Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes

Dirk Büche, Peter Stoll, Rolf Dornberger, and Petros Koumoutsakos

***Abstract*—Evolutionary Algorithms have been applied to single and multiple objectives optimization problems, with a strong emphasis on problems, solved through numerical simulations. However in several engineering problems, there is limited availability of suitable models and there is need for optimization of realistic or experimental configurations. The multiobjective optimization of an experimental set-up is addressed in this work. Experimental setups present a number of challenges to any optimization technique including: availability only of pointwise information, experimental noise in the objective function, uncontrolled changing of environmental conditions and measurement failure.**

This work introduces a multiobjective evolutionary algorithm capable of handling noisy problems with a particular emphasis on robustness against unexpected measurements (outliers). The algorithm is based on the Strength Pareto Evolutionary Algorithm (SPEA) of Zitzler and Thiele and includes the new concepts of domination dependent lifetime, reevaluation of solutions and modifications in the update of the archive population. Several tests on prototypical functions underline the improvements in convergence speed and robustness of the extended algorithm.

The proposed algorithm is implemented to the Pareto optimization of the combustion process of a stationary gas turbine in an industrial setup. The Pareto front is constructed for the objectives of minimization of NO_x emissions and reduction of the pressure fluctuations (pulsation) of the flame. Both objectives are conflicting affecting the environment and the lifetime of the turbine, respectively. The optimization leads a Pareto front corresponding to reduced emissions and pulsation of the burner. The physical implications of the solutions are discussed and the algorithm is evaluated.

***Index Terms*—Combustion instabilities, emission reduction, evolutionary algorithms, gas turbine combustion, multiobjective optimization, noisy objective functions.**

I. Introduction

AUTOMATED optimization is an important aspect of technical product design. In an engineering environment it usually implies the development of an optimization algorithm integrated in an automated setup for the modification of parameters of the design. For complex problems such as the combustion process, numerical simulations are not widely used as a predictive tool due to the complexity of the physical phenomena under investigation. Although intensive research efforts are underway on this front, experimental setups are widely used for the study and optimization of combustion processes.

Manuscript received April 5, 2002; revised May 30, 2002.

D. Büche and P. Koumoutsakos are with the Institute of Computational Science, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (e-mail: bueche@inf.ethz.ch; petros@inf.ethz.ch).

P. Stoll is with Alstom Power Technology, Segelhof, 5405 Dättwil, Switzerland (e-mail: peter.stoll@power.alstom.com).

R. Dornberger is with the University of Applied Sciences Solothurn, Northwestern Switzerland, Olten, Switzerland (e-mail: rolf.dornberger@fhso.ch).

Digital Object Identifier 10.1109/TSMCB.2002.804372

The optimization of the combustion process of a stationary gas turbine is a challenging real-world application with conflicting objectives. New governmental laws on emission taxes and global agreements on emission reduction such as the Kyoto resolution on greenhouse-gases (1997, 2001) demand expensive, highly thermodynamically efficient power plants with low emissions. On the other hand, the liberalization of the electric power market puts pressure on overall production costs.

In recent years the use of gas turbines among new power plants has significantly increased due to a number of appealing properties: Using natural gas instead of coal or oil leads to a cleaner combustion, while moderate installation and operating costs and a high thermodynamically efficiency reduce overall energy production costs. Moreover, using the exhaust heat for a steam turbine in a combined cycle is one way to increase power output and efficiency of the plant.

A central component in the design of a gas turbine is the design of the burners in the combustion chamber. The burners mix air and fuel and combust them continuously. This is different to Diesel engines, which combust in a cyclic manner. The design of a burner addresses two main objectives: First, the burner should mix air and fuel uniformly for low emissions, since the presence of areas of rich combustion results in increased NO_x emissions and a nonhomogeneous temperature distribution may damage the turbine blades. Second, the burner should produce a stable combustion flame, avoiding undesired pulsations. Pulsations are due to thermo acoustic waves, which occur in particular for lean combustion when operating under part load condition. They reduce the lifetime of the turbine by fatigue and by destroying the film cooling along the blades surface. These two objectives are conflicting, thus motivating the requirement for a variety of designs as manifested on a Pareto front. The lack of viable analytical models and the limited information about the underlying physical processes involved, makes combustion processes a suitable candidate for the optimization using stochastic optimization techniques such as evolutionary algorithms [8].

Evolutionary Algorithms (EAs) are biologically inspired optimization algorithms, incorporating operators such as mutation, recombination and fitness based selection of parameters. EAs use a set of solutions (population), to converge to the optimal design(s). The population-based search allows easy parallelization and information can be accumulated so as to generate accelerated algorithms [12]. EAs are robust optimization methods, which do not require gradients of the objective function and may avoid termination at local minima.

EAs operate so as to continuously obtain an improvement of the objective function by exploiting progressively acquired information. While EAs have found several applications for single objective optimization, industrial applications often entail multiple, conflicting objectives. In this context, the concept of dominance allows a partial ordering of solutions. A solution is dominating another solution, if it is superior or equal in all objectives, but at least superior in one objective. The complete set of nondominated solutions is referred to the Pareto set of solutions, after the work of the engineer and economist Vilfredo Pareto [18] and represents the best solutions to the problem.

A classical and still widely employed approach to handle multiple objectives is the aggregation [13] of all objectives into a single, *a priori* defined figure of merit. Objectives are usually aggregated by a weighted-sum or a constraint approach. This weighting behavior implies prior knowledge about the problem and is dependent on the *a priori* unknown shape of the Pareto front. While point-to-point search methods converge to one Pareto solution at a time, evolutionary algorithms can exploit the population-based feature and converge to the Pareto-set in a single optimization run. Therefore much effort has been spent over the past twenty years on the development and application of evolutionary algorithms for Pareto optimization. Promising methods have been proposed and compared by several researchers [3], [25], [26]. An exhaustive list of references can be found on the web page of Coello [4]. The various multiobjective evolutionary algorithms are usually distinguished by their fitness assignment operators, while the mutation, and the crossover operators are usually adopted from standard single-objective algorithms. Pareto optimization methods, which use the dominance criterion for the fitness assignment are widely used as Pareto dominance is key issue in determining, if one solution performs better than the other [9]. Two of the most prominent multiobjective evolutionary algorithms are the Nondominated Sorting Genetic Algorithm (NSGA) of Srinivas and Deb [21], and the Strength Pareto Evolutionary Algorithm (SPEA) of Zitzler and Thiele [26].

NSGA assigns fitness by nondominated sorting of the population as described by Goldberg [10]. The nondominated solutions of the population are assigned the highest fitness and are removed from the population. Then, the nondominated solutions of the remaining population are assigned a lower fitness. This is repeated until all solutions are sorted. Within each layer of nondominated solutions phenotypic fitness sharing is used in order to preserve diversity.

SPEA uses the nondominated solutions for the fitness assignment. The nondominated solutions are assigned the highest fitness. The fitness of a dominated solution decays with the number of nondominated solutions by which it is dominated. A main difference of SPEA to NSGA is the use of elitism, a technique of preserving always the best solutions obtained so far. In multiobjective optimization, elitism is performed by preserving the nondominated solutions in an archive [25]. The parents of the next generation are selected out of the current population and the archive.

Although the number of applications in the field of multiobjective (Pareto) optimization is increasing, problems with noisy objective functions are rarely considered, even though noise is present in almost every real-world application. As evolutionary algorithms do not require gradient information, they are already inherently robust to small amounts of noise, a feature which is sufficient for many problems. In several experiments however, large-amplitude noise is induced from various sources, such as unsteady operating conditions, limited measurement precision, and time averaging in restricted sampling time. In addition, measurements may fail, leading to erroneous outliers, characterized by nonphysical objective values. Standard multiobjective evolutionary algorithms cannot handle these difficulties, and there is a need to extend their basic components to overcome these difficulties.

While for single objective optimization, several studies of noisy objective functions have already been performed [17], [20], for multiobjective optimization, limited results are available in literature. Averaging the parent population, a remedy for noisy single objective problems, is not useful in this case since a diverse population is desired to converge toward the Pareto front. Two recent publications [14], [22], adapt the Pareto ranking scheme [10] to noisy solutions by defining probabilities of dominance between them. Both methods assume either a uniform or normal distribution of the noise and can benefit from *a priori* knowledge of its magnitude.

In this paper we introduce three new principles in order to improve robustness against noise. First, a dominance-dependent lifetime is assigned to each individual. The lifetime is inversely proportional to the number of solutions it dominates. This limits the impact of a solution in the overall population. In addition, we enable nondominated solutions to be reevaluated after their lifetime expires and define an extended update mechanism for the archive.

This paper is organized as follows: First, the principles of multiobjective optimization are described and the SPEA algorithm is presented. Then we present an overview on modifications for SPEA in order to handle noise and introduce a new approach called the noise-tolerant SPEA (NT-SPEA). All algorithms are analyzed on noisy and noise-free test functions. The analyzed noise reflects the characteristics of the intended application. Finally the application of NT-SPEA to the optimization of a gas turbine burner is presented, showing the capabilities of the new approach. The optimization leads a Pareto front corresponding to reduced emissions and pulsation of the burner. The physical implications of the solutions are discussed.

II. Multiobjective Evolutionary Algorithms

A. Definition of Multiobjective Optimization

A multiobjective optimization problem can be described by an objective vector f and a corresponding set of design variables x. Without loss of generality we can consider the minimization of f. Formally

$$\min f(x) = (f_1(x),\, f_2(x),\, \ldots,\, f_m(x)) \in F \quad \text{where } x = (x_1,\, x_2,\, \ldots,\, x_n) \in X \tag{1}$$

where $X \in \mathcal{R}^n$ is the n-dimensional design space, $F \in \mathcal{R}^m$ is the m-dimensional objective space. Here both the design and objective space are real spaces, as they correspond to continuous variables and measured objectives for the proposed application. A partial ordering can be applied to solutions in the objective space F by the dominance criterion. A solution a in X is said to dominate a solution b in X ($a \succ b$), if it is superior or equal in all objectives and at least superior in one objective. This is expressed as

$$a \succ b, \text{ if } \quad \forall i \in \{1, 2, \ldots, m\}: \; f_i(a) \leq f_i(b) \wedge \\ \exists j \in \{1, 2, \ldots, m\}: \; f_j(a) < f_j(b). \qquad (2)$$

The solution a is said to be indifferent to a solution c, if neither solution is dominating the other one. When no *a priori* preference is defined among the objectives, dominance is the only way to determine, if one solution performs better than the other [9]. The complete set of Pareto ideal solutions represents the best solutions to a problem. In other words, starting from a Pareto solution, one objective can only be improved at the expense of at least one other objective. From the Pareto definition, two targets have to be considered by the formulation of an evolutionary optimization algorithm for Pareto optimization. On one hand, the algorithm must be able to converge sufficiently fast toward the Pareto front, while on the other, it must preserve diversity among its population in order to be able to spread over the whole Pareto front. A common difficulty is the focusing of the population on a certain part of the Pareto front, which is known as genetic drift [11]. In single objective optimization the latter issue is unimportant, since convergence to a single (global) optimum is desired.

B. Basic Elements of a Multiobjective Evolutionary Algorithm

Evolutionary Algorithms are optimization algorithms, incorporating concepts such as fitness based selection, recombination and mutation. EAs start with a set of λ randomly generated solutions, which are referred to as the population P. For a multiobjective problem, a *selection operator* selects in average the less dominated solutions from P and places them in a parent population P_p of size μ. A selection operator is described in Section II-C.

The *recombination operator* chooses randomly individuals from the parent population P_p and recombines them into a child. With 50% probability each, uniform recombination with two parents or no recombination is chosen for all performed optimizations in this paper.

For the *mutation operator*, the variables of a child are mutated by adding normally distributed random numbers with a standard deviation σ of 0.1, relative to the interval size in which the variable is defined, and a mutation probability p_M of 20%. This normally distributed mutation reflects the natural principle that small mutations occur more often than large ones.

A termination criterion for the evolution may be the maximal allowed number of generations.

C. Strength Pareto Evolutionary Algorithm

The Strength Pareto Evolutionary Algorithm (SPEA) of Zitzler and Thiele [26] is a well-established Pareto-optimization algorithm. The advantages and drawbacks of the method have been extensively discussed [27], [26]. SPEA describes a selection operator, while the recombination and mutation operator can be used from a single objective algorithm or, e.g., from Section II-B.

The algorithm entails a fitness assignment and selection mechanism based on the concept of elitism. SPEA uses the nondominated solutions for the fitness assignment. First, the fitness of each nondominated solution is computed as the fraction of the population, which it dominates. The fitness of a dominated individual is equal to one plus the fitness of each nondominated solution by which it is dominated. This fitness assignment guarantees that the fitness of nondominated solutions is always lower than the fitness of the dominated.

Elitism is a technique of preserving always the best solutions obtained so far. In multiobjective optimization, elitism is performed by storing the nondominated solutions in an archive A [25]. In the selection process individuals of the current population P and of the archive A are competing in a binary tournament where contrary to the standard tournament selection the solution with the lower fitness wins.

In order to preserve diversity in the archive and to keep its size limited, a clustering algorithm is used. Clustering removes solutions in areas of high density as measured in the objective space.

The studies of Zitzler and Thiele [26] have illustrated that elitism improves the performance of multiobjective evolutionary algorithms on noise-free test problems. Elitism is inserting nondominated solutions in the selection process, and thus increasing the selection pressure. An increasing number of multiobjective algorithms followed this observation. For example, NSGA was updated by its inventors to NSGA-II, which contains "controlled elitism" [6]. Some researchers state elitism as a necessity for multiobjective optimization [25], since information may be lost by the stochastic selection operator. However, this advantage is debatable for noisy objective functions.

Selection is performed by a binary tournament. All solutions of the population P and the archive A are put in one pot. Then, always two solutions are taken from the pot without replacement. These two solutions participate in a tournament. The winner is the solution with the lower fitness, which is copied into the parent population P_p. If the pot is empty, it gets refilled until the desired size μ of P_p is reached.

With the SPEA algorithm, a multiobjective evolutionary algorithm can be written as follows.

Algorithm 1
1. **begin**
2. Generate an initial population P of random individuals and an empty archive A.
3. Evaluate the objectives of the individuals in P.
4. **while** termination criterion is not fulfilled **do**
5. Update archive A: Add a copy of the current population P to A and remove the dominated individuals from A. Limit the size of A by clustering.
6. Fitness assignment: Assign fitness to the individuals in P and A.
7. Selection: Use tournament selection for selecting the parent population P_p from $P \cup A$.
8. Recombination: Generate a new population P by recombination of the individuals in P_p.
9. Mutation: Mutate the individuals in P.
10. P is the population of the next generation.
11. Evaluate the objectives of the individuals in P.
12. **end while**
13. **end**

III. Multiobjective Evolutionary Algorithms for Noisy Applications

For optimization noisy applications like real-world problems and experimental setups, modifications are needed to the standard multiobjective evolutionary algorithms in order increase their robustness. This section starts with the definition of noise, and then different modifications for SPEA in order to be more robust to noise are presented.

A. Definition of Noise in Applications

In experiments and industrial configurations we can always detect different results for repeated measurements of the same operating point. The differences are attributed to noise and unobserved factors in the setup.

Noise may occurs in various areas in the experiment: The setting of the operating conditions is within a limited precision. In the realization, the operating condition may vary over time and finally measurement errors occur. It is up to the careful setup by the experimenter to keep the noise within a limited range. We define this noise, which is present in all measured experiments, as **experimental noise**. It is often modeled by a normal distribution with defined mean and standard deviation, which define *a priori* knowledge of the processes involved.

In addition, during an automated optimization cycle, an experimental measurement may fail completely, producing **outliers**, i.e., arbitrary nonphysical results. This occurs very rarely, but may have large impact on the automated process optimization if not recognized by a supervisor or captured by some penalty function. Outliers cannot be described by a statistical model with given mean and deviation, but are best modeled by a probability of occurrence. Noise and outliers influence the multiobjective optimization process by misleading the selection operation. Hence unrealistic inferior solutions may dominate superior ones, thus delaying or completely misleading the convergence to an unrealistic Pareto front.

B. Nonelitistic Strength Pareto Evolutionary Algorithm

The presence of noise affects the fitness assigned to an individual. This may cause inferior solutions to occasionally win in the selection process. Multiobjective evolutionary algorithms, which implement elitism, would then select these solutions into the archive, thus misleading the entire optimization run by participating in the selection process. More important, these solutions may dominate other solutions in the archive and in the worst case all other solutions in the archive are then removed. In order to avoid this, a first and simple modification of *original SPEA* of Zitzler and Thiele [26] is proposed. We define a *nonelitistic* SPEA algorithm. In each generation, the archive is filled with the nondominated solutions of the current population. Nondominated solutions from previous generations are not considered.

C. Statistical Strength Pareto Evolutionary Algorithm

Reevaluating a solution several times and taking the mean as a statistical estimate can decrease the level of noise in an objective function. Implementing this approach into SPEA is simple and is in the following referred to as *statistical SPEA*. The disadvantage of this concept is the increased evaluation cost per solution. A lower limit for a statistical estimate is seven evaluations. This number is used for the performance comparison in the next section.

D. Estimate Strength Pareto Evolutionary Algorithm

The Estimate Strength Pareto Evolutionary Algorithm (ESPEA) of Teich [22] modifies the SPEA algorithm in order to be more robust to noise by introducing a *probability of dominance*. It is assumed that each objective value f cannot be computed exactly, but can be bounded within a *property interval* $[f^L, f^U]$, where f^L and f^U are the lower and upper bound of the interval, respectively. In addition, the probability of the function value is assumed to be uniform within the interval. These assumptions lead to the new definition of a probability of dominance. If two solutions with overlapping property intervals are compared, the dominance has to be assigned by a probability. Teich computed the probability for minimizing an arbitrary number of m objectives. If two solutions a and b with the property intervals $[a_i^L, a_i^U]$ and $[b_i^L, b_i^U]$, $i = 1, \ldots, m$, respectively, are compared, the probability that a dominates b is given by

$$P(a \succ b) = \prod_{i=1}^{m} \begin{cases} 0, & \text{if } a_i^L > b_i^U \\ 1, & \text{if } a_i^U < b_i^L \\ \dfrac{1}{a_i^U - a_i^L} \displaystyle\int_{y=\min\{a_i^L, b_i^L\}}^{b_i^L} dy \\ + \displaystyle\int_{y=\max\{a_i^L, b_i^L\}}^{\min\{a_i^U, b_i^U\}} \dfrac{b_i^U - y}{b_I^U - b_i^L} \, dy, & \text{otherwise.} \end{cases} \quad (3)$$

Three different cases can be distinguished from the equation. The solution a does not dominate b $(P(a \succ b) = 0)$ if at least one lower bound of the property intervals a_i^L is larger than the corresponding the upper bound b_i^U. Second, the solution a dominates b $(P(a \succ b) = 1)$, if the upper bound of all the property interval a_i^U are smaller than the lower bounds b_i^L for all objectives. In the third case, a dominates b with a certain probability $P(a \succ b) \in]0, 1[$, if for all objectives i the lower bound a_i^L is smaller than b_i^U and at least one bound a_i^U is larger than b_i^L.

Assuming that the values for a and b, obtained by test functions or real applications, are in the middle of the property intervals and both intervals are of size 2δ, the interval bounds can be computed as $a_i^L = a_i - \delta$, $a_i^U = a_i + \delta$, $b_i^L = b_i - \delta$ and $b_i^U = b_i + \delta$ and (3) can be rewritten as

$$P(a \succ b) = \prod_{i=1}^{m} \begin{cases} 0, & \text{if } a_i > (b_i + 2\delta) \\ 1, & \text{if } a_i < (b_i - 2\delta) \\ \dfrac{1}{2\delta}(b_i - a_i + \delta) \\ + \dfrac{1}{8\delta^2} \operatorname{sgn}(a_i - b_i)(a_i - b_i)^2, & \text{otherwise} \end{cases} \quad (4)$$

where sgn is the signum function. With the probability of dominance, solutions are nondominated with a certain probability, making modifications of the archive update necessary. For each solution $a(i)$, the mean probability R of being dominated by a solution $a(j)$ is computed by

$$R(i) = \frac{1}{N-1} \sum_{j \in \{P \cup A\} : j \neq i} P(a(j) \succ a(i)) \qquad (5)$$

where N is the number of solutions of the unification of the population P and the archive A.

Here, a simplification of Teich's update of the archive is used. First, the current population P is added to the archive A. Then, all solutions with $R(i) > \alpha$ are removed from the archive. For increasing the parameter α, more solutions are added to the archive and the archive changes from the nondominated front to a fuzzy nondominated front. This approach corresponds with the results of Arnold and Beyer [1]. They computed the progress rates of the (μ, λ) evolution strategy for noisy single objective problems and found that selecting a set of μ parents out of λ individuals leads to a higher convergence speed than just selecting the best individual. This observation is in contrast to the noise-free case, where selecting the best solution leads to the highest convergence speed. The (μ, λ) strategy harmonizes with the fuzzy nondominated front.

For better comparison, we use the standard clustering algorithm of SPEA. This is valid, since the core aspect of the ESPEA is the concept of a dominance probability and not the clustering. The fitness is assigned in two steps. First, the fitness S of the archive solutions is computed as

$$S(i) = \frac{1}{N+1} \sum_{j \in \{P \cup A\}} P(a(i) \succ a(j)). \qquad (6)$$

The fitness of a solution in the population is equal to one plus the fitness of the archive solutions, by which it is dominated with a probability larger than a threshold α. ESPEA contains several strategy parameters, which are the threshold α and the size of the property intervals. A drawback is the necessary knowledge of the interval sizes *a priori* of the optimization, such that the intervals reflect the size of the noise in the objective functions.

E. Noise-Tolerant Strength Pareto Evolutionary Algorithm

We propose new modifications for SPEA and define this resulting algorithm as the Noise-tolerant Strength Pareto Evolutionary Algorithm (NT-SPEA). In Section III-B, we described a nonelitistic SPEA in order to avoid the risk of getting stuck in outliers of the optimization process. One disadvantage of this algorithm is that noise reduces the selection pressure [17], suggesting that elitism, which is increasing the selection pressure by conserving nondominated solutions, should be used to compensate. To successfully use elitism in a noisy environment, further modifications are needed to ensure fast convergence while maintaining robustness to noise.

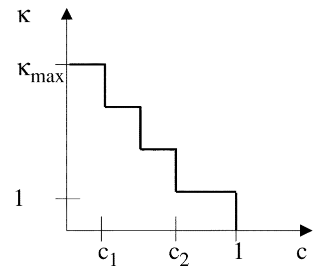

Fig. 1. Dependence of the lifetime κ of an individual on the fraction c of the archive, which it dominates. κ decreases from a maximal value $\kappa_{\max}$, if the individual dominates more than the fraction c_1 until it reaches a lifetime of $\kappa = 1$ at c_2.

We propose three modifications for an extended multiobjective algorithm for noisy environments.

1) *Domination Dependent Lifetime:* In contrast to elitism, which may preserve elite (nondominated) solutions for an infinite time, a lifetime κ is assigned to each individual. For evolution strategies, algorithms with implemented lifetime κ are referred to as (μ, κ, λ) algorithms [2]. In this work this concept is extended to multiple objectives such that the lifetime is variable and related to the dominance of a solution. The lifetime is shortened, if the solution dominates a major part of the archive. This limits the impact of a solution and safeguards against outliers.

2) *Reevaluation of Solutions:* It is common to delete solutions with expired lifetime. We propose to reevaluate archive solutions with expired lifetime and add them to the population. This enables good solutions to stay in the evolutionary process, but their objective values will change due to the noise in the reevaluation.

3) *Extended Update of the Archive:* The SPEA algorithm updates the archive always by adding the current population to the archive and removing the dominated solutions. We extend the update to *all* solutions with nonexpired lifetime. This hinders loss of information, since solutions which were removed by clustering or domination may reenter the archive.

With these features NT-SPEA uses the advantage of an archive as convergence accelerator, but it reduces the risk induced by outliers.

The dominance-dependent lifetime of an individual is assigned according to Fig. 1. The lifetime is measured in generations. For dominating less than a fraction c_1 of the archive A, the maximal lifetime $\kappa = \kappa_{\max}$ is assigned to the individual. For dominating more than a fraction c_2 of A, the minimal lifetime of $\kappa = 1$ is assigned. In-between these two fractions, the lifetime is interpolated in discrete steps of one generation. The dominance-dependent lifetime reduces the impact of a solution. An individual that dominates a large fraction of the archive has a high chance of being selected in the selection process, but is assigned the shortest lifetime.

While the principle of limited lifetime is a key element to remove outliers, the reevaluation allows good solutions to stay in the selection process by reentering the archive. In the case of an outlier, it is not likely, that the reevaluated copy is again an outlier with good objective values and hence it would not reenter the archive. On the other hand, solutions with good design variable settings are likely be nondominated again, if the effect of noise is limited.

The extended update considers the nondominated among all solutions with nonexpired lifetime for the update of the archive. Since the assigned lifetime differs between the solutions, the set of nondominated solutions changes. Dominated solutions become nondominated, if the lifetime of their dominator expires. This is especially important if a noisy solution or an outlier dominates a large fraction of the archive. The dominated solutions are then removed from the archive. The noisy solution or outlier is assigned a short lifetime. After the lifetime expires the removed nondominated solutions may be reselected to the archive. With the original update of SPEA, their information is lost. After the update of the archive, the clustering algorithm of SPEA is used in order to get a limited number of uniformly distributed archive solutions. Solutions of the population and archive participate in the selection process.

With these three modifications, the noise-tolerant SPEA is given by the following.

Algorithm 2
1. **begin**
2. Generate an initial population P of random individuals and an empty archive A.
3. Define a maximal lifetime $\kappa_{\max}$ for individuals (in generations).
4. Evaluate the objectives of the individuals in P.
5. **while** termination criterion is not fulfilled **do**
6. Assign lifetime: Compute for each individual in P the fraction of the archive that it dominates. The lifetime κ of the individual is inverse proportional to the fraction (see Fig. 1).
7. Update A: Remove all solutions from A and refill it with all solutions, whose lifetime is not expired. Then remove all dominated solutions. Limit the size of A by clustering.
8. Fitness assignment: Assign fitness to the individuals in P and A.
9. Selection: Use tournament selection for selecting the parent population P_p from $P \cup A$.
10. Recombination: Generate a new population P by recombination of the individuals in P_p.
11. Mutation: Mutate the individuals in P.
12. Reevaluation: Select the solutions from A with expiring lifetime and add a copy for reevaluation to the population P
13. P is the population of the next generation.
14. Evaluate the objectives of the individuals in P.
15. **end while**
16. **end**

IV. Performance Comparison

A. Generation of Test Functions

A wide variety of noise-free test problems for multiobjective optimization can be found in the literature. A number of review articles have been listed by van Veldhuizen and Lamont [23] and Deb [5]. From Deb, a two-objective minimization problem for an arbitrary number of design variables $x_{1,\ldots,n}$ is chosen and implemented as the first noise-free **test function 1**

$$f^{(1)} = \begin{bmatrix} f_1^{(1)} \\ f_2^{(1)} \end{bmatrix} = \begin{bmatrix} x_1 \\ \dfrac{1}{x_1}\left(1 + \displaystyle\sum_{j=2}^{n} x_j^2\right) \end{bmatrix} \tag{7}$$

with $x_1 \in [0.5;\, 2]$ and $x_{2,\ldots,n} \in [-2.0;\, 2]$. For the *experimental noise*, we assume a normal distribution with zero mean and standard deviation σ_N. A noisy test function is generated by adding this noise to test function 1, leading to **test function 2**

$$f_i^{(2)} = f_i^{(1)} + N(0,\, \sigma_N^2), \qquad i = 1,\, 2 \tag{8}$$

where $N(0,\, \sigma_N^2)$ is a normally distributed random number with zero mean and standard deviation σ_N. The standard deviation is set to $\sigma_N = 0.8$ and the random number is computed separately for each objective and individual in the evolution.

A second type of noise was introduced in Section III-A as the random occurrence of *outliers*. For the modeling in a test function, we define a probability p_o for the occurrence. Since we consider the minimization of positive functions, reducing the objective value has a stronger influence on the optimization process by giving a solution a higher chance to survive. Therefore we divide the objective value by a factor of ten, if an outlier occurs. The large factor is chosen in order to produce a significant change in the objective value. In mathematical form, we define **test function 3** as

$$f_i^{(3)} = \begin{cases} \frac{1}{10} f_i^{(1)}, & \text{if } p < p_o,\, p \in U(0,\, 1) \\ f_i^{(1)}, & \text{otherwise,} \end{cases} \qquad i = 1,\, 2 \tag{9}$$

where $U(0,\, 1)$ is a uniform distribution of random numbers within the interval [0, 1]. The probability of an outliner is small and set to $p_o = 0.01$.

For analyzing the scaling of the optimization algorithms over the number of objectives, a three-objective test function is defined as **test function 4**, which is a generalization of the sphere model to multiple objectives [16]

$$f_i^{(4)} = (1 - x_i)^2 + \sum_{j=1,\, j \neq i}^{n} x_j^2, \qquad i = 1,\, 2,\, 3 \tag{10}$$

with $x_{1,\ldots,n} \in [-2.0;\, 2]$. Analog to the generation of test functions 2 and 3, we add the *experimental noise* and *outliers* to test function 4 and obtain **test function 5**

$$f^{(5_i)} = f_i^{(4)} + N(0,\, \sigma_N^2), \qquad i = 1,\, 2,\, 3 \tag{11}$$

and **test function 6**

$$f^{(6_i)} = \begin{cases} \frac{1}{10} f_i^{(4)}, & \text{if } p < p_o,\, p \in U(0,\, 1) \\ f_i^{(4)}, & \text{otherwise,} \end{cases} \qquad i = 1,\, 2,\, 3. \tag{12}$$

B. Performance Measures

In order to compare different optimization algorithms on the six test functions, performance measures are needed. In multi-objective optimization, the definition of the quality of an optimization usually considers two different aspects. The quality is dependent on the convergence speed of the optimization as well as on the wide and uniform distribution of the solutions along the Pareto front. This is different from single objective optimization where convergence is sufficient, since there exists a single global optimum.

In literature several performance measures are proposed. Van Veldhuizen and Lamont [24] present an overview with performance measures in the design and objective space. Since the test functions contain noise in their objective functions, measuring the performance in objective space is difficult. Instead the performance of the optimizer is measured in design space. Here, the performance measure P is defined as the distance in design space of evaluated solutions to the analytical Pareto front.

To evaluate this performance measure, ten points $x'(k),\ k = 1, \ldots, 10$ are distributed uniformly in design space along the analytical Pareto front. To each point $x'(k)$ the closest of all solutions $x(j)$ of an optimization run is searched and the distance is computed. The mean of the resulting ten distances is taken as performance measure P

$$P = \frac{1}{10} \sum_{k=1}^{10} \min_j (\|x(j) - x'(k)\|). \qquad (13)$$

For the test functions 1–3 the analytical Pareto front is given by $x_1 \in [0.5;\ 2]$ and $x_{2,\ldots,n} = 0$ [5]. The ten uniformly distributed points are

$$x'_1(k) = \frac{1}{2} + \frac{1}{6}(k-1), \qquad x'_{2,\ldots,n}(k) = 0. \qquad (14)$$

The analytical Pareto front of test functions four is convex. Thus, it can be computed by performing a weighted-sum aggregation of all objectives into one function. The derivation of this function with respect to all variables x_j leads to n equations. An elimination of the weighting factors from this set of equations leads to the analytical Pareto front, given by

$$x_1 + x_2 + x_3 = 1,\ x_{4,\ldots,n} = 0, \qquad \text{with } x_{1,2,3} \geq 0. \quad (15)$$

Ten approximately uniform distributed points on the Pareto front of test functions 4–6 are obtained by computing all combinations of $x'_{1,2,3} \in [0,\ 1/3,\ 2/3,\ 1]$, such that (15) is fulfilled.

C. Performance Analysis of Original and Modified Algorithms

In the following, the performance of the algorithms introduced in Section III are numerically analyzed on the six test functions. For all optimization algorithms, a parent and child population of $\mu = \lambda = 60$ is used, with an archive size of 20 for the two-objective test functions 1–3 and a size of 50 for the three-objective test functions 4–6.

The recombination and mutation operators of Section III-B are used. The number of design variables n is set to $n = 7$. This number is equal to the number of design variables of the burner optimization problem, which is addressed in the next section.

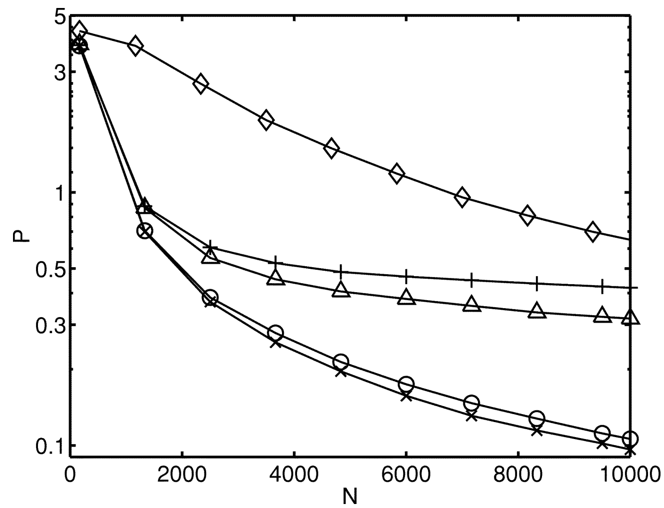

Fig. 2. Convergence of the NT-SPEA [circular symbol] on the noise-free test function 1, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond], and ESPEA [triangle].

Since evolutionary algorithms are stochastic algorithms, the result of 100 optimization runs is averaged for each test function.

Some of the analyzed algorithms contain heuristic parameters. No heuristic parameters have to be set for SPEA, the nonelitistic SPEA and the statistical SPEA. For NT-SPEA, the fractions c_1 and c_2 are set to 0.1 and 0.3, respectively, and a maximal lifetime $\kappa_{\max} = 4$ is used. A discussion of these settings is introduced in the next section. For ESPEA, a performance analysis is made for all combinations of $\alpha \in$ [0.008, 0.01, 0.015, 0.02, 0.04, 0.07, 0.1, 0.2, 0.5] and a property interval size of $(a_i^U - a_i^L) = 2\delta \in$ [0, 0.2, 0.4, 1.0, 2.0, 3.0, 4.0]. In average, the best results of ESPEA on all test problems is obtained with $\alpha = 0.04$ and $\delta = 0.2$.

For the two-objective and noise-free test function 1, the results are given in Fig. 2. The performance measure P is plotted in a logarithmic scale over the number of evaluated solutions N. The measure P reflects distance of the optimization to uniformly distributed points along the analytical Pareto front. In the beginning of the optimization run, P drops rapidly and levels off at the end of the run. The optimization levels off, since a limited population and archive size cannot exactly approximate the Pareto front, thus the distance to the uniform distributed Pareto points stagnates at a certain level.

The performance of the different algorithms varies significantly. The slowest convergence is observed for the statistical SPEA. Evaluating a solution is seven times more expensive than for the other algorithm, since the mean of seven function evaluations is computed (Section III-C). Within the same number of computed solution the statistical SPEA proceeds just by 1/7 of the number of possible generations.

The second slowest is the nonelitistic SPEA, due to the lack of an archive for storing the nondominated solutions. ESPEA shows better performance since the algorithm contains an archive. In addition to the original SPEA, the archive can contain a fraction of dominated solutions. Increasing α and the property interval size raises this fraction and the selection pressure decreases. The best performance can be found for NT-SPEA and the original SPEA. In contrast to the ESPEA, for

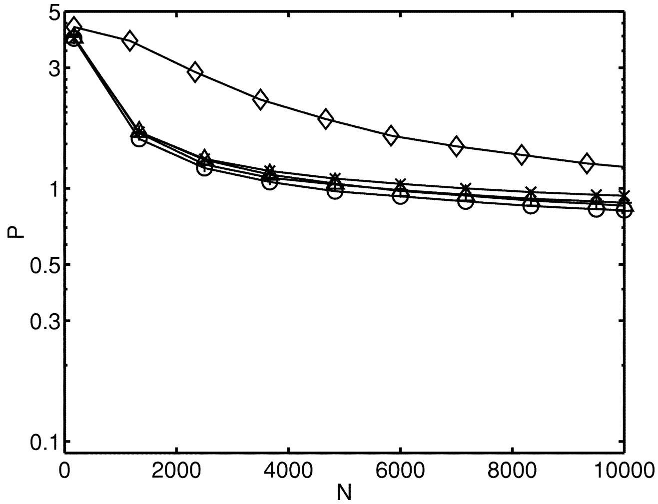

Fig. 3. Convergence of the NT-SPEA [circular symbol] on test function 2 with normally distributed noise, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond], and ESPEA [triangle].

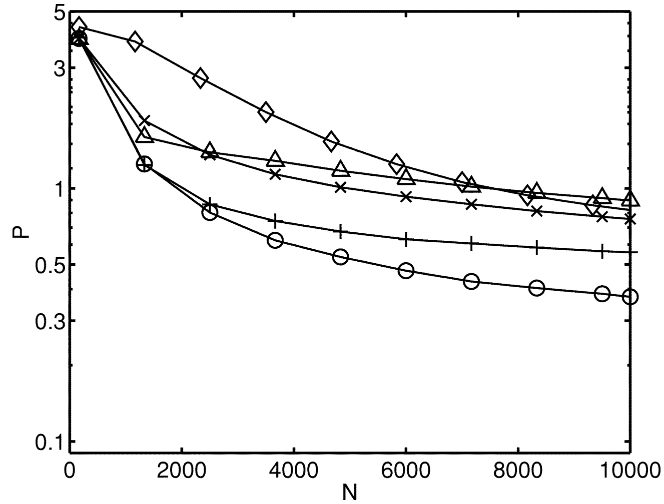

Fig. 4. Convergence of the NT-SPEA [circular symbol] on test function 3 with outliers, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond], and ESPEA [triangle].

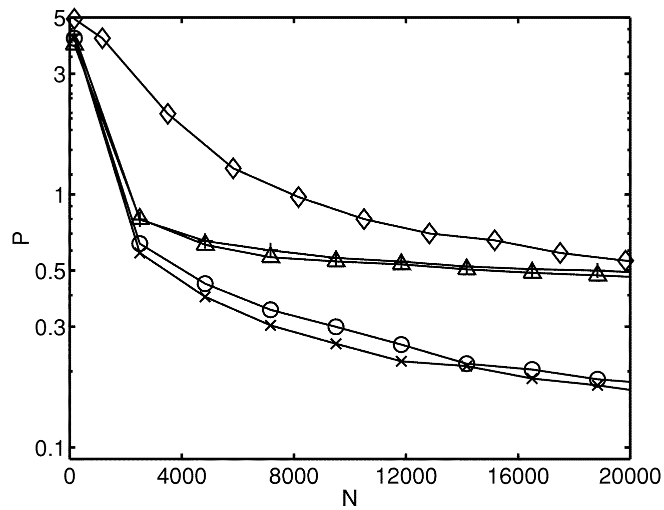

Fig. 5. Convergence of the NT-SPEA [circular symbol] on the noise-free test function 4, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond], and ESPEA [triangle].

which the archive can contain dominated solutions, the archive of NT-SPEA and the original SPEA contain just nondominated solutions and thus a higher selection pressure. According to the theoretical analysis of Arnold and Beyer [1], high selection pressure is an advantage on noise-free and unimodal functions.

The NT-SPEA reevaluates solutions, although this is not necessary for a noise-free test function. Since the fraction of reevaluated solutions is small, however, this disadvantage is small and the algorithm performs well even on noise-free test problem.

Test function two includes normally distributed noise. The standard deviation is set to $\sigma_N = 0.8$ and is about the same magnitude as the objective values of the analytical Pareto front, which are within 0.5 and 2. The convergence behavior of the different algorithms is illustrated in Fig. 3. The convergence speed for the noisy test function is drastically reduced compared to the noise-free test function 1 and the convergence levels off at a higher value of P. Excluding the statistical SPEA, the difference in performance between the algorithms is smaller compared to test function 1. Here, elitism in form of the original SPEA is a disadvantage. The nonelitistic SPEA performs superior to the original SPEA. ESPEA converges about equally to the nonelitistic SPEA. NT-SPEA converges best, due to the compromise between using an archive and limiting the risk of getting stuck in noisy solutions by a limited and dominance-dependent lifetime of solutions.

For the test function 3, an error probability of $p_o = 1\%$ per objective is defined. For this two-objective problem, the probability that at least one objective contains an error is therefore about 2%. In other words about one individual in the population of 60 individuals contains an error and is thus an outlier.

The results of the numerical analysis are given in Fig. 4. Again, NT-SPEA performs best and the nonelitistic SPEA performs better than the original SPEA. Analysis of the convergence of the original SPEA shows that the algorithm gets stuck in the outliers. Outliers occur with a small probability and it is unlikely that they are removed from the archive. This explains why the nonelitistic SPEA performs significantly better than the original one. ESPEA shows no advantage for this test function, compared to the original SPEA.

The performance of the NT-SPEA is superior to all other algorithms. It avoids getting stuck in outliers. The shortest lifetime is assigned to outliers, which dominate a large part of the Pareto front. Since they are reevaluated after their lifetime has expired and the probability that an error occurs again is low, they will be removed from the archive. This allows solutions with larger lifetime than the outliers to reenter the archive after the outlier is removed.

The test functions 4–6 contain three objectives. Obtaining a solution of the same quality as for the two-objective test functions 1–3 in terms of the performance measure P needs noticeable more iterations. The convergence tendencies between the different algorithms are still comparable to the two-objective test functions. Especially the relative convergence of the different algorithms on the noise-free test function 4 (Fig. 5 is similar test function 1). SPEA and NT-SPEA perform demonstratively best on this function.

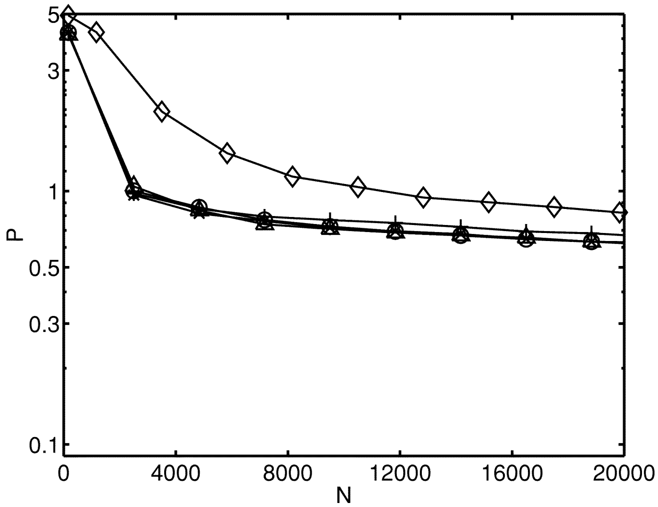

Fig. 6. Convergence of the NT-SPEA [circular symbol] on test function 5 with normally distributed noise, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond], and ESPEA [triangle].

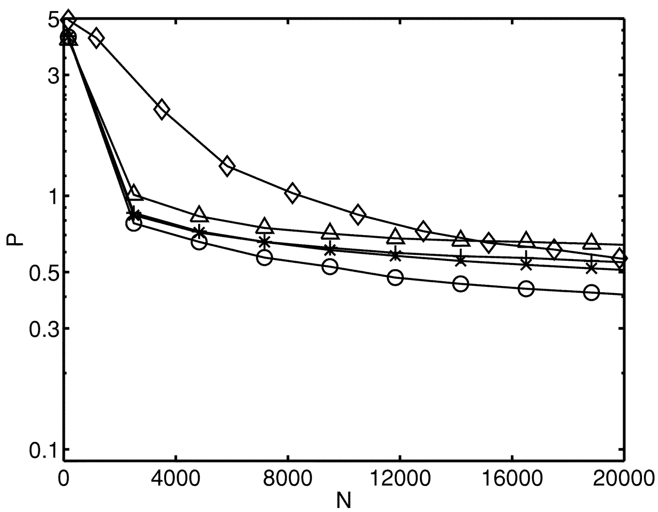

Fig. 7. Convergence of the NT-SPEA [circular symbol] on test function 6 with outliers, compared with the original SPEA [cross symbol], the nonelitistic SPEA [plus symbol], the statistical SPEA [diamond] and ESPEA [triangle].

NT-SPEA, ESPEA, and the nonelitistic SPEA show equal convergence on the noisy test function 5, as illustrated in Fig. 6. The differences are within the sampling tolerance. Slightly inferior convergence is obtained with the original SPEA, demonstrating again the disadvantage of elitism in form of an archive of nondominated solutions with infinite lifetime.

Test function 6 contains similar to test function 3, an error probability of 1% per objective. For the three-objective problem, the probability that an individual contains an error in at least on objective is about 3%, thus about two of the 60 individuals in a population are outliers. The convergence of the different algorithms is plotted in Fig. 7. Similar to test function 3, NT-SPEA performs best, but here the original SPEA performs slightly superior than the nonelitistic SPEA.

Summing the results from the six test functions, we found that elitism, implemented by the archive of the original SPEA is a convergence accelerator for noise-free problems. For noisy problems it is a disadvantage and the nonelitistic SPEA performs in average better.

The relative behavior of the different algorithms shows similar tendencies for two and three objectives. For three objectives, however, the differences are smaller.

The statistical SPEA includes the drawback of multiple function evaluation per solution and is except for test functions 3 and 6 slower than all other implementation in the considered number of function evaluations N, but will perform better for larger values N as indicated by the largest slope in P for larger N, especially for test functions 3 and 6. Again, the differences are smaller for three objectives.

The settings of ESPEA for α and δ are very problem dependent and lead to large performance differences. The best convergence for the noise-free test function 1 is obtained for $\alpha = 0.008$ and $\delta = 0$, a setting which leads to an algorithm and convergence similar to the original SPEA. For test function 2, increasing α to 0.04, but keeping a property interval $\delta = 0$ leads to the best result. Increasing α introduces dominated solutions to the archive. A positive effect of a property interval $\delta > 0$ for the noisy function could not be found. Test function three contains outliers and the ideal settings are $\alpha = 0.2$ and $\delta = 1.5$. These settings differ tremendously from the previous two settings, especially in the property interval, but the performance on this function is still poor. In addition, compared to the other algorithms, ESPEA performs better for two objectives than for three objectives.

In contrast, a marginal problem dependence is found the parameters c_1, c_2 and $k_{\max}$ of NT-SPEA. This is analyzed in more detail in the next section.

Comparing the mean behavior of the algorithms over all test functions, NT-SPEA performs clearly best. One possibility for a mean performance analysis for all six test functions is obtained by summing the minimal value of P for each algorithm over all test function. NT-SPEA clearly results in the smallest value with $\sum_{i=1}^{6} P_i(N = \max) = 1.75$, where $\max = 10\,000$ for test functions 1–3 and $\max = 20\,000$ for test functions 4–6. NT-SPEA is followed by the original SPEA (1.97), ESPEA (2.02) and the nonelitistic SPEA (2.17) and finally the statistical SPEA (3.21).

D. Discussion of the Heuristic Parameters c_1, c_2 and $k_{\max}$

The NT-SPEA algorithm, which is described in Section III-E includes the heuristic parameters c_1, c_2 and $k_{\max}$. Such parameters are often set by experimental analysis of various settings on different test functions. We proposed to set the parameters as $c_1 = 0.1$, $c_2 = 0.3$ and $k_{\max} = 4$. The guiding concepts behind the settings are the following: The value for the maximal lifetime $k_{\max}$ is a trade-off between noise-free and noisy test functions. For noise-free functions, reevaluating does not lead to new information, since the reevaluated solution equals the original. Thus, a larger maximal lifetime (and increased values for c_1 and c_2) is preferable avoiding the reevaluation of solutions.

In contrast, for noisy problems, it is reasonable to limit the lifetime of a solution in the archive, in order to avoid a misleading of the entire optimization process by noisy archive solutions. Here, we store a solution in the archive for at most four generations. The time has to be short enough to avoid that the optimization is misled by very noisy archive solutions (outliers).

TABLE I
SENSITIVITY ANALYSIS OF NT-SPEA ON THE HEURISTIC PARAMETERS c_1, c_2 AND $k_{\max}$. NT-SPEA SHOWS SMALL PERFORMANCE VARIATION OVER A WIDE RANGE OF PARAMETER SETTINGS

test function	result	Heuristic Parameters c_1	c_2	k_{max}
1	min(P)=0.099	0.10	0.20	2
	max(P)=0.113	0.15	0.30	8
2	min(P)=0.804	0.10	0.20	4
	max(P)=0.835	0.05	0.10	2
3	min(P)=0.346	0.10	0.20	4
	max(P)=0.661	0.20	0.50	8
4	min(P)=0.174	0.05	0.15	4
	max(P)=0.218	0.10	0.20	2
5	min(P)=0.345	0.10	0.20	4
	max(P)=0.449	0.10	0.15	2
6	min(P)=0.583	0.10	0.30	8
	max(P)=0.669	0.10	0.15	2

In addition the time has to be larger than one generation, since solutions should be able to reenter the archive after an outlier, which dominates these solutions, is removed after his shorter lifetime has expired. We assume that a solution, which dominates less than 10% of the archive ($=c_1$), should be assigned the maximal lifetime $\kappa = \kappa_{\max}$, while a solution, which dominates more than 30% ($=c_2$) should be reevaluated already in the next generation.

The following parameter analysis underlines that the parameter settings are robust and their influence on the algorithm performance is minor over a large range. The performance analysis of Section IV-C is repeated with all possible combinations of c_1, $c_2 \in$ [0.05, 0.1, 0.15, 0.2, 0.3, 0.5] and $k_{\max} \in$ [2, 4, 8], while the constraint $c_1 < c_2$ is observed. For all combinations and all test functions, the performance measure P was computed as the mean of 100 independent runs. Table I contains the obtained performance measures $\min(P)$ and $\max(P)$ for the best and worst parameter combination, respectively, and the referring heuristic parameters for all test functions.

For the noise-free test functions 1 and 4, all settings performed almost identical and all settings performed better than the nonelitistic SPEA, the statistical SPEA and ESPEA. Reevaluation is not necessary, since the original and reevaluated solution are identical. Thus, reevaluating many solutions will decrease the performance. Beneath influencing the number of reevaluated solutions, the maximal lifetime $k_{\max}$ has a second effect. Since the archive is updated with all solutions with nonexpired lifetime, solutions may reenter the archive after they were removed by clustering. This seems to have a negative effect on the noise-free function, since one setup with $k_{\max} = 8$ performed worst.

Differences in the performance are also small for the test functions 2 and 5, which contain experimental noise. In general, on these two test functions the differences between the different implementations of SPEA are the smallest.

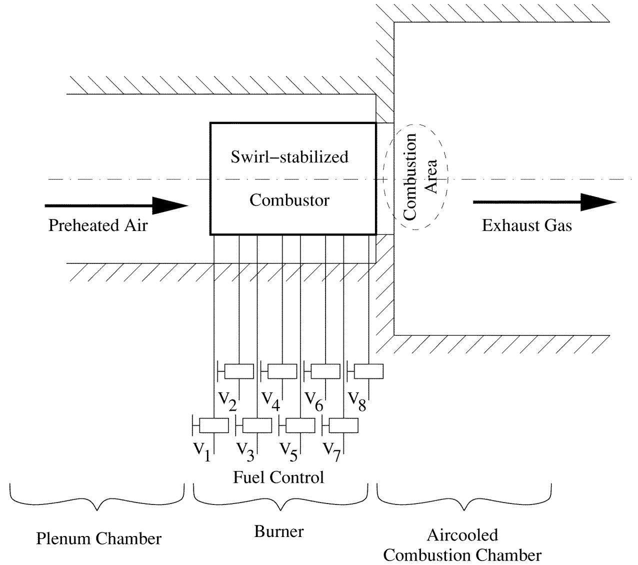

Fig. 8. Sketch of the atmospheric combustion test-rig with a low-emission swirl stabilized burner. The fuel flow through the injection holes are the design variables of the setup. The NO_x emissions and the pulsation of the burner are the objectives to be minimized.

Test functions 3 and 6 contain a small percentage of outliers. This seems to have a major effect on the performance of the different algorithms. Since SPEA performs poor on this problem, the setting of the NT-SPEA algorithm, which is closest to SPEA, performs worst in this comparison. Due to the large maximal lifetime $k_{\max} = 8$ together with the large values $c_1 = 0.2$, $c_2 = 0.5$, the algorithm is in danger of getting stuck in outliers with a long maximal lifetime, thus misleading the algorithm.

Summarizing the results of for all test functions, the heuristic parameters c_1, c_2 and $k_{\max}$ can be set general enough in order to perform well on noise-free and noisy problems, as well as problems with a rare occurrence of outliers. Varying the settings over a large range has minor effect on the performance. Beneath the better performance, this is a major advantage to the ESPEA algorithm, which is very sensitive on the settings of the heuristic parameters.

V. OPTIMIZATION OF A BURNER IN A GAS TURBINE COMBUSTION TEST-RIG

A. *Atmospheric Combustor Test-Rig*

Gas turbines operate by compressing air in a compressor, which then reacts with fuel in a combustion chamber and is finally expanded in a turbine. The difference in power between the turbine output and the compressor input is the net power to generate electricity. The combustion chambers of Alstom's larger gas turbines, e.g., GT24 and GT26, are annular around the turbine axis with a set of burners aligned in the annulus.

We consider the optimization of a single burner in an atmospheric test-rig as illustrated in Fig. 8. Preheated air enters the test-rig from the plenum chamber and is mixed with fuel in the low-emission burner by swirl. The burner stabilizes the combustion flame in a predefined combustion area by a controlled vortex breakdown. The fuel is natural gas or oil and is injected through injection holes, which are uniformly distributed along the burner. A detailed description is given by Jansohn *et al.* [15].

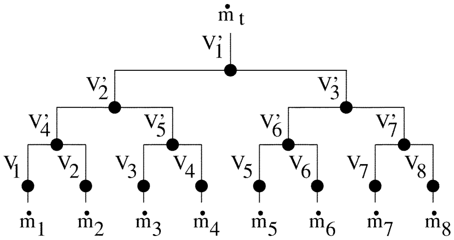

Fig. 9. Encoding of the fuel flow $\dot{m}_i$ through the eight valves V_i of the test-rig. Since the total mass flow $\dot{m}_t$ is fixed, the eight fuel flows can be encoded by seven virtual valves V_j'.

Various investigations have been made in order to reduce pulsations and emissions of the burner by active and passive control mechanisms. Paschereit *et al.* [19] reduced the pulsations in the experimental test-rig by an acoustic actuation in a closed control loop. We consider a passive control mechanism, choosing the fuel flow rates through the injection holes of the burner as design variables of the setup, due to the low modification cost for the gas turbine compared to an active control system. Eight continuous valves $V_{i,\, i=1,\, \ldots,\, 8}$ are used to control the fuel rates. Each valve V_i controls the mass flow $\dot{m}_i$ through a set of adjacent injection holes along the burner axis.

In order to keep the operating conditions constant, the total fuel mass flow $\dot{m}_t = \sum_{i=1}^{8} \dot{m}_i$ is fixed, reducing the number of free design variables for the optimization from eight to 7. Fig. 9 shows the implemented encoding for the eight values V_i by seven virtual valves $V'_{j,\, j=1,\, \ldots,\, 7}$. The total mass flow is split by a first virtual valve V_1' into two flows, with each of the flows feeding either the first or second half of the real valves. The next layer consists of two virtual valves V_2' and V_3' and splits the two flow into four. Finally, the virtual valves V_4', V_5', V_6', and V_7' feed the real valves V_i and determine the fuel flows $\dot{m}_i$. While the evolutionary algorithm operates with the seven virtual valves, the real valves are used in the test-rig. A detailed description about the experimental setup and the fuel control can be found in [7]. In the following, we refer to the real valves V_i and the real fuel flows $\dot{m}_i$.

The NO_x emissions and the pulsation of the burner are the two objectives to be minimized in a Pareto optimization setup. Pulsations are thermo-acoustic combustion instabilities, involving feedback cycles between pressure, velocity and heat release fluctuations. The NO_x emissions occur at high combustion temperature, which arise in centers of rich combustion due to inhomogeneous mixing of fuel and air. No constraints are imposed on the objective functions.

B. Optimization Results

An optimization run is performed using NT-SPEA with a population and archive size of 15 and evaluating a total of 326 different burner settings within one working-day. All solutions are plotted in Fig. 10 in order to show the possible decrease in NO_x emissions and pulsations by the optimization compared to the given standard burner configuration and between the best and worst designs.

The given standard burner configuration is marked in the figure and represents a setting with equal mass flow through all valves. Some solutions found by the optimization process dominate the standard configuration, i.e., are superior in both

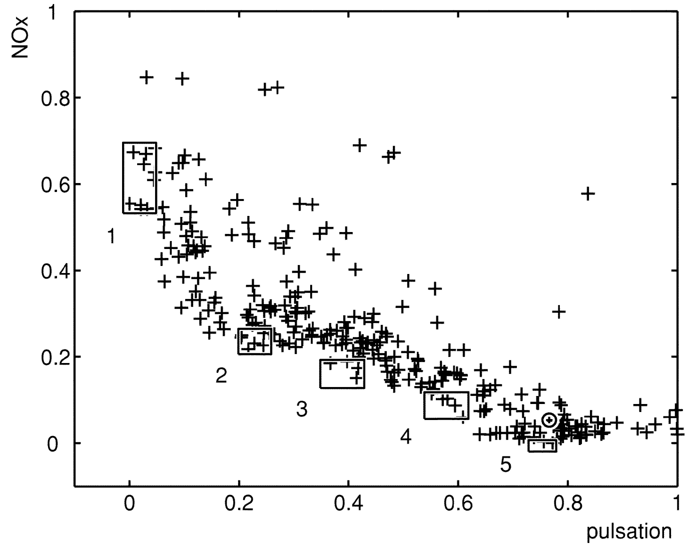

Fig. 10. All measured solutions of the burner optimization run [plus symbol] and given standard burner configuration [circular symbol]. Five boxes mark different areas along the nondominated front.

objectives. Thus the optimization run is successful, delivering improved solutions for both objectives. The occurrence of a wide nondominated front underlines the conflict in minimizing both objectives and just (Pareto) compromise solutions can be found.

In the figure, the objectives are noisy. Thus, drawing just the nondominated front and picking one solution from the front is risky from the point of view, that an inferior solution is picked, which is nondominated due to the noise in its objective values. Picking an area close to the nondominated front increases the confidence in the front, especially if the valve settings are quite similar for the solutions in the area. A second reason for not drawing just the nondominated front is the possible shift of the front toward smaller objective values. The objectives contain noise and the selected nondominated solutions may improve due to noise leading to smaller objective values. In addition, we are more interested in the valve settings than in the exact objective values, since the valve settings indicate the included physics.

Five areas along the nondominated front are picked and marked by boxes. For the solutions within the boxes, the valve settings are printed in Fig. 11. Fig. 8 shows the arrangement of the valves in the combustor. For better illustration, the settings are connected with a line and the dash-dotted line shows the standard burner configuration with equal mass flow through all valves. Within each box, the settings of the different solutions are in deed quite similar.

Box one and five are at the extreme ends of the Pareto front. Box one represents Pareto solutions with high NO_x emissions, but low pulsation. The corresponding valve settings show an increased fuel mass flow at valves one, two, and four, while the flow at valves five and six is reduced. The fundamental mechanism corresponding to these settings is the fact that the increased mass flow through valves one and two leads to rich combustion in the center of the burner. The rich combustion zone stabilizes the combustion like a pilot flame, but increases the NO_x emissions. The lean zones are close to the middle of the burner at valves five and six.

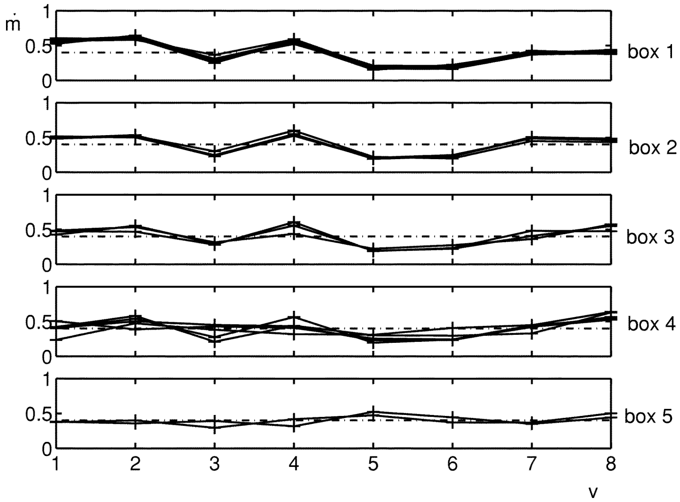

Fig. 11. Mass flow $\dot{m}$ through the valves $V_{i,\, i=1,\dots,8}$ for solutions along the nondominated front, marked by five boxes of Fig. 10.

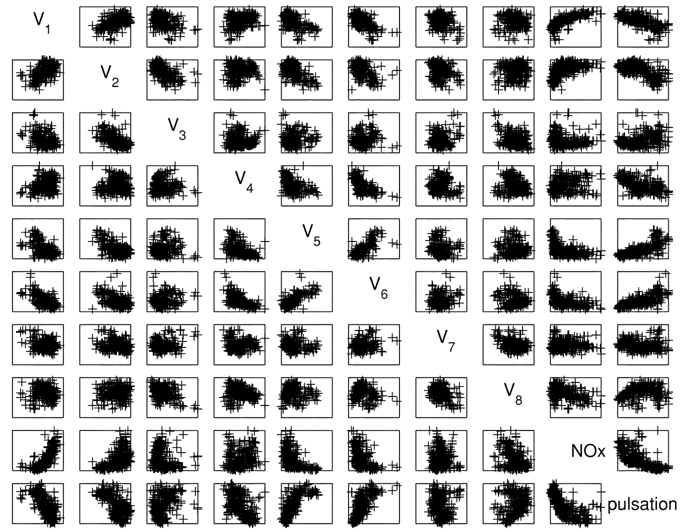

Fig. 12. Scatterplot representing all possible combinations of 2-D plots for the valves $V_{i,\, i=1,\dots,8}$ and the objectives NO_x and pulsation.

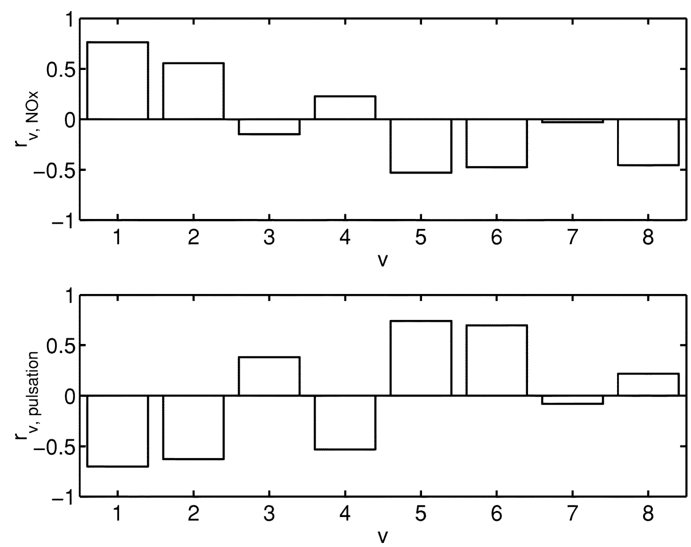

Fig. 13. Correlation coefficient r between the mass flow through the valves $V_{i,\, i=1,\dots,8}$ and the objectives NO_x and pulsation.

Box five contains solutions with minimal NO_x emissions, but high pulsation. The mass flow through each valve is about equal, generating no rich combustion zones. Compared to the standard burner configuration, the small mass flow increase at valves five and eight and decrease at three and four leads to lower NO_x emissions, while the pulsation is unchanged.

C. Statistical Analysis

One of the interesting features of the resulting nondominated front is the almost linear change in valve settings along the front. At Box one, five valves have either strongly increased or decreased mass flow and their amplitude is constantly decreasing from Box one to five until it reaches an almost equal mass flow for all valves in Box five. This indicates simple dependencies of the valves with the objective functions. Fig. 12 contains a scatterplot for the valve settings and objective functions of all measured solutions. A scatterplot contains all possible two-dimensional (2-D) subspace plots for all design variables and objectives. The plot in column nine and row ten contains the objective space with the nondominated front. Most interesting are the two last rows, containing the correlation of the valves with the objective functions. For example, the horizontal and vertical axis of the plot in row nine, column one represent valve one and the NO_x emission, respectively. Strong correlation is expressed by narrow stripes under $\pm 45°$ to the axis. An axially symmetrical area of solutions implies no correlation. Strong correlation can be observed between valves one, two, five, six, and the two-objective functions.

The correlation coefficients $r_{V_i,\, NO_x}$, and $r_{V_i,\, \text{pulsation}}$ for the values and objectives are given in Fig. 13. They complement the results from the scatterplot. For all valves, the correlation coefficients have opposite signs for the two objectives. Therefore, changing the fuel injection in any of the valves improves always one objective while the other is worsened. Large coefficients indicate a strong correlation and occur between valves one, two, five, six and the two objective functions. For increasing the mass flow through valve one and four, the emissions increase while the pulsation decreases. For valves five and six, this is vice versa.

It has to be considered that these observations hold for the solutions obtained through an optimization process. The distribution of the solutions in the scatterplot in Fig. 12 illustrates that they do not cover the whole design space. Hence, these solutions are not uniformly distributed in the design space and may not be representative.

D. Noise Analysis

The NT-SPEA algorithm that is used for the burner optimization contains the special feature of reevaluating solutions after their lifetime expires. Among the 326 evaluated solutions, 40 were reevaluated at least once by the optimizer. Comparing the difference in NO_x between a solution and the reevaluated one, the maximal difference is about 8% of the objective range and the mean difference is 2%. For the pulsation, the maximal and mean difference is 13% and 4%, respectively. Thus, the noise in the pulsation is more critical to the optimization. The large ratio between the maximal and mean difference indicate the rare occurrence of outliers and the presence of noise in the objective measurement of all solutions.

VI. Conclusions

A novel noise-tolerant multiobjective evolutionary algorithm (NT-SPEA) is introduced with increased robustness for applications prone to noise and outliers. The algorithm introduces the concepts of domination-dependent lifetime, the reevaluation of nondominated solutions and an extended update mechanism for the archive. These concepts have been applied to SPEA and can be transferred to any elitistic multiobjective algorithm.

A convergence comparison for various implementations of SPEA has been performed on noisy and noise-free test functions. In general, a decrease in convergence is observed when noise is introduced. The concept of elitism is analyzed in the presence of noise. In the absence of noise, elitism can be used as a convergence accelerator. However, for different types of noise, elitism can imply a significant disadvantage, since the optimization can get misled by outliers.

The NT-SPEA overcomes the problem by introducing dominance-dependent lifetime and accelerates the convergence by using an archive. The archive is modified by the reevaluation of nondominated solutions and an extended update. For the noise-free test problems, NT-SPEA shows similar convergence to the original SPEA, which converges best. This is a major advantage compared to a nonelitistic and a statistical implementation of SPEA and the ESPEA of Teich.

While NT-SPEA performs equal or superior to the best of the other implementations for problems with normally distributed noise, it clearly outperforms all algorithms for problems with outliers. The discussion of the heuristic parameters shows that they have minor influence on the performance over a wide parameter range. A further advantage, which is not discussed in the paper, is that NT-SPEA can handle moving optima over time or changing environmental conditions. The algorithm reevaluates solutions after a limited lifetime, therefore adapts the objective values according to the changing values.

The algorithm is successfully applied to an automated optimization of gas turbine burners. The process produces in an automated fashion an experimental nondominated front for minimizing pulsation and emissions of an industrial burner. Automated optimization can be considered a supporting tool in the design process, complementing physical understanding as well as trial-and-error design. Future work will focus on using larger numbers of valves, leading to more flexibility in the fuel distribution and allowing axially asymmetric distribution. In addition, binary valves (on/off) will be used, reducing the modification cost for adapting a burner in a real machine according to the optimization results. The present algorithm is under modification to account for these discrete configurations.

Acknowledgment

The results were obtained at the atmospheric test-rig of Alstom Power in Baden-Dättwil, Switzerland. For the realization of the experiments the authors wish to thank B. Schuermans. Special thanks to C. O. Paschereit for several useful discussions, as well as for the provision of the test-rig.

References

[1] D. V. Arnold and H.-G. Beyer, "Investigation of the (μ, λ)-ES in the presence of noise," in *Proc. CEC'01 Conf.*, Piscataway, NJ, 2001.

[2] T. Bäck, F. Hoffmeister, and H.-P. Schwefel, "A survey of evolution strategies," in *Proceedings of the Fourth International Conference on Genetic Algorithms and Their Applications*, R. K. Belew, Ed: Morgan Kaufmann Publishers, 1991.

[3] C. A. Coello Coello, "An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends," in *Congress on Evolutionary Computation*, DC: IEEE Service Center, 1999, pp. 3–13.

[4] ——, (2001, Last accessed Oct.) List of references on evolutionary multi-objective optimization. [Online]. Available: http://www.lania.mx/~ccoello/EMOO/EMOObib.html

[5] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *IEEE J. Evol. Comput.*, vol. 7, pp. 205–230, 1999.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Proc. Parallel Problem Solving from Nature VI Conf.*, 2000, pp. 849–858.

[7] R. Dornberger, P. Stoll, Ch. O. Paschereit, B. Schuermans, D. Büche, and P. Koumoutsakos, "Redundanzfreier Ansteuerungsmechanismus der Ventile zur Beeinflussung des Mischungsprofiles für die Kontrolle verbrennungsgetriebener Schwingungen," German Patent 101 04 150.0, ALSTOM Power (Schweiz) AG, 2001.

[8] ——, "Numerisch experimentelle Optimierung des Mischungsprofils für die Kontrolle verbrennungsgetriebener Schwingungen und Emissionen," German Patent 101 04 151.9, ALSTOM Power (Schweiz) AG, 2001.

[9] M. C. Fonseca and P. J. Fleming, "Multi-objective genetic algorithms made easy: Selection, sharing and mating restrictions," in *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. New York: IEEE, 1995, pp. 45–52, no. 414.

[10] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[11] D. E. Goldberg and P. Segrest, "Finite Markov chain analysis of genetic algorithms," in *Proceedings of the Second International Conference on Genetic Algorithms*, Grafenstette, Ed. Mahwah, NJ: Lawrence Erlbaum, 1987.

[12] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.

[13] J. Horn, "Multicriterion decision making," in *Handbook of Evolutionary Computation*, T. Bäck, D. B. Fogel, and Z. Michalewicz, Eds. New York: Oxford Univ. Press, 1997, vol. F1.9, pp. 1–15.

[14] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, Zitzler *et al.*, Eds., Zürich, Switzerland, 2001.

[15] P. Jansohn, T. Ruck, C. Steinbach, H.-P. Knöpfel, and T. Sattelmayer, "Development of the advanced EV (AEV) burner for the ABB GTX 100 gas turbine," in *ASME Turbo Asia 97*, Singapore, 1997.

[16] M. Laumanns, G. Rudolph, and H.-P. Schwefel, "Mutation control and convergence in evolutionary multi-objective optimization," in *Proc. 7th Int. Mendel Conf. Soft Computing*, Brno, Czech Republic, 2001.

[17] B. L. Miller and D. E. Goldberg, "Genetic algorithms, selection schemes, and the varying effects of noise," University of Illinois at Urbana-Champaign, Illinois Genetic Algorithm Laboratory, IlliGAL Report no. 95 005, 1995.

[18] V. Pareto, *Manuale die Economia Politica*. Milano, Italy: Societa Editrice Libraria, 1906.

[19] O. Paschereit, E. Gutmark, and W. Weisenstein, "Structure and control of thermoacoustic instabilities in a gas-turbine combustor," *Combust. Sci. Technol.*, vol. 138, pp. 213–232, 1998.

[20] I. Rechenberg, *Evolutionsstrategie '94*. Stuttgart, Germany: Frommann-Holzboog Verlag, 1994.

[21] N. Srinivas and K. Deb, "Multiobjective optimization using non-dominated Sorting in genetic algorithms," in *Evolutionary Computation*. Cambridge, MA: MIT Press, 1995, vol. 2, no. 3, pp. 221–248.

[22] J. Teich, "Pareto-front exploration with uncertain objectives," in *Proceedings of the First Conference on Evolutionary Multi-Criterion Optimization*, Zitzler *et al.*, Eds., Zürich, Switzerland, 2001.

[23] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective evolutionary algorithm test suite," in *Proceedings of the 1999 ACM Symposium an Applied Computing*, J. Carroll *et al.*, Eds. New York: ACM, 1999, pp. 351–357.

[24] ——, "On measuring multiobjective evolutionary algorithm performance," in *Proc. 2000 Congr. Evolutionary Computation*, 2000, pp. 204–211.
[25] ——, "Multiobjective evolutionary algorithms: Analyzing state-of-the-art," in *Evolutionary Computing*. Cambridge, MA: MIT Press, 2000, vol. 8, no. 2, pp. 125–147.
[26] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto Approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, 1999.
[27] Zitzler *et al.*, *Proc. 1st Conf. Evolutionary Multi-Criterion Optimization*, Zürich, Switzerland, 2001.

Dirk Büche received the Dipl. degree in aerospace engineering from the University of Stuttgart, Germany, in 1999. Since 2000, he has been a Ph.D. student at the Institute of Computational Science (ICoS), Swiss Federal Institute of Technologies (ETH), Switzerland. He is involved in a cooperation of ICoS with Alstom (Switzerland) AG for the optimization of realistic large-scale turbomachinery components. From September 1997 to June 1998, he was a graduate student at the Northwestern University, Evanston, IL.

From November 1999 to April 2000, he worked as a Researcher at the ABB-ALSTOM research center in Dättwil, Switzerland. His current research interests include the development of evolutionary optimization algorithms with a focus on multiobjective problems and response surface techniques for accelerated convergence.

Peter Stoll received the Dipl. degree in aerospace engineering from the University of Stuttgart, Germany, in 1994, and the Ph.D. degree in aerospace engineering in 2000.

From 1999 to 2002, he worked as Scientist, R&D Engineer, and Project Leader at ALSTOM Power Technology Ltd at Baden-Dättwil, Switzerland, in the field of turbomachinery optimization and software engineering. Since 2002 he works at science + computing ag in Tübingen, Germany. In addition, he is a lecturer in computer science at the University of Applied Sciences of Zurich. His fields of activity are software engineering, optimization, computer science, and numerical algorithms.

Rolf Dornberger received the Dipl. degree in aerospace engineering from the University of Stuttgart, Germany, in 1994, and the Ph.D. degree in aerospace engineering in 1998. From 1994 to 1998, he studied business administration for post-graduates at the University of Distance Education Hagen, Germany.

From 1998 to 1999, he worked as scientist and R&D engineer at ABB Corporate Research Ltd., and from 1999 to 2001, as project leader for software engineering at ALSTOM Power Technology Ltd. at Baden-Dättwil, Switzerland. From 2001 to 2002, he was team leader of technology ventures at Atraxis Group AG, Zurich, Switzerland. Since 2001, he has been a Lecturer in computer science at the University of Applied Sciences of Zurich. Since 2002, he has been a Professor in business information systems and information technology at the University of Applied Sciences Solothurn Northwestern Switzerland. His current research interests are advanced computational technologies, their application to engineering and business problems, computer science and software engineering.

Petros Koumoutsakos received the Dipl. degree in naval architecture and mechanical engineering from the National Technical University of Athens, Greece, in 1986, the M.Sc. degree in naval architecture from the University of Michigan, Ann Arbor, in 1987, and the M.Sc. degree in aeronautics and the Ph.D. degree in aeronautics and applied mathematics from the California Institute of Technology, Pasadena, in 1988 and 1992, respectively.

From 1992 to 1994, he was a National Science Foundation Postdoctoral Fellow in parallel supercomputing with the California Institute of Technology. Since 1994, he has been a Senior Research Associate with the Center for Turbulence Research, NASA Ames/Stanford University. From September 1997 to June 2000, he was an Assistant Professor of Computational Fluid Dynamics, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland. Since July 2000 he has been a Full Professor of Computational Sciences at ETHZ. His current research interests are in the areas of multiscale computation and biologically inspired optimization and the application of these techniques to problems of interest in the areas of engineering and life sciences.