objective functions, i.e. one of the objective functions is affected by additional random influences. Three different evolutionary algorithms for multiobjective problems are analyzed in this way: the Covariance Matrix Adaption Evolution Strategy (CMA-ES), the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), and the Particle Swarm Optimization (PSO). The results are presented and analyzed with respect to the resulting Pareto fronts and with respect to the distribution of variable values during the algorithm run. It can be observed that all three algorithms are basically able to derive suitable results. However, only PSO leads to a sparse Pareto front in case of noisy and non-noisy situations while CMA and NSGA-II perform similarly well. In some situations for NSGA-II and more clearly for CMA-ES specific patterns for the variable values (denoted as striae in this paper) can be observed which appear to be partly caused by the noise.**

*Index Terms*—**multiobjective optimization, evolutionary algorithms, noise, evolutionary multiobjective optimization, robust optimization**

## I. Introduction

This paper analyzes the effects of noisy evaluation functions when applied to evolutionary algorithms for solving multi-objective problems. Noise may occur due to random influences on the objective values of an optimization problem. Another reason to consider noise is when particular data of the optimization problem appears to be unknown. In both situations, a main research question is whether respective optimization algorithms, which are not aware of the specific form or consequences of noise and which work for a problem formulation without noise are sufficiently suitable also for a situation with noise.

For such algorithms, an interesting behavior is observed and the results are presented. Further details for these experiments are outlined in Chapter 4 of the thesis of Ryter [1] where six well-known multi-objective test functions from [2] are used to systematically compare evolutionary approaches. Each of the functions involves a particular feature that is known to cause difficulties in the evolutionary optimization process, mainly with respect to the convergence to the Pareto-optimal front.

In this paper, we consider the function $T_1$ (see below in Section III) for the experiments since its Pareto-optimal front is shaped like a typical convex Pareto-optimal front which can be interpreted and compared easily. This test function is extended by the addition of noise to one of the objective functions which leads to the modified test function $T_1'$. We apply noise only to one of the objective functions in order to see the resulting effects of noise more clearly.

A set of evolutionary algorithms are now applied to the undisturbed test functions and the test function disturbed with different magnitude. Three evolutionary optimization algorithms are compared in this way: the Covariance Matrix Adaption Evolution Strategy (CMA-ES), the Non-dominated Sorting Genetic Algorithm-II (NSGA-II), and the Particle Swarm Optimization (PSO).

This paper is structured as follows. Section II contains a short wrap-up of related work in the area of multiobjective problems with noise. Afterwards the test function $T_1$ and the extended test function $T_1'$ are explained in Section III. Next, the algorithms used are introduced in Section IV. The results of the experiments are presented in Section V. In Section VI, the results are discussed and conclusions and an outlook to future research in this area are presented.

## II. Related Work

Evolutionary algorithms and other nature-inspired metaheuristics can be assumed to cope well with noise because the respective biological concepts must deal with uncertain environments, vaguely measured objective values or dynamically changing conditions. It is often mentioned, that such algorithms can be considered as robust optimization concepts. Nature-inspired optimization methods can be considered as robust with respect to random changes or noise as such changes occur in nature. Such kind of robustness is therefore optimized along with the regular optimization process as we find it in evolutionary algorithms and elated approaches. However, in various studies the specific influence of noise has been studied in more detail.

For instance, Potavin *et al.* [3] stated that the problem of noise is naturally tackled by the genetic algorithm because, per definition, these algorithms are robust to

Manuscript received January 17, 2020; revised June 18, 2020.


doi: 10.12720/jait.11.3.128-134

noise, which confirms the earlier findings of Yun-can Xue *et al.* [4] that genetic algorithms are not sensitive to noise. Evolutionary algorithms, which are robust against noise, are capable of coping with the problem of noise as discussed by Eskandari and Geiger [5] and Qi *et al.* [6]. Further results for evolution strategies under the presence of noise are provided by [7], [8] and [9].

However, those examples usually consider only single-objective optimization problems. How is it in terms of multiobjective optimization? Does it behave similar or do the algorithms react differently? Some publications, which study related problems, are cited in [10], [11] and [12]. In Reference [10] it is analyzed whether noise added to the objective functions can even increase the robustness of optimization results in the context of evolutionary algorithms. The authors suggest a modified ranking process, which is capable of producing improved results. In Reference [11] three noise-handling features are investigated: an experiential learning directed perturbation operator that adapts the magnitude and direction of variation according to past experiences, a gene adaptation selection strategy that helps the evolutionary search in escaping local optima, and a possibilistic archiving model based on the concept of possibility and necessity. In Reference [12], two types of anti-noise methods are analyzed in the context of the NSGA2 method: probabilistic and re-sampling methods, The results indicate that the probabilistic approach has a better convergence to the Pareto optimal front, but it loses diversity quickly. On the other hand, methods based on re-sampling are more robust against noise but are computationally very expensive to use. Based on the findings that frequently test functions have a bias towards the region where the robust solutions are located, a framework for the construction of robust continuous multiobjective test functions based on different noise-induced features is studied in [13]. However, these noise-induced features can provide different difficulties to the optimization algorithms. It is found that some noise-induced features present greater challenges to robust MOEAs as compared to existing robust test functions.

In the paper by Rakshit *et al.* [14], noise ("measurement noise") is explored in the context of a differential evolution approach for multiobjective optimization. Llorà and Goldberg [15] analyzed noise in the context of Multiobjective Learning Classifier Systems, which are an approach based on Genetic Algorithms. Although previous studies mainly focus on the performance or improvements of evolutionary algorithms under noise, there are also some studies dealing with other methods. For instance, in [16] noise is analyzed in the context of the artificial bee colonies algorithm. In Reference [17], noise is investigated in the framework of ant colony optimization. Another example is [18] were noise was studied in relationship to particle swarm optimization.

## III. Test Functions

Different test functions are defined to test the algorithms' reactions to features of Pareto-optimal fronts [2]. For this paper, only the first test function $T_1$ is used.

The definition of $T_1$ is given in as follows in (1).

$$f_0(x) = x_o$$

$$g(x_1, \dots, x_m) = 1 + 9 \sum_{i=2}^{m} x_i/(m-1) \qquad (1)$$

$$h(f_0, f_1) = 1 - \sqrt{f_0/f_1}$$

This test function has a convex Pareto-optimal front where $m$=30, and $x_i \in [0, 1]$. The Pareto-optimal front is formed with $g(x)$=1.

In order to test how the evolutionary algorithms react to noise, the function is extended by adding a random number to its second objective function $g(x_2, ..., x_m)$. This leads to the extended test function $T_1'$ defined as follows (2):

$$f_0(x) = x_o$$

$$g(x_1, \dots, x_m) = 1 + 9 \sum_{i=2}^{m} x_i/(m-1) + r \qquad (2)$$

$$h(f_0, f_1) = 1 - \sqrt{f_0/f_1}$$

The only change is the addition of the variable $r$, which represents a random number. The value range for this random variable is defined according to the required noise level: $r \in [0, 0]$ (no noise), $r \in [-0.05, 0.05]$ (weak noise) or $r \in [-0.5, 0.5]$ (strong noise).

In the experiments, the probability of noise application in the evaluation of the objective function is specified. Two probabilities are used: 1% application rate and 10% application rate. This results into five different noise settings for the experiments shown in Table I.

TABLE I. Noise Settings

| | Strong noise | Weak noise |
|---|---|---|
| High probability | ±0.5, p = 0.1 | ±0.05, p = 0.1 |
| Small probability | ±0.5, p = 0.01 | ±0.05, p = 0.01 |
| | No noise applied | |

## IV. Algorithms and Settings

In order to identify the effects of a noisy evaluation function to evolutionary optimization algorithms, a set of different algorithms has to be defined. Three different methods are selected to study the effects on different evolutionary algorithms: the Covariance Matrix Adaption Evolution Strategy (CMA-ES) [19], the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [20], and the Particle Swarm Optimization (PSO) [21]. By choosing an evolution strategy, a genetic algorithm and a swarm based optimization approach, three sufficiently different areas of evolutionary computation are covered.

To make the results comparable, all algorithms are aligned with their settings so that every optimization run performs the same amount of fitness evaluations for every different algorithm. All algorithms share the following settings:

- 100 individuals
- 250 generations

Depending on the given algorithm, other settings were required such as crossover probability or particle weights.

All the algorithms have some sort of built-in elitism. The NSGA-II performs its non-dominated sorting, the CMA-ES is an elitist evolution strategy and the PSO uses the best previous location of a particle and the global best solutions as attraction points for the particle movement. Strictly speaking, the PSO has no built-in elitism. However, the global best solution and the particle's best-known solutions are kept during the optimization process, similar to preserving superior individuals within elitist algorithms.

Thus, an exceptionally well performing individual can be kept inside the population due to elitism. However, such an individual could be identified just because of a noisy evaluation function. How will the algorithms behave in such a case?

## V. Experimentation and Results

To study the effects caused by noisy evaluation functions, each algorithm is tested against the unmodified test function $T_1$ and the extended test function $T_1^{'}$ with added noise. The differences between those two experiments can then be isolated and analyzed.

The analysis of the effects is performed qualitatively. Therefore, the results which are studied are from single experimental runs and do not represent an average from several runs. The experiments were conducted multiple times and the discovered effects were the same for all runs.

The first comparison between the achieved results from established algorithms compared to the tested algorithms by [2] by solving $T_1$ are not discussed here. It can be said that all three tested algorithms were able to converge at least as reliable and quick as the winning SPEA algorithm from [2].

The results from the $T_1$ function are, however, used to compare them with the results from the modified test function $T_1^{'}$.

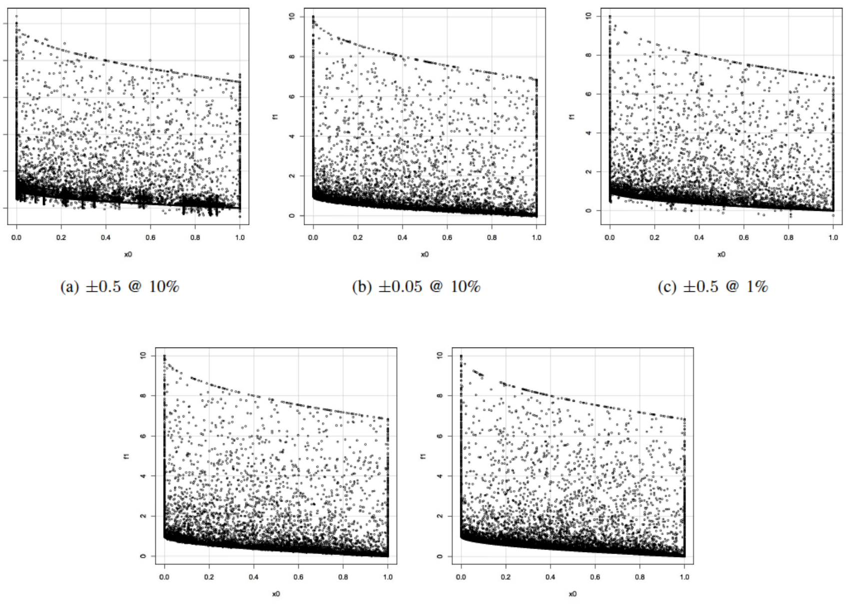
### *A. CMA-ES*

For the CMA-ES the following settings are used:

- lambda (number of children of each family): 1
- lambdaMO (number of families): 100
- pThreshold (percentage of children better than parents): 0.44
- limitSigma: false (do not stop if a limit value for sigma is reached)
- limitObjective: false (do not stop if limit values for the objectives are reached)
- updateRule: normal (with eigenvalue decomposition)
- newSelect: disabled (do not use the strict selection rule to increase diversity)

As a first result, the scatterplots for both objective functions $f_0$ ($f_0$=$x_0$) and $f_1$ for each noise setting are shown in Fig. 1. It is observable that the CMA-ES is able to achieve the Pareto-optimal front even with a noisy evaluation function. The noise, however, is clearly visible when negative noise is applied to individuals, which are already on or near the Pareto-optimal front as they are placed below that front (for example in Fig. 1a).

An interesting observation is that the distribution of objective values collapses when noise is applied. A comparison of the value distribution of $x_0$ (unaffected by noise) between the $T_1$ without noise compared to $T_1^{'}$ with strong noise often applied is shown in Fig. 2. It can be observed that around the 7'000th evaluation the diversity of values for $x_0$ is reduced to around 15 different values by a noisy evaluation function included in the optimization process. The difference is substantial compared to the well-distributed value range for $T_1$ without noise throughout the complete optimization process.



(a) ±0.5 @ 10%

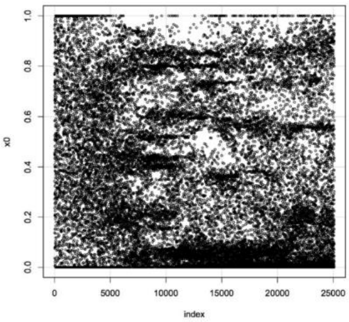
(b) ±0.05 @ 10%

(c) ±0.5 @ 1%

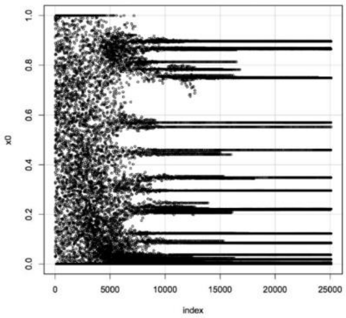
(d) ±0.05 @ 1%

(e) No noise

Figure 1. Scatterplots of the different solution sets of CMA-ES (x axis: $x_0$; y axis: $f_1$).



(a) $\pm 0.05$ @ 1%



(b) No noise

Figure 2. Scatterplots of the distribution of $x_0$ using CMA-ES (x axis: index; y axis: $x_0$).

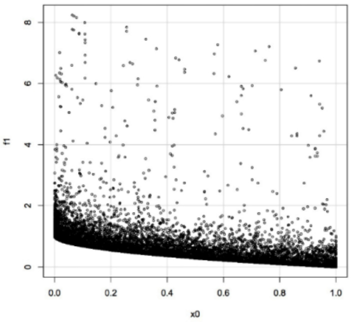
### B. NSGA-II

For the NSGA-II algorithm, the following settings of parameters are used:

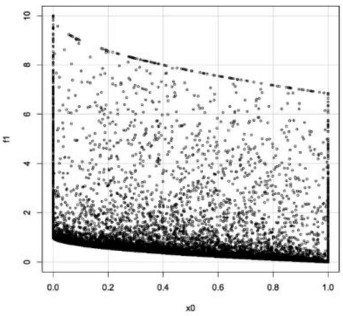
- SBX Crossover probability: 0.9
- SBX Crossover distribution index: 20
- Polynomial mutation probability: 0.5
- Polynomial mutation distribution: 20
- Selection method: BinaryTournament 2

The scatterplots for both objective functions are very similar to the ones in Fig. 1 with the difference that the dominated solutions found by the NSGA-II are closer grouped towards the Pareto-optimal front than the ones found by CMA-ES (see Fig. 3).

The most interesting observation during the experiments with the NSGA-II is the effect caused by strong noise, which is seldom applied (±0.5 with $p$=0.01). A comparison of different distributions for $x_0$ with strong noise and no noise are shown in Fig. 4. The effect visible in Fig. 4c is noteworthy since this behavior is very different from every other distribution of $x_0$. Beside this interesting pattern, the NSGA-II is very robust against noise and efficiently finds the Pareto-optimal front for all noise settings.



(a) NSGA-II



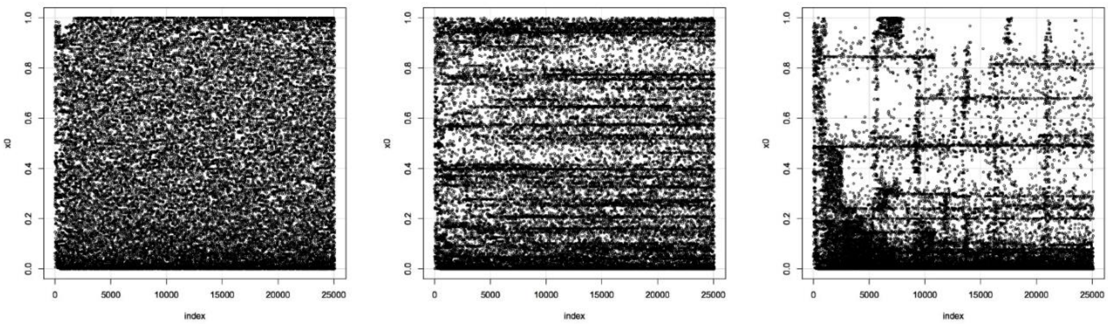
(b) CMA-ES

Figure 3. Solution comparison for NSGA-II and CMA-ES ($T_1$) (x axis: $x_0$; y axis: $f_1$).

### C. PSO

For the PSO the following settings are used:

- Repository size (percent of leading particles): 30
- Inertia (delay of speed change): 0.8
- Weight C1 (the particle’s best direction): 0.4
- Weight C2 (the leading particle’s best direction): 0.4
- Mutation probability: 0.0
- Set lower limit on velocity: false



(a) No noise (b) $\pm 0.5$ with $p = 0.1$ (c) $\pm 0.5$ with $p = 0.01$

Figure 4. Scatterplots of the distribution of $x_0$ using NSGA-II (x axis: index; y axis: $x_0$).

The first interesting finding is that the PSO algorithm is not able to achieve a closed line of solutions for the Pareto-optimal set. In fact, the Pareto-optimal set merely consists of twelve solutions only after 25'100 evaluations (see Fig. 5).

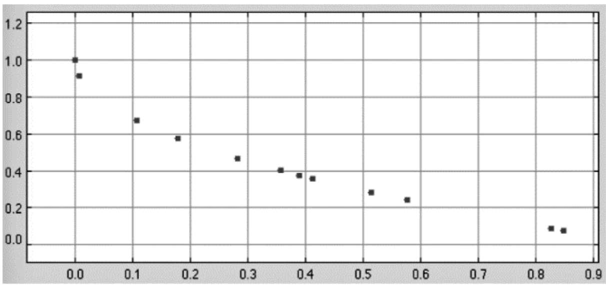
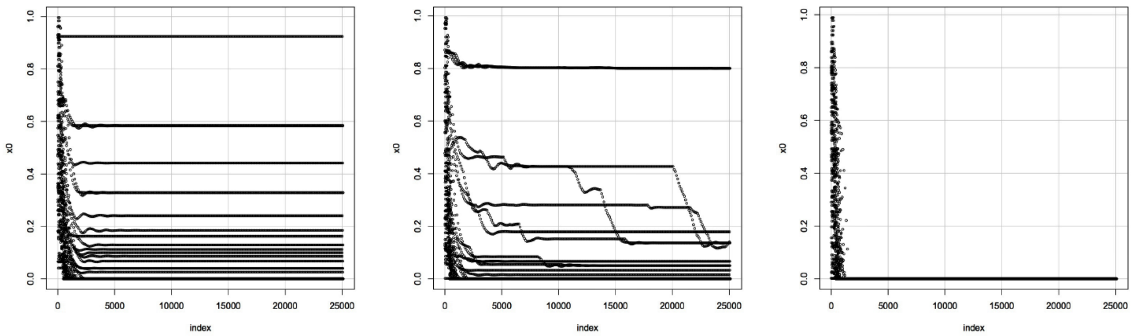


Figure 5. Pareto-optimal front for $T_1$ by PSO.

This leads to the next observation regarding the value distribution of $x_0$ during the optimization process. Unlike for the CMA-ES or the NSGA-II, the PSO already struggles with this distribution for the test function $T_1$ without noise being applied to it. In the scatterplots for this distribution in Fig. 6, the reason is visible. In Fig. 6a where the test function $T_1$ is optimized without any noise effects, the PSO starts to limit itself to only a few values for the complete optimization process. It seems that those particles represent such a good solution that all the neighboring particles are attracted by them and stop searching for better solutions. When the noise is introduced, those single solutions are even combined which leads to an even worse distribution (see Fig. 6b). When heavy noise is frequently applied, as in Fig. 6c, then the PSO collapses to one single solution for $x_0$.



(a) No noise (b) $\pm 0.05$ with $p = 0.1$ (c) $\pm 0.5$ with $p = 0.1$

Figure 6. Scatterplots of the distribution of $x_0$ using PSO (x axis: index; y axis: $x_0$).

## VI. Discussion and Conclusions

From the experiments, it became obvious that the considered algorithms are suitable to find reasonable results also in the presence of noise to a different extent. For CMA-ES and NSGA-II, the resulting Pareto fronts are quite similar in case of no noise and with different levels of noise. The defective effects on the Pareto front, which can be seen especially as outliers below the Pareto front in Fig. 1(a) and Fig. 1(c) are comparably small even in situations with stronger or more frequent noise.

In case of the PSO, we observe a significantly thinned out Pareto front and this effect intensifies in the presence of noise. This phenomenon can be described as a collapse of $x_0$ values when noise is applied (even more than without noise). This can be attributed to the often observed phenomenon of premature convergence [22]. We assume that a better tuning of the PSO should be able to better deal with the perceived optimization problem.

When looking at the distribution of variable values (shown for the example of $x_0$) instead of objective values we observe specific patterns over time (striae). Obviously, only specific values for these variables match with good solutions that should converge to the Pareto front. In case of CMA-ES, such striae are created after about 7000 evaluations when noise applied. The respective pattern is less clear and $x_0$ values are noisier compared to the non-noise situation. Interestingly for NSGA-II, such striae are not observed in the noise-free experiment but striae appear in experiments with noise. Similar to CMA-ES, these striae (or, respectively, the $x_0$ values) are noisy. Occasionally, "vertical" striae can be observed, which can be interpreted as a sudden increase of diversity in the population during an experiment (see Fig. 4(c) for the NSGAII with rarely applied strong noise).

Considering the distribution of variable values in general and the specific striae phenomenon we should note that for the considered optimization problem it is not required to have variable values such as $x_0$ only at or around a few particular values (as obvious, e.g., in Fig. 4(a)). However, it might be sufficient to work only with a limited number of such values to get a well-approximated Pareto front. If an algorithm is rather "conservative" in producing new solutions there may be a tendency to stick towards found variable values which lead to good results. This effect might be increased in the presence of noise and result in unreliable evaluations of the objective functions.

For future research, it is suggested to perform further studies of the effects of noise in multiobjective optimization. For the large number of respective optimization methods, especially from the field of metaheuristics and nature-inspired algorithms, only few results are available by now. Moreover, it would be interesting to study further types of optimization problems. Especially for combinatorial optimization problems, the effects of noise may become more severe due to the discrete character of the variable space and the constraints, which need to be observed. In general, it can be assumed that such optimization problems (especially NP-hard problems) suffer more from the presence of noise. In any case, we can assume that smaller changes of the problem formulation can lead to "jumps" in the optimal solutions due to the discontinuous nature of such optimization problems. From a methodological point of view, quantitative studies of the noise effects are certainly important to quantify the effects but further qualitative studies may provide further insights in the noise effects, which may be helpful to improve optimization algorithms and to better solve of more complex optimization problems that are affected by noise.

## Conflict of Interest

The authors declare that there is no conflict of interest related to this publication.

## Author Contributions

Remo Ryter worked out the research project in details and conducted the numerical experiments within his Master Thesis. A major part of the text and all figures are prepared by him. Thomas Hanne contributed parts of the text, edited the document and conducted some revisions. Rolf Dornberger designed and supervised the research project. He also contributed to the revision of the paper. All authors had approved the final version.

## References


[1] R. Ryter, "Multi-objective optimization of real-time portfolio data using evolutionary computation," master's thesis, University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, 2015.

[2] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173-95, Jan. 2000.

[3] J. Potavin, N. Jongswat, and W. Premchaiswadi, "Applying fuzzygenetic mining in conformance and dependency relations," in *Proc. 10th International Conference on ICT and Knowledge Engineering*, 2012, pp. 228-235.

[4] Y. C. Xue, Q. W. Yang, and Z. Q. Mei, "Combined algorithm for timevarying system based on the damped least-squares estimation algorithm and improved genetic algorithm," in *Proc. International Conference on Machine Learning and Cybernetics*, 2003, vol. 3, pp. 1814-1819.

[5] H. Eskandari and C. D. Geiger, "Evolutionary multiobjective optimization in noisy problem environments," *Journal of Heuristics*, vol. 15, no. 6, pp. 559-595, 2009.

[6] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "Moea/d with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231-264, 2014.

[7] D. V. Arnold and H. G. Beyer, "A comparison of evolution strategies with other direct search methods in the presence of noise," *Computational Optimization and Applications*, vol. 24, no. 1, pp. 135-159, 2003.

[8] H. G. Beyer, "Evolutionary algorithms in noisy environments: Theoretical issues and guidelines for practice," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 239-267, 2000.

[9] D. V. Arnold and H. G. Beyer, "A general noise model and its effects on evolution strategy performance," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 380-391, 2006.

[10] E. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Evolutionary Multi-criterion Optimization*, Springer, 2001, pp. 329-343.

[11] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 354-381, 2007.

[12] L. T. Bui, H. A. Abbass, and D. Essam, "Fitness inheritance for noisy evolutionary multi-objective optimization," in *Proc. the 7th Annual Conference on Genetic and Evolutionary Computation*, 2005, pp. 779-785.

[13] C. K. Goh, K. C. Tan, C. Y. Cheong, and Y. S. Ong, "An investigation on noise-induced features in robust evolutionary multi-objective optimization," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5960-5980, 2010.

[14] P. Rakshit, A. Konar, S. Das, L. C. Jain, and A. K. Nagar, "Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 7, pp. 922-937, 2013.

[15] X. Llorà and D. E. Goldberg, "Bounding the effect of noise in multiobjective learning classifier systems," *Evolutionary Computation*, vol. 11, no. 3, pp. 279-298, 2003.

[16] P. Rakshit, A. Konar, and A. K. Nagar, "Artificial bee colony induced multi-objective optimization in presence of noise," in *Proc. IEEE Congress on Evolutionary Computation (CEC)*, July 2014, pp. 3176-3183.

[17] T. Friedrich, T. Kötzing, M. S. Krejca, and A. M. Sutton, "Robustness of ant colony optimization to noise," *Evolutionary Computation*, vol. 24, no. 2, pp. 237-254, 2016.

[18] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimizer in noisy and continuously changing environments," *Methods*, vol. 5, no. 6, p. 23, 2001.

[19] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutionary Computation*, vol. 15, no. 1, pp. 1-28, Jan. 2007.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, Apr. 2002.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. International Conference on Neural Networks*, 1995, vol. 4, pp. 1942-1948.

[22] I. C. Trelea, "The particle swarm optimization algorithm: Convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317-325, 2003.




**Thomas Hanne** received his master's degrees in economics and computer science, and a PhD in economics. From 1999 to 2007 he worked at the Fraunhofer Institute for Industrial Mathematics (ITWM) as a senior scientist. Since then he is professor for information systems at the FHNW University of Applied Sciences and Arts Northwestern Switzerland and the head of the Competence Center Systems Engineering since 2012.
Thomas Hanne is an author of more than 140 journal articles, conference papers, and other publications and editor of several journals and special issues. His current research interests include multicriteria decision analysis, evolutionary algorithms, metaheuristics, optimization, simulation, systems engineering, software development, logistics, and supply chain management.

**Rolf Dornberger** is a full professor and lecturer at the FHNW University of Applied Sciences and Arts Northwestern Switzerland FHNW since 2002. He holds the position as the head of the Institute for Information Systems at the School of Business FHNW since 2007 as well as the head of the competence centre New Trends & Innovation since 2013. Formerly, he was the head of the competence centres Technology, Organization & People (2014-17) and Systems Engineering (2006-10). He holds a PhD and a Diploma in Air- and Aerospace Engineering. Before becoming a professor, he worked in industry in different management positions as a consultant, IT officer and senior researcher in different engineering, technology, and IT companies in the field of energy, software, IT, and airline business. His current research interests include artificial intelligence, particularly computational intelligence and optimization, robotics, human-machine interaction, the Internet of things as well as innovation management and learning didactics.