Runtime Analysis of an Evolutionary Algorithm for Stochastic Multi-Objective Combinatorial Optimization

Walter J. Gutjahr*

Department of Statistics and Operations Research, University of Vienna,
Universitaetsstr. 5/9, 1010 Vienna, Austria

Abstract

For stochastic multi-objective combinatorial optimization (SMOCO) problems, the Adaptive Pareto Sampling (APS) framework has been proposed, which is based on sampling and on solution of deterministic multi-objective subproblems. We show that when plugging in the well-known Simple Evolutionary Multiobjective Optimizer (SEMO) as a subprocedure into APS, ϵ-dominance has to be used to achieve fast convergence to the Pareto front. Two general theorems are presented indicating how runtime complexity results for APS can be derived from corresponding results for SEMO. This may be a starting point for the runtime analysis of evolutionary SMOCO algorithms.

Keywords: multi-objective optimization, stochastic optimization, combinatorial optimization, evolutionary algorithm, runtime analysis.

1 Introduction

In the recent years, both the fields of *multiobjective combinatorial optimization* (MOCO) and of *stochastic combinatorial optimization* (SCO) have made considerable progress, and numerous articles appeared in these areas. Authors in the MOCO field argue that most real-world problems are characterized by multiple objectives, and authors from SCO emphasize that in real-world situations, decision makers typically face a smaller or larger degree of uncertainty on parameters on which the decisions have to be based. If we join these arguments, we arrive at the conjecture that in a very broad range of applications of decision technology, realistic quantitative models will be characterized by the presence of *both* multiple objectives *and* uncertainty (the latter represented by stochastic models). This makes it desirable that, in particular, the toolkit for combinatorial optimization is extended to methods solving *stochastic multi-objective combinatorial optimization* (SMOCO) problems. Indeed, problems of this type have been described in diverse areas such as financial engineering (Spronk et al., 2005), production planning (Hnainen et al., 2010), transportation logistics (Hassan-Pour et al., 2009), supply chain management (Amodeo et al., 2009), health care management (Baesler and Sepúlveda, 2001), or project management (Gutjahr and Reiter, 2010).

A general *methodology* for tackling SMOCO problems has begun to develop only recently. Both in the MOCO and in the SCO field, evolutionary algorithms (EAs) play an important role, as seen, e.g., from the surveys (Coello Coello, 2006) for MOCO and (Jin and Branke, 2005; Bianchi et al., 2008) for SCO. This suggests an application of EAs also to SMOCO.

*E-Mail: walter.gutjahr@univie.ac.at. Tel: +43 1 4277 38632. Fax: +43 1 4277 38639.

Let us give a short review of available approaches for the solution of SMOCO problems. Concerning the multi-objective part, we focus on the solution concept of determining *Pareto-optimal* (or: *efficient*) solutions, which appears to be best-suited for the purposes of a decision maker who does not want (or is not able) to weigh the different objectives against each other *a priori*. Caballero et al. (2004) deal with stochastic multi-objective *continuous* optimization problems, which differ from the stochastic multi-objective *combinatorial* optimization problems investigated in this paper. Nevertheless, their work is also interesting in our context since they outline the two basic alternative pathways along which a stochastic multi-objective problem can be attacked: Either it is reduced, in a first step, to a *deterministic* counterpart problem, which is then still multi-objective and can be solved by suitable techniques of multi-objective optimization, or it is first reduced to a *single-objective* problem, which is then still stochastic and can be solved by techniques of stochastic optimization. In the present work, we choose the first alternative—called the "multi-objective" approach in (Caballero et al., 2004)—, which seems to be the more common in the literature, but let us mention that also the second alternative—called the "stochastic" approach in (Caballero et al., 2004)—has been applied in the SMOCO area, e.g., by adopting the quality indicator concept for scalarizing the given multi-objective problem, see (Basseur and Zitzler, 2006) or (Liefooghe et al., 2007).

The most frequent approach to the solution of a SMOCO problem consists in using a fixed sample of random *scenarios*[1] in order to reduce the stochastic (multi-objective) problem to a deterministic (multi-objective) one, and then to apply some exact or metaheuristic technique to obtain the set of Pareto-optimal solutions w.r.t. the average objective function values over the sample. If an exact technique is chosen, authors typically apply the so-called ϵ-constraint method for further reducing the multi-objective problem version to a series of single-objective ones, and solve the resulting single-objective combinatorial optimization problems by mathematical programming (MP) methods. Examples for this solution strategy can be found in (Guillén et al., 2005; Cardona-Valdés et al., 2011; Franca et al., 2010). Alternatively, it is also possible to apply a metaheuristic to deal with the multi-objective problem. For an example, see (Claro et al., 2010a; Claro et al.,2010b), where a multi-objective heuristic hybridizing Tabu Search and Variable Neighborhood Search is used for this purpose.

Fixed-sample approaches have the limitation that the probability model specified by the drawn sample scenarios is only an approximation to the original probability model given by the problem formulation. As a consequence, finally, a changed problem is solved. The changed problem may considerably differ from the original one especially in a situation where the high computational effort for solving the resulting MOCO problem only allows the consideration of a relatively small sample of scenarios, which bears the risk that practically relevant scenarios are not represented among the chosen scenarios. In the case where the probability model includes events that are rare but connected with high losses, this may be particularly detrimental. In the area of single-objective sampling-based stochastic optimization, *variable-sample* methods have been used as a possible alternative to fixed-sample methods. Contrary to techniques working with a fixed randomly selected sample, the application of a variable-sample method can be shown to converge, under suitable conditions, to the solution of the original stochastic optimization problem; see, e.g., (Homem-de-Mello, 2003).

Also in the SMOCO field, some papers use variable-sample approaches. Typically, variable-sample techniques for SMOCO rely on modifications of evolutionary algorithms, especially multi-objective genetic algorithms such as the NSGA-II algorithm by Deb et al. (2002). Whereas the first articles of this kind, e.g., (Hughes, 2001; Teich; 2001; Gutjahr, 2005), pursue the variable-sample

[1] In the stochastic programming literature, a scenario is a specific outcome of a random experiment, see (Birge and Louveaux, 1997), p. 50.

idea in a rather empirical way, it has been shown in (Gutjahr, 2009; Gutjahr and Reiter, 2010) that by a suitable iterative procedure called Adaptive Pareto Sampling (APS) exchanging random samples from iteration to iteration, convergence of the currently proposed solution set to the exact set of Pareto-optimal solutions with probability one can be ensured under mild conditions. However, the mentioned articles do not provide analytical results on the *speed* of convergence. This topic is the subject of the present paper.

We shall study an APS version where the Simple Evolutionary Multiobjective Optimizer (SEMO) by Laumanns et al. (2002b) is used for the solution of the occurring deterministic multiobjective subproblems. As a performance measure, we shall use the expected runtime until the exact Pareto front of the SMOCO problem has been found. It will turn out that for obtaining an efficient algorithmic variant, SEMO has to be modified by including the concept of ϵ-dominance, using a sufficiently small value of ϵ. Our theorems show how available runtime results for SEMO on MOCO problems can be translated to runtime results for our APS version on corresponding SMOCO problems. For a recent survey on theoretical analysis of multi-objective evolutionary algorithms in general and SEMO in particular, the reader is referred to Brockhoff (2011).

The plan of this paper is the following: Section 2 presents the investigated APS algorithm. Section 3 introduces the analyzed class of SMOCO problems and explains the sampling technique. In section 4, the way how SEMO is modified by an ϵ-dominance concept and how it is incorporated into APS is described. Section 5 derives the general results in expected runtime and provides examples. The final section 6 gives concluding remarks.

2 Adaptive Pareto Sampling

We focus on the case of two objective functions. However, the algorithm APS described below can be generalized to more than two objectives in a straightforward way.[2] Let a stochastic bi-objective combinatorial optimization problem

$$\text{Maximize } (F_1(x), F_2(x)) \quad \text{subject to} \quad x \in X \tag{1}$$

with $F_i(x) = \mathbb{E}(f_i(x,\omega))$ $(i = 1, 2)$ be given, where X is a finite decision space, and ω denotes the influence of randomness.[3] Maximizing the vector of the two objective functions is understood in the sense of determining the Pareto front of the problem.

We make use of the following customary definitions: For $z_i = F_i(x)$ and $z'_i = F_i(x')$ $(i = 1, 2;\ x, x' \in X)$, x' *dominates* x if $z'_i \geq z_i$ for $i = 1, 2$, and $z'_i > z_i$ for at least one i. In this case, we write $z' \succ z$. Furthermore, x is *nondominated* by some $S \subseteq X$ if there is no $x' \in S$ such that x' dominates x, and x is *efficient* (or: Pareto-optimal) if it is nondominated by X. For abbreviation, we shall write $F(x)$ for the image point $(F_1(x), F_2(x)) \in \mathbb{R}^2$ in objective space to which solution x is mapped by the objective functions. The set X^* of all efficient solutions is the *efficient set*; its image in objective space is the *Pareto front* $F(X^*) = \{F(x) \mid x \in X^*\}$.

The *Adaptive Pareto Sampling* (APS) framework algorithm has been proposed in (Gutjahr, 2009) and further investigated in (Gutjahr and Reiter, 2010). Its purpose is to enable the solution of (1) in the frequently occurring case where the expectation $F_i(x) = \mathbb{E}(f_i(x,\omega))$ cannot be computed directly (or only at high computational costs), but has to be estimated by sampling instead. For

[2] Presumably, also our runtime results in section 5 can be extended to more than two objectives by a rather straightforward generalization of the proofs, but we leave this topic outside the scope of the present paper.

[3] Formally, for each x and i, the function $f_i(x,\cdot)$ is a random variable on a probability space $(\Omega, \mathcal{A}, P)$, and ω is an element of the sample space Ω. Each ω determines a specific realization of the random variables $f_i(x,\cdot)$, i.e., a scenario; hence the notation $f_i(x,\omega)$. As customary in the stochastic programming literature, we shall identify ω with the scenario determined by it (cf. Birge and Louveaux, 1997).

getting an estimate of $\mathbb{E}(f_i(x,\omega))$, APS draws s random *scenarios* $\omega_1,\ldots,\omega_s$ independently from each other. Then, the *sample average estimate* $\bar{f}_i(x)$ of $F_i(x)$ is given by

$$\bar{f}_i(x) = \frac{1}{s}\sum_{\nu=1}^{s} f_i(x,\omega_\nu) \approx \mathbb{E}(f_i(x,\omega)). \tag{2}$$

Evidently, the sample average estimate is an unbiased estimator for $F_i(x)$. An approximation to the solution of the given problem (1) can be computed by solving a related problem where the two expectations forming the objective functions are replaced by the corresponding sample average estimates under a sample of size s. In this way, we obtain the following deterministic bi-objective problem:

$$\text{Maximize } (\bar{f}_1(x), \bar{f}_2(x)) \quad \text{subject to} \quad x \in X. \tag{3}$$

We call problem (3) the *bi-objective sample average approximation* (BSAA) problem corresponding to the original problem (1). The vector $((\bar{f}_1(x), \bar{f}_2(x))$ will shortly written as $\bar{f}(x)$.

In Fig. 1, we present the pseudocode of the APS algorithm. The algorithm is iterative and works with a current solution set $L^{(k)}$ which is updated from iteration to iteration. In each of these iterations, first of all a corresponding deterministic BSAA problem is solved in order to obtain a *proposal* for the solution set. After that, the elements of the solution of the BSAA problem are merged with those contained in $L^{(k-1)}$, the elements in the union of both sets are *evaluated* based on independent samples for each solution and each objective function, and dominated elements (w.r.t. the evaluation results) are eliminated. This yields the new solution set. The sample sizes are controlled by sequences (s_k) and $(\bar{s}_k)$ of positive integers ($k = 1, 2, \ldots$).

Procedure APS

initialize the solution set $L^{(0)}$ as the empty set;
for iteration $k = 1, 2, \ldots$ {
  (a) *solution proposal*:
    draw a sample $\{\omega_1, \ldots, \omega_{s_k}\}$ of s_k scenarios;
    for the drawn sample, determine the Pareto-optimal set $S^{(k)}$ of the BSAA problem with sample size s_k;
  (b) *solution evaluation*:
    for each $x \in L^{(k-1)} \cup S^{(k)}$ and each $i = 1, 2$ {
      draw a sample $\{\omega'_1, \ldots, \omega'_{\bar{s}_k}\}$ of $\bar{s}_k$ scenarios;
      based on this sample, determine an estimate of $F_i(x)$;
    }
    obtain $L^{(k)}$ as the set of nondominated solutions in $L^{(k-1)} \cup S^{(k)}$ according to the objective function estimates just determined;
}

Fig. 1. Pseudocode of the procedure APS (Adaptive Pareto Sampling).

The determination of the efficient set $S^{(k)}$ in part (a) of APS can either be performed by a (multi-objective) exact algorithm, e.g., the Adaptive ε-Constraint Algorithm by Laumanns et al. (2006), or alternatively by a (multi-objective) metaheuristic.

As shown in (Gutjahr, 2009), APS converges under mild conditions with probability 1 to the efficient set. This extends even to the situation where the deterministic subproblem is only solved heuristically, provided that the used heuristic possesses a suitable convergence property itself. The most essential condition is that the sample size $\bar{s}_k$ for the solution evaluation part is increased

sufficiently fast; linear growth is sufficient. In particular cases, convergence can be obtained for a scheme where the sample size s_k for the solution proposal part is kept constant, but to ensure this, certain problem-specific conditions have to be verified. The problem-specific check can be dropped and replaced by a mild (random-search type) convergence property of the heuristic for the solution of the deterministic subproblem, if a scheme is used where also s_k is increased, see (Gutjahr and Reiter, 2010). In the following, we choose the scheme

$$s_k = k, \;\; \bar{s}_k = mk \quad (k = 1, 2, \ldots) \tag{4}$$

with an integer constant $m \geq 1$.

To give an example of application, let us consider a location problem where out of a set of n possible locations (nodes) for public facilities, a subset of nodes has to be selected. In each selected node j, a capacitated facility is to be built ("opened") at a random cost C_j with known distribution function Φ_j $(j = 1, \ldots, n)$. The facilities provide service for r population nodes. The demand in population node ρ is a random variable Q_ρ with a known distribution function Ψ_ρ $(\rho = 1, \ldots, r)$. Suppose that costs and demands in different nodes are independent. From independent random numbers $\omega_1, \ldots, \omega_{n+r}$ uniformly distributed on [0,1], realizations of C_j and Q_ρ can be computed by $C_j = \Phi_i^{-1}(\omega_j)$ and $Q_\rho = \Psi_\rho^{-1}(\omega_{n+\rho})$, respectively. We can consider $\omega = (\omega_1, \ldots, \omega_{n+r})$ as a particular scenario and write $C_j = C_j(\omega)$ and $Q_\rho = Q_\rho(\omega)$. The decision is given by the vector $x = (x_1, \ldots, x_n)$ where the variable x_j takes the value 1 if a facility in node j is opened and 0 otherwise. As the objective functions, let us consider the expectations $F_1(x)$ and $F_2(x)$ of the negative total cost $-f_1(x, \omega) = -\sum_{j=1}^{n} C_j(\omega)\, x_i$ and of a public benefit function $f_2(x, \omega) = G(x_1, \ldots, x_n, Q_1(\omega), \ldots, Q_r(\omega))$, respectively, where G builds on the distances of customers to opened facilities that can serve them under the given capacity restrictions. If a procedure computing the function G and a random number generator is available, APS can be applied to solve this problem, based on the BSAA estimates (2). Note that although in this example, the expectation $F_1(x)$ could also be computed analytically, i.e., without sampling, this may be very difficult or impossible for the expectation $F_2(x)$ since G is typically a complex nonlinear function.

3 Analyzed Problems and Scenario Sampling

3.1 Considered Problem Class

From now on, the investigation will be restricted to a somewhat more special case of the general class of problems of the form (1). As the search space X, the set $\{0, 1\}^n$ of binary strings of length n will be considered. (Of course, the assumption that further constraints are absent, which is also made in many other works on the analysis of evolutionary algorithms, is rather strong. It should be noted, however, that by the use of penalty functions, constrained problems can be included in the present framework as well.) By $X^* \subseteq X$, we denote the efficient set. Concerning the influence of randomness, we assume that the noise terms

$$\xi_i(x, \omega) = f_i(x, \omega) - \mathbb{E}(f_i(x, \omega)) = f_i(x, \omega) - F_i(x) \quad (i = 1, 2;\; x \in X) \tag{5}$$

are independent. We start with the assumption that the noise terms are normally distributed with fixed variance σ^2:

$$\xi_i(x, \omega) \sim \mathcal{N}(0, \sigma^2) \quad (i = 1, 2;\; x \in X). \tag{6}$$

We will then also proceed to situations where the noise terms follow other distributions. The assumptions above are less restrictive than they look like: Using binary problem encodings is

standard in applications of evolutionary algorithms, which justifies the special choice of X. The assumption of independence of the noise terms is w.l.o.g., since dependent noise terms would leave the optimization problem (1) unchanged.[4] If the variances of the noise terms are not equal, we may replace them by their maximum value σ^2, which gives a conservative estimate as long as upper runtime bounds are investigated. The assumption that the noise distribution is Gaussian, finally, is suggested by the observation that in the application of APS to a problem of the above kind, the noise only enters through sample averages over i.i.d. random variables, and if the number of iterations is large, the sample size k is large as well during later iterations, which, by the Central Limit Theorem, shows that we do not obtain a completely different situation if we replace the given noise distribution by a normal distribution.

3.2 Sampling Procedure

Mathematically, the sampling procedure in the *solution proposal* part of APS can be imagined as the random choice of a sample $\{\omega_1, \ldots, \omega_k\}$ of $s_k = k$ independent scenarios where each scenario ω_ν can also be represented by a $[2^n \times 2]$ matrix

$$\begin{bmatrix} \xi_1(x^{(1)}, \omega_\nu), & \xi_2(x^{(1)}, \omega_\nu) \\ & \cdots \\ & \cdots \\ \xi_1(x^{(2^n)}, \omega_\nu), & \xi_2(x^{(2^n)}, \omega_\nu) \end{bmatrix} \tag{7}$$

of noise terms assigned to the elements $x^{(1)}, \ldots, x^{(2^n)}$ of $X = \{0,1\}^n$. It is clear that in the computer implementation, not all line vectors of the $[2^n \times 2]$ matrix above have to be generated at once when the νth sample is created: only those line vectors are of interest that correspond to the single solutions x examined in the current iteration by the (possibly heuristic) solution algorithm for the deterministic subproblem. These line vectors are generated in the order as they are needed. E.g., if we would apply random search with ten random solutions to get an approximate solution to the BSAA problem, we would choose ten random elements x, corresponding to ten lines, and generate the noise terms only for these ten lines within each of the k matrices.[5] The $\bar{s}_k = mk$ scenarios $\bar{\omega}_\nu$ in the *solution evaluation* part of APS can be represented by analogous matrices.

The objective functions of the BSAA problem are then

$$\bar{f}_i(x) = F_i(x) + \xi_i(x) \quad (i = 1, 2) \tag{8}$$

with

$$\xi_i(x) = \frac{1}{k} \sum_{\nu=1}^{k} \xi_i(x, \omega_\nu) \quad (i = 1, 2). \tag{9}$$

[4] Of course, dependence could make a difference in the way the sampling is done in the *solution proposal* part of APS. However, in order to stay within the premises of the following analysis, we may always replace dependent sampling by independent sampling, drawing a specific, independent scenario for each x and each i, as it is done in the *solution evaluation* part. This may deteriorate the performance of the algorithm, but as we are interested in upper runtime bounds, this is not a problem.

[5] In particular, the described procedure implies that if the solution algorithm for the deterministic subproblem examines one and the same solution x more than once, always the same noise term realizations should be used. For this purpose, the algorithm has to store the sample average estimates for the already examined solutions in a look-up table. Such an implementation may be less efficient than one where, if a solution is examined a second time, the noise is evaluated by a new independent simulation. For large n, however, the case where an evolutionary search algorithm returns in a later iteration to a solution that has already been examined before is comparably rare, such that it can be expected that in terms of the quality evolution of the successively examined solutions, the second implementation variant will behave very similarly to the variant investigated in this paper.

For a given sample $\{\omega_1, \ldots, \omega_k\}$, the functions $\xi_i : X \to \mathbb{R}$ are well-defined. In the sequel, the sample error terms $\xi_i(x)$ will be comprised to the vector $\xi(x) = (\xi_1(x), \xi_2(x))$.

4 Incorporating SEMO into APS

4.1 Basic Version

As mentioned in Section 2, any exact or heuristic algorithm for multi-objective combinatorial optimization problems can be plugged in into APS for the solution of the BSAA subproblem. In the following, we shall discuss the use of the well-known SEMO (Simple Evolutionary Multiobjective Optimizer) algorithm (Laumanns et al., 2002b) for this purpose. The pseudocode of SEMO is presented in Fig. 2. Therein, $f(x) = (f_1(x), f_2(x))$ is the vector of objective function values of solution $x \in X$, and $\succ$ is the dominance relation in objective space defined in Section 2. SEMO uses an archive $\mathcal{P}$ which is initialized by a random solution and successively updated by the addition of non-dominated mutants and the removal of elements dominated by mutants. In the pseudocode, the termination criterion is left unspecified; later, we shall use a maximum number of iterations as a termination criterion. When SEMO is terminated, the current $\mathcal{P}$ is taken as the proposed approximation to the efficient set.

Procedure SEMO

Choose $x \in X$ uniformly at random;
$\mathcal{P} = \{x\}$;
loop
  choose $x \in \mathcal{P}$ uniformly at random;
  create x' by flipping a randomly chosen bit of x;
  $\mathcal{P} = \mathcal{P} \setminus \{y \in \mathcal{P} \mid f(x') \succ f(y)\}$;
  if $\nexists y \in \mathcal{P} : f(y) \succ f(x') \vee f(y) = f(x')$ **then** $\mathcal{P} = \mathcal{P} \cup \{x'\}$;
end loop

Fig. 2. Pseudocode of SEMO.

Although the algorithm APS/SEMO obtained by applying SEMO as a subprocedure of APS "works", its efficiency is usually rather poor. The reasons can be explained at hand of the following example:

Example 1. Consider the following stochastic extension of the bi-objective LOTZ ("Leading Ones, Trailing Zeros") problem introduced in (Laumanns et al., 2002b). For $x = (x_1, \ldots, x_n) \in X = \{0,1\}^n$, problem LOTZ is given by

$$\text{Maximize LOTZ}(x) = \left(\sum_{j=1}^{n} \prod_{r=1}^{j} x_r, \sum_{j=1}^{n} \prod_{r=j}^{n} (1 - x_r) \right). \tag{10}$$

The first and second objective function is the number of leading ones and the number of trailing zeros, respectively, in the binary string x. We extend LOTZ to the *stochastic* bi-objective problem SLOTZ by adding Gaussian noise: The objectives of SLOTZ are the expectations $F_1(x)$ and $F_2(x)$ of the random functions

$$(f_1(x,\omega), f_2(x,\omega)) = \text{LOTZ}(x) + (\xi_1(x,\omega),\ \xi_2(x,\omega)) \tag{11}$$

with independent noise variables $\xi_1(x,\omega)$ and $\xi_2(x,\omega)$ distributed according to (6). Obviously, $F(x) = \text{LOTZ}(x)$. Applying a black-box consideration, it is assumed that the solution algorithm does not "know" this but has to work based only on the observation of the noisy realization vectors $(f_1(x,\omega), f_2(x,\omega))$.

An application of APS/SEMO to SLOTZ is *inefficient* for the following two reasons: Also in the case where the sample error vectors $\xi(x)$ are very small,

(a) different solutions x and x' with the same image $F(x) = F(x')$ in objective space (e.g., 10110100 and 10100100) usually have different sample average evaluations, which lets the archive $\mathcal{P}$ grow too fast, and

(b) it may easily happen that although x is dominated by x' (but the points in objective space are identical in one component), the sample average evaluations of x and x' are incomparable, which misleads the search by SEMO.

In order to modify APS/SEMO to an algorithm overcoming these difficulties, we adopt the idea of *ϵ-dominance*, which has also turned out as useful in deterministic multi-objective optimization.

4.2 A Variant Based on ϵ-Dominance

The key idea is to replace—during the execution of the algorithm—the standard dominance relation $\succ$ by an "ϵ-tolerant" version $\succ_\epsilon$, but to do this in such a way that for small ϵ, the efficient set is nevertheless determined based on the original relation $\succ$. The concept of ϵ-dominance has frequently been applied in the multi-objective optimization literature; for examples, see (Laumanns et al., 2002a; Grosan, 2004; Schuetze et al., 2007; Horoba and Neumann, 2008). In this paper, the following definition will be used (we give reasons below):

Definition 1. Let $\epsilon > 0$. For two points z and z' in the objective space $\mathbb{R}^2$, we say that

(a) z' weakly ϵ-dominates z, written as $z' \succeq_\epsilon z$, if and only if $z'_i \geq z_i - \epsilon$ $(i = 1, 2)$.

(b) z' is ϵ-equal to z, written as $z' =_\epsilon z$, if and only if $|z'_i - z_i| \leq \epsilon$ $(i = 1, 2)$.

(c) z' ϵ-dominates z, written as $z' \succ_\epsilon z$, if and only if $z' \succeq_\epsilon z$ and not $z' =_\epsilon z$. In other words, z' ϵ-dominates z exactly if $z'_i \geq z_i - \epsilon$ $(i = 1, 2)$ and $\exists i : z'_i > z_i + \epsilon$.

For two points x and x' in the solution space, ϵ-dominance or ϵ-equality holds if it does so for their respective image points z and z' in the objective space. For fixed $z \in \mathbb{R}^2$, we define:

$$\begin{aligned}
\mathcal{D}(z) &= \{u \in \mathbb{R}^2 \,|\, u \succ z\} \quad \text{(set of points dominating point } z\text{)}, \\
\mathcal{D}_\epsilon(z) &= \{u \in \mathbb{R}^2 \,|\, u \succ_\epsilon z\} \quad \text{(set of points } \epsilon\text{-dominating point } z\text{)}, \\
\mathcal{E}_\epsilon(z) &= \{u \in \mathbb{R}^2 \,|\, u =_\epsilon z\} \quad \text{(set of points } \epsilon\text{-equal to } z\text{)}.
\end{aligned}$$

Furthermore, we set $\mathcal{H}_\epsilon(z) = \mathcal{E}_\epsilon(z) \cup (\mathcal{D}_\epsilon(z) \setminus \mathcal{D}(z))$. Fig. 3 illustrates the sets defined above.

Let us discuss the connections of Definition 1 to the definitions in the literature. The relation $\succeq_\epsilon$ is an additive (or "absolute") version of the multiplicative (or "relative") definition of ϵ-dominance given in (Laumanns et al., 2002a), where z' is said to ϵ-dominate z if and only if $(1+\epsilon) \cdot z'_i \geq z_i$ for all i. In our context, it is more natural to choose the additive version because also noise is assumed to be additive. Except of a marginal modification, such an additive version has also been used in (Schuetze et al. 2007). Note that the relation $\succeq_\epsilon$ relaxes the ordinary definition of dominance. This

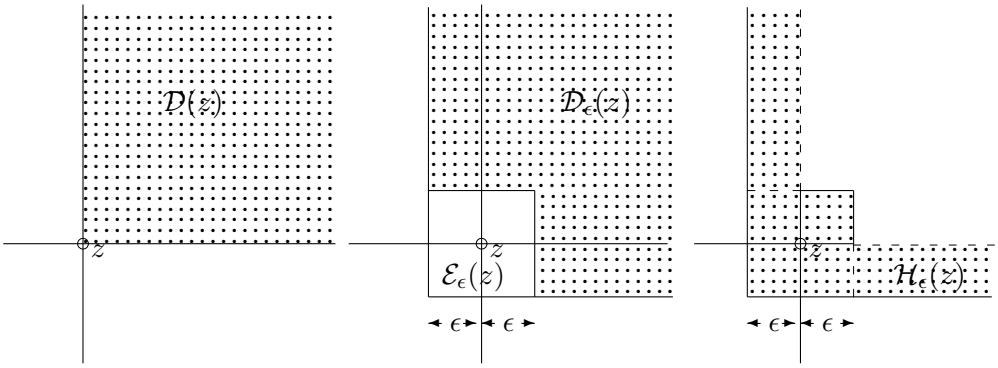

Fig. 3. Illustration of the sets $\mathcal{D}(z)$, $\mathcal{D}_\epsilon(z)$, $\mathcal{E}_\epsilon(z)$ and $\mathcal{H}_\epsilon(z)$ related to Definition 1.

is not true anymore for $\succ_\epsilon$, i.e., in the case where $z' =_\epsilon z$ is excluded. In this latter case, we obtain an "indented" dominance region which is similar (but not identical) to that of the ϵ-dominance relation used in (Grosan, 2004). At the end of this section, we shall see that the exemption of $z' =_\epsilon z$ is useful in our context, because it removes one possible source for a too rapid growth of the archive $\mathcal{P}$ in the case of noisy evaluations.

Definition 2. Let $\mathcal{M} \subseteq \mathbb{R}^2$ be a finite set of cardinality larger than one. By

$$\epsilon^*_{\mathcal{M}} = \min\{|\zeta_i - \zeta'_i| > 0 : \zeta \in \mathcal{M},\ \zeta' \in \mathcal{M},\ i = 1, 2\},$$

we denote the minimal nonzero distance between two points in $\mathcal{M}$ in one of the two coordinates. By the finiteness of $\mathcal{M}$, we have $\epsilon^*_{\mathcal{M}} > 0$.

In our proofs, we shall use the fact that for $\epsilon < \epsilon^*_{\mathcal{M}}$,

$$\mathcal{H}_\epsilon(z) \cap \mathcal{M} = \{z\} \quad \forall z \in \mathcal{M}. \tag{12}$$

This follows immediately from the observation that each point in $\mathcal{H}_\epsilon(z)$, except z itself, has in at least one coordinate a strictly positive distance to z that is smaller or equal to $\epsilon < \epsilon^*_{\mathcal{M}}$, and can therefore not belong to $\mathcal{M}$ by definition of $\epsilon^*_{\mathcal{M}}$. An immediate consequence of (12) is that for $\epsilon < \epsilon^*_{\mathcal{M}}$, also

$$\mathcal{E}_\epsilon(z) \cap \mathcal{M} = \{z\} \quad \forall z \in \mathcal{M}, \tag{13}$$

since $\mathcal{E}_\epsilon(z) \subset \mathcal{H}_\epsilon(z)$.

It is not difficult to verify that for $\epsilon > 0$ small enough, ϵ-dominance coincides with dominance, and ϵ-equality coincides with equality:[6]

Proposition 1. For $\epsilon < \epsilon^*_{\mathcal{M}}$ and $z, z' \in \mathcal{M}$,

$$z' \succ z \ \Leftrightarrow\ z' \succ_\epsilon z \quad \text{and} \quad z' = z \ \Leftrightarrow\ z' =_\epsilon z.$$

Proof. Easy. □

The modified overall algorithm, which we shall call APS/ϵ-SEMO, differs from APS/SEMO by the following simple changes: Each comparison of objective function values via "$\succ$" is replaced

[6]Observe that contrary to $\succ_\epsilon$, the relation $\succeq_\epsilon$ of Definition 1 does not coincide with ordinary dominance even for small ϵ because z does not dominate itself.

by a comparison via "$\succ_\epsilon$", and each comparison of objective function values via "$=$" is replaced by a comparison via "$=_\epsilon$". More precisely, the following two modifications have to be performed: First, in the *solution proposal* part of APS, SEMO is replaced by its ϵ-tolerant version, the algorithm ϵ-SEMO shown in Fig. 4. Secondly, also in the *solution evaluation* part of APS, the *trimming*, i.e., the statement "obtain $L^{(k)}$ as the set of nondominated solutions in $L^{(k-1)} \cup S^{(k)}$ according to the objective function estimates just determined", is now implemented in an ϵ-tolerant way by the procedure ϵ-TRIM described in Fig. 5. Therein, for abbreviation, $L^{(k-1)} \cup S^{(k)}$ is denoted by X^+, and $L^{(k)}$ is denoted by X^-. In both procedures, for $f(x)$, the sample average approximation $\bar{f}(x)$ of $F(x)$ has to be inserted when they are used within the APS framework.

Procedure ϵ-SEMO

Choose $x \in X$ uniformly at random;
$\mathcal{P} = \{x\}$;
loop
  choose $x \in \mathcal{P}$ uniformly at random;
  create x' by flipping a randomly chosen bit of x;
  $\mathcal{P} = \mathcal{P} \setminus \{y \in \mathcal{P} \mid f(x') \succ_\epsilon f(y)\}$;
  if $\nexists y \in \mathcal{P} : f(y) \succ_\epsilon f(x') \vee f(y) =_\epsilon f(x')$ **then** $\mathcal{P} = \mathcal{P} \cup \{x'\}$;
end loop

Fig. 4. Pseudocode of ϵ-SEMO.

Procedure ϵ-TRIM(X^+)

$X^- = \emptyset$;
for all $x \in X^+$
  $X^- = X^- \setminus \{y \in X^- \mid f(x) \succ_\epsilon f(y)\}$;
  if $\nexists y \in X^- : f(y) \succ_\epsilon f(x) \vee f(y) =_\epsilon f(x)$ **then** $X^- = X^- \cup \{x\}$;
end for all

Fig. 5. ϵ-tolerant trimming of solution set $X^+ = L^{(k-1)} \cup S^{(k)}$ in the *solution evaluation* part of APS.

Returning to Example 1, we observe that the disadvantages (a) and (b) of APS/SEMO mentioned there do not occur anymore with APS/ϵ-SEMO, as soon as the noise has become small enough: First, for small noise, sample average evaluations of two different solutions x and x' with $F(x) = F(x')$ are considered as ϵ-equal, with the consequence that only one of the two solutions is kept in the archive. Secondly, if x dominates x' but $F(x)$ and $F(x')$ are equal in one component, then for small noise (and small ϵ), the sample average evaluation of x still ϵ-dominates that of x' (cf. Fig. 3).

5 Runtime Analysis

In this section, we turn to the analysis of the expected runtime that is needed until APS/ϵ-SEMO proposes a solution set whose image (in objective space) is the entire Pareto front of problem (1), i.e., until $F(\mathcal{P})$ equals $F(X^*)$. This runtime is called the *optimization time*. The expected optimization time according to the definition above has been used in several investigations of MOCO algorithms, see, e.g., (Neumann and Wegener, 2006). Note that the stopping criterion does not require to find all Pareto-optimal solutions; it is sufficient to find, for each z on the Pareto front, at

least one solution x with $F(x) = z$. When this has been achieved, we shortly say that the Pareto front has been found. This does not mean that the current (disturbed) objective function value estimates $f(x)$ coincide with the points on the Pareto front, but only that the Pareto front is covered with respect to the undisturbed evaluations $F(x)$ of the proposed solutions.

In the theory of evolutionary algorithms, it is customary to use the number of fitness evaluations as a measure for the runtime required by an algorithm. Essentially, we shall follow this convention, but in our context where fitness approximations are based on random samples, it is necessary to extend the definition in order to obtain a meaningful measure, since an evaluation based on a sample of size, say, one million, will certainly require much more computation time than an evaluation based on a sample of size one. For this reason, the following runtime unit definition will be applied:

Runtime measure: An approximate objective function evaluation (2) based on sample size s is assumed to require s time units.

Essentially, Theorem 1 below (which is proven with the help of three lemmas) states that by passing from the deterministic to the noisy version of the problem, the optimization time is only increased by a certain factor; this factor is polynomially bounded in important special cases. Before precisely formulating the result and giving a rigorous proof, let us shortly outline the key proof ideas.

First, it is shown (Lemma 1) that for ϵ small enough, function evaluations disturbed by less than $\epsilon/2$ will induce an ϵ-dominance relation between image points that is equivalent to the original dominance relation between the image points for undisturbed function evaluations, and analogously for the ϵ-equality relation. Thus, if the function evaluation vectors obtained from the sample averages fall into sufficiently small squares around the exact objective function vectors, the ϵ-dominance relation on the noisy problem behaves precisely like the ordinary dominance relation on the deterministic problem.

Secondly, the following consequence of the increasing sample size used in APS/ϵ-SEMOit is shown (Lemma 2 and Corollaries 2–4): The probability that in a given iteration, all function vector estimates fall into squares of a given size around the exact objective function vectors, can be bounded from below by certain explicit expressions (tending to one with increasing iteration index). Note that increasing the sample size in a sample average estimate decreases the probability of large deviations from the true value.

Furthermore, it is also necessary to show (Lemma 3) that in the solution evaluation part of APS/ϵ-SEMO, already found efficient solutions are confirmed, provided that the disturbances by the sample average estimates are small enough.

In the proof of Theorem 1, finally, the obtained expressions are used to bound the probability of the "sample success event", i.e., the event that in a certain iteration, the noisy evaluations do not lead to a wrong judgement about mutual dominance relations between the currently considered solutions. The desired bound on the expected runtime is then obtained by adding (i) the time required to reach an iteration such that in all subsequent iterations, the sample success event has become likely enough, and (ii) the expected optimization time after this iteration.

In the following, $||x||_\infty = \max\{|x_1|, |x_2|\}$ denotes the sup-norm, and as usual, the image set $\{F(x) \in \mathbb{R}^2 \,|\, x \in X\}$ is denoted by $F(X)$.

Lemma 1. Let $z, z' \in F(X)$ and $u, u' \in \mathbb{R}^2$ with $||z - u||_\infty < \epsilon/2$ and $||z' - u'||_\infty < \epsilon/2$, where $\epsilon < \epsilon^*/2$ with $\epsilon^* = \epsilon^*_{F(X)}$ according to Definition 2. Then

$$u \succ_\epsilon u' \;\Leftrightarrow\; z \succ z' \quad \text{and} \quad u =_\epsilon u' \;\Leftrightarrow\; z = z'.$$

Proof. (a) We start by showing the second equivalence. If $z = z'$, then

$$|u_i - u'_i| \leq |u_i - z_i| + |z'_i - u'_i| < \epsilon/2 + \epsilon/2 = \epsilon \quad (i = 1, 2)$$

and hence $u =_\epsilon u'$. Conversely, let $z \neq z'$. Then, according to the second equivalence of Proposition 1, $z \neq_{\epsilon'} z'$ for $\epsilon' = 2\epsilon < \epsilon^*$. Hence there is an i with $|z_i - z'_i| > \epsilon' = 2\epsilon$. Since

$$2\epsilon < |z_i - z'_i| \leq |z_i - u_i| + |u_i - u'_i| + |u'_i - z'_i| < |u_i - u'_i| + \epsilon,$$

this implies $|u_i - u'_i| > \epsilon$, and therefore $u \neq_\epsilon u'$.

(b) Now we show the first equivalence. Let $u \succ_\epsilon u'$. Since $u \succ_\epsilon u'$ and $u =_\epsilon u'$ cannot hold simultaneously, $u \succ_\epsilon u'$ implies $u \neq_\epsilon u'$ and hence also $z \neq z'$ by the already shown second equivalence of the lemma. We show $z \succ z'$ by contradiction: Assume $z \succ z'$ does not hold. Because of $z \neq z'$, this means that either $z' \succ z$, or z and z' are incomparable. In both cases, there is a coordinate i such that $z'_i > z_i$. By definition of ϵ^* and because of $2\epsilon < \epsilon^*$, even $z'_i > z_i + 2\epsilon$ must hold in this case. It follows that

$$u_i \; < \; z_i + \frac{\epsilon}{2} \; < \; z'_i - \frac{3\epsilon}{2} \; < \; u'_i + \frac{\epsilon}{2} - \frac{3\epsilon}{2} \; = \; u'_i - \epsilon.$$

However, $u \succ_\epsilon u'$ entails $u_i > u'_i - \epsilon$, which contradicts the just derived inequality.

Conversely, let $z \succ z'$. If $z_1 > z'_1$ and $z_2 > z'_2$, it follows that $z_i > z'_i + 2\epsilon$ $(i = 1, 2)$ and thus $u_i > u'_i + \epsilon$ $(i = 1, 2)$ analogously as above, so $u \succ_\epsilon u'$. Now assume that there is an i such that $z_i = z'_i$; w.l.o.g., let $i = 1$. From $z \succ z'$ we conclude $z \neq z'$ and thus in particular $z_2 > z'_2 + 2\epsilon$. It follows that $u_2 > u'_2 + \epsilon$. Furthermore,

$$u_1 \; > \; z_1 - \frac{\epsilon}{2} \; = \; z'_1 - \frac{\epsilon}{2} \; > \; u'_1 - \frac{\epsilon}{2} \; - \; \frac{\epsilon}{2} \; = \; u'_1 - \epsilon.$$

In total, we have $u \succ_\epsilon u'$. □

Corollary 1. Suppose that for the noise terms in a fixed iteration k of the part *solution proposal* of APS, it holds that

$$||\xi(x)||_\infty < \frac{\epsilon}{2} \quad \forall x \in X \quad \text{with } \epsilon < \frac{1}{2}\,\epsilon^*_{F(X)}.$$

Then ϵ-SEMO, applied to the problem with objective function vector $\bar{f}$, produces the same sequence of mutated solutions x' and the same sequence of current solution sets $\mathcal{P}$ as SEMO, applied to the problem with objective function vector F, provided that the same (pseudo-)random numbers for the choice of the initial solutions and the bit mutations are used. In particular, the runtime of ϵ-SEMO is then just k times the runtime of SEMO.

Proof. As seen by comparing ϵ-SEMO with SEMO, it suffices to show that for two solutions $x^{(1)}$ and $x^{(2)}$,

$$\bar{f}(x^{(1)}) \succ_\epsilon \bar{f}(x^{(2)}) \;\Leftrightarrow\; F(x^{(1)}) \succ F(x^{(2)}) \text{ and } \bar{f}(x^{(1)}) =_\epsilon \bar{f}(x^{(2)}) \;\Leftrightarrow\; F(x^{(1)}) = F(x^{(2)}). \qquad (14)$$

According to (8), we have $\bar{f}(x) = F(x) + \xi(x)$, so by assumption, $||F(x) - \bar{f}(x)||_\infty < \epsilon/2$ with $\epsilon < (1/2)\,\epsilon^*_{F(X)}$ for all $x \in X$. Thus, with $z = F(x^{(1)}) \in F(X)$, $z' = F(x^{(2)}) \in F(X)$, $u = \bar{f}(x^{(1)}) \in \mathbb{R}^2$ and $u' = \bar{f}(x^{(2)}) \in \mathbb{R}^2$, the conditions of Lemma 1 are satisfied. The first and the second equivalence of the Lemma translate then immediately into the first and the second equivalence of (14), respectively. The statement about the runtime follows directly from the equivalent behavior of SEMO and ϵ-SEMO in the considered iteration and from our chosen runtime definition, since in iteration k, the sample size k is applied. □

For the sequel, it is convenient to introduce the notation

$$Q_r(z) = \{\zeta \in \mathbb{R}^2 \mid ||\zeta - z||_\infty < r\}$$

for the open square with half side length $r > 0$ around z. Moreover, let $\mathbf{0} = (0, 0)$.

Lemma 2 below is a technical auxiliary result, providing a bound for the probability that sample averages of Gaussian random variables do not deviate "too much" from their means. We shall apply it later (in Corollaries 2 to 4) to the sample average estimates used in APS.

Lemma 2. For fixed $r > 0$, i.i.d. random variables $\xi_i^{(\nu)} \sim \mathcal{N}(0, \sigma^2)$ $(i = 1, 2;\ \nu = 1, \ldots, k)$, and with

$$k^+ = \frac{2\sigma^2}{r^2} \ln \frac{4\sigma}{r\sqrt{2\pi}}, \tag{15}$$

the probability

$$\pi_k = Pr\left\{ \left(\frac{1}{k} \sum_{\nu=1}^{k} \xi_1^{(\nu)},\ \frac{1}{k} \sum_{\nu=1}^{k} \xi_2^{(\nu)} \right) \in Q_r(\mathbf{0}) \right\}$$

is bounded below by

$$\pi_k \geq 1 - \frac{4\sigma}{r\sqrt{2\pi k}} \exp\left(-\frac{r^2 k}{2\sigma^2} \right) \tag{16}$$

for each $k \geq k^+$.

Proof. With $Y_i = \frac{1}{k} \sum_{\nu=1}^{k} \xi_i^{(\nu)} \sim \mathcal{N}(0, \frac{\sigma^2}{k})$, we find

$$\pi_k = Pr\{|Y_i| < r\ (i = 1, 2)\} = \left(1 - 2\phi\left(-\frac{r\sqrt{k}}{\sigma} \right) \right)^2,$$

where $\phi(x)$ denotes the cumulative distribution function of the standard normal distribution. Since $\phi(x) \leq -\varphi(x)/x$ with $\varphi(x) = \phi'(x) = (2\pi)^{-1/2} \exp(-x^2/2)$ for $x < 0$, one obtains

$$1 - 2\phi\left(-\frac{r\sqrt{k}}{\sigma} \right) \geq 1 - \frac{2\sigma}{r\sqrt{2\pi k}} \exp\left(-\frac{r^2 k}{2\sigma^2} \right).$$

For $k \geq k^+$, the bound on the r.h.s. is positive, since then

$$\frac{2\sigma}{r\sqrt{2\pi k}} \exp\left(-\frac{r^2 k}{2\sigma^2} \right) \leq \frac{2\sigma}{r\sqrt{2\pi}} \exp\left(-\frac{r^2 k}{2\sigma^2} \right) \leq \frac{2\sigma}{r\sqrt{2\pi}} \exp\left(-\frac{r^2}{2\sigma^2} \frac{2\sigma^2}{r^2} \ln \frac{4\sigma}{r\sqrt{2\pi}} \right) = \frac{1}{2}. \tag{17}$$

Therefore,

$$\pi_k \geq \left(1 - \frac{2\sigma}{r\sqrt{2\pi k}} \exp\left(-\frac{r^2 k}{2\sigma^2} \right) \right)^2.$$

From the last inequality, the statement of the Lemma follows by $(1 - z)^2 \geq 1 - 2z$ $(z \in \mathbb{R})$. □

Let us now apply Lemma 2 to the solution proposal part of APS by choosing the variables $\xi_i^{(\nu)}$ as the noise terms $\xi_i(x, \omega_\nu)$, such that $\frac{1}{k} \sum_{\nu=1}^{k} \xi_i^{(\nu)}$ becomes equal to the error term $\xi_i(x)$ of the sample average estimate as given by (9):

Corollary 2. For $k \geq k^+$, the probability

$$\eta_k = Pr\left\{ (\xi_1(x),\ \xi_2(x)) \in Q_r(\mathbf{0})\ \forall x \in \{0, 1\}^n \right\}$$

that for the *solution proposal* part of iteration k of APS, the error terms of the sample average estimates are in $Q_r(\mathbf{0})$ for all $x \in X$, is bounded below by

$$\eta_k \;\geq\; 1 - \frac{2^{n+2}\sigma}{r\sqrt{2\pi k}}\,\exp\left(-\frac{r^2 k}{2\sigma^2}\right).$$

Proof. In the Appendix.

Corollary 3. Let η_k^* denote the probability that not only for the *solution proposal* part of iteration k of APS, but also for the *solution evaluation* part of this iteration, the error terms of the sample average estimates are in $Q_r(\mathbf{0})$ for all $x \in X$. Then, for $k \geq k^+$,

$$\eta_k^* \;\geq\; 1 - \frac{2^{n+3}\sigma}{r\sqrt{2\pi k}}\,\exp\left(-\frac{r^2 k}{2\sigma^2}\right).$$

Proof. In the Appendix.

Corollary 4. For η_k^* as in Corollary 3, $C \in \mathbb{R}$, and with

$$\bar{k}_0 = \bar{k}_0(n) = \left\lceil \frac{2\sigma^2}{r^2}\max\left(\ln\frac{4\sigma}{r\sqrt{2\pi}},\,(n+3)\ln 2 + C\right)\right\rceil,$$

it holds that

$$\eta_k^* \;\geq\; 1 - \frac{\sigma}{r\sqrt{2\pi}}\,\exp(-C) \quad \forall k \geq \bar{k}_0.$$

Proof. In the Appendix.

Lemma 3. If $F(X^*) \subseteq F(X^+)$ holds for the efficient set X^*, and if for the error terms of the sample average estimates in the *solution evaluation* part of iteration k of APS, the inequalities $||\xi(x)||_\infty < \epsilon/2 \quad \forall x \in X$ with $\epsilon < (1/2)\,\epsilon^*_{F(X)}$ hold, then the application of ϵ-TRIM(X^+) in this iteration produces an X^- with $F(X^-) = F(X^*)$.

Proof. We have to show that after the execution of ϵ-TRIM, (i) for every $z \in F(X^*)$, an element x with $F(x) = z$ is contained in X^-, and (ii) no element $y \notin X^*$ is contained in X^-. The proof uses Lemma 1 in an analogous way as it is used in the proof of Corollary 1.

To verify (i), let $z \in F(X^*)$. Because of $F(X^*) \subseteq F(X^+)$, there is an element $x \in X^+$ with $F(x) = z$, so $x \in X^*$ and there is an iteration of ϵ-TRIM in which x is the current element of X^+ to be checked for possible insertion. Because of Lemma 1, $\bar{f}(y) \succ_\epsilon \bar{f}(x)$ for some other solution y would entail $F(y) \succ F(x)$ in contradiction to the efficiency of x. Again by Lemma 1, $\bar{f}(y) =_\epsilon \bar{f}(x)$ entails $F(y) = F(x)$. Therefore, if there is an element y in the current set X^- with $\bar{f}(y) \succ_\epsilon \bar{f}(x) \vee \bar{f}(y) =_\epsilon \bar{f}(x)$, then $F(y) = F(x)$. As a consequence, either x is added to X^- in this iteration, or X^+ already contains a y with $F(y) = F(x) = z$. In both cases, $F(X^-)$ contains z after the current insertion trial. It remains to show that the element of X^- with image z (let it call it x for simplicity) will not be removed from X^- in any later iteration. A removal of x could only take place if another element x' with $\bar{f}(x') \succ_\epsilon \bar{f}(x)$ would be added. However, by Lemma 1, this would entail $F(x') \succ F(x)$ and thus contradict the efficiency of x.

To verify (ii), let $y \notin X^*$. Then there is a $z \in F(X^*)$ such that $z \succ F(y)$. Because of $F(X^*) \subseteq F(X^+)$, there is an iteration where an element x with $F(x) = z$ is the current element to be checked for possible insertion. If in the beginning of this iteration, y is contained in X^-, it will be removed from X^-, because $\bar{f}(x) \succ_\epsilon \bar{f}(y)$ by Lemma 1, and x will be inserted instead. In a later iteration, y cannot be added to X^- anymore, because x or an element x' with $F(x') = F(x)$ is already in X^-, which prevents the insertion of y. □

Now we are in the position to prove our main results, Theorems 1 and 2 below. For a complete specification of APS/ϵ-SEMO, we have still to define how many iterations of ϵ-SEMO are to be performed in each iteration of APS. In agreement with our general assumption that we can base our analysis on corresponding results for SEMO on the *deterministic* (i.e., noise-free) limiting case of the problem under consideration, we make the mentioned decision dependent on an upper bound h_n for the expected number of function evaluations required by SEMO to find the Pareto front of a noise-free problem instance of size n: In each iteration of APS, exactly $2h_n$ iterations of ϵ-SEMO are executed.

Theorem 1. Let the assumptions in Section 3 be satisfied, let h_n be an upper bound for the expected optimization time of SEMO on noise-free problems of instance size n, let ϵ_n^* denote the minimum of the values $\epsilon_{F(X)}^*$ (given by Definition 2) over all problem instances of size n, let σ_n denote the maximum of the noise standard deviations σ over all problem instances of size n, and let

$$k_0 = k_0(n) = \left\lceil \frac{8\sigma_n^2}{\epsilon^2} \left((n+3)\,\ln 2 + \ln \frac{4\sigma_n}{\epsilon\sqrt{2\pi}} \right) \right\rceil \tag{18}$$

where $\epsilon = \epsilon_n = \epsilon_n^*/4$. Then

$$(2h_n + 1)(m+1)\left(\frac{1}{3}\,k_0^3 + k_0^2 + 6k_0 + 21\right)$$

is an upper bound for the expected optimization time of APS/ϵ-SEMO in an implementation where each call of ϵ-SEMO performs $2h_n$ iterations.

Proof. We set $r = \epsilon/2$ in Corollaries 2–4, such that for the error terms $\xi(x)$ $(x \in X)$, the condition $||\xi(x)||_\infty < \epsilon/2$ $(x \in X)$ with $\epsilon < (1/2)\,\epsilon_{F(X)}^*$ required in Corollary 1 as well as in Lemma 3 is satisfied. Furthermore, in Corollary 4, we choose the constant C as $C = -\ln(\epsilon\sqrt{2\pi}/(4\sigma))$. By this special choice, Corollary 4 yields

$$\eta_k^* \;\geq\; 1 - \frac{\sigma}{(\epsilon/2)\sqrt{2\pi}} \cdot \frac{\epsilon\sqrt{2\pi}}{4\sigma} \;=\; \frac{1}{2} \quad \forall k \geq \bar{k}_0 \tag{19}$$

with

$$\bar{k}_0 \;=\; \left\lceil \frac{8\sigma^2}{\epsilon^2} \;\max\left(\ln \frac{8\sigma}{\epsilon\sqrt{2\pi}},\; (n+3)\ln 2 + \ln \frac{4\sigma}{\epsilon\sqrt{2\pi}} \right) \right\rceil .$$

Therein, the first argument of "max" is smaller than the second argument and can therefore be omitted. Moreover, σ can obviously be replaced by an upper bound σ_n without making (19) invalid. Therefore, $\bar{k}_0$ can be replaced by k_0 as defined in (18). Hence, eq. (19) says that in each iteration $k \geq k_0$ of APS, with a probability of at least $1/2$, samples are drawn for which the error terms both in the solution proposal part and in the solution evaluation part all lie within the square $Q_{\epsilon/2}(\mathbf{0})$. We call the event that this happens the *sample success event*. Let us denote the ensemble of samples used for the *solution proposal* in a fixed iteration k by "sample 1", and the ensemble of samples used for the *solution evaluation* in this iteration by "sample 2".

Let us now use Corollary 1. It ensures that in the sample success event, the procedure ϵ-SEMO, applied to the disturbed problem according to sample 1, behaves equivalently to the procedure SEMO, applied to the noise-free problem. In particular, if given an unlimited number of iterations, ϵ-SEMO finds then the Pareto front in an expected runtime with upper bound h_n. Denoting the runtime until $F(X^*)$ has been found by T, the well-known Markov inequality yields

$$Pr(T \geq 2h_n) \;\leq\; Pr(T \geq 2\mathbb{E}(T)) \;\leq\; 1/2,$$

such that within $2h_n$ iterations, the Pareto front will be found with probability of at least $1/2$. Observe that as soon as SEMO has found $F(X^*)$ for the first time in the sense that the current $\mathcal{P}$ satisfies $F(\mathcal{P}) = F(X^*)$, it preserves this current solution $\mathcal{P}$ during the remaining iterations until iteration $2h_n$, and again by Corollary 1, in the sample success event, the same holds for ϵ-SEMO.

By Lemma 3, in the sample success event, the *solution evaluation* part of APS, applied to the disturbed problem according to sample 2, confirms the $\mathcal{P}$ with image $F(X^*)$ if the last has been found by ϵ-SEMO in the solution proposal part.

In total, we obtain that in each iteration $k \geq k_0$, with a probability larger or equal to $(1/2) \cdot (1/2) = 1/4$, the solution set $L^{(k)}$ proposed by APS at the end of this iteration has image $F(X^*)$. Note that the random numbers used for obtaining the two samples and those used for the mutations during the run of ϵ-SEMO are independent, and that in the sample success event, the error terms do not matter anymore. Moreover, for future use, it should be noted that the lower overall success probability bound $1/4$ for an iteration $k \geq k_0$ holds independently from the events in previous iterations.

Now we compute the runtime required by iteration k of APS/ϵ-SEMO. In this iteration, ϵ-SEMO performs $2h_n$ iterations, i.e., at most $2h_n$ new solutions x' are evaluated in addition to the initial solution. Because of $s_k = k$, this yields a runtime smaller or equal to $(2h_n + 1)\, k$ for the *solution proposal* part. The set $L^{(k-1)} \cup S^{(k)}$ of solutions to be evaluated in the *solution evaluation* part contains at most $k \cdot (2h_n + 1)$ elements, because in each APS iteration, at most $2h_n + 1$ new solutions can be proposed. Considering $\bar{s}_k = mk$, this gives an upper runtime bound of $(2h_n + 1)mk^2$ for the *solution evaluation* part in iteration k. Therefore, the cost of the entire iteration is bounded from above by $(2h_n + 1)(k + mk^2) \leq Rk^2$ with $R = (2h_n + 1)(m + 1)$.

The total runtime in iterations 1 to $k_0 - 1$ of APS is therefore bounded from above by

$$R \sum_{k=1}^{k_0-1} k^2 \leq R \int_0^{k_0} k^2 dk = R\, \frac{k_0^3}{3}. \tag{20}$$

Finally, we turn to the expected total runtime in iterations k_0, $k_0 + 1$ etc. until X^* has been identified. We use the lower bound $1/4$ on the total success probability in an iteration $k \geq k_0$ and assume pessimistically that the total success probability is equal to $1/4$ instead of larger or equal. This gives the following upper bound for the expected optimization time after the start of iteration k_0:

$$\begin{aligned} &R\, k_0^2 \cdot \frac{1}{4} + R\, (k_0 + 1)^2 \cdot \frac{3}{4} \cdot \frac{1}{4} + R\, (k_0 + 2)^2 \cdot \left(\frac{3}{4}\right)^2 \cdot \frac{1}{4} + \ldots \\ &= \frac{R}{4} \sum_{k=k_0}^{\infty} k^2\, \xi^{k-k_0} = \frac{R}{4} \sum_{k=0}^{\infty} (k^2 + 2k_0 k + k_0^2)\, \xi^k. \end{aligned}$$

with $\xi = 3/4$. By using the well-known summation formulas for the series $\sum_k \xi^k$, $\sum_k k\xi^k$ and $\sum_k k^2 \xi^k$, we can evaluate the last expression as $R\, (k_0^2 + 6k_0 + 21)$. Addition to (20) and insertion of the expression for R yields the result. □

Remark 1. From an application point of view, Theorem 1 raises the question how ϵ and the runtime given to ϵ-SEMO in each call of this procedure should be chosen in an implementation of APS/ϵ-SEMO if ϵ_n^* and/or the bound h_n are not known. A pragmatic answer is to *tune* both parameters experimentally for the considered class of problem instances, such as it is usually done for other parameters of a metaheuristic. Obviously, an upper bound for the expected optimization time under the choice of these two parameters as described in Theorem 1 is also an upper bound for the expected optimization time under the choice of the (experimentally found) *best* combination of

these parameters. A more sophisticate way of handling the issue would be to gradually decrease ϵ_n during the run of the APS framework algorithm, and to gradually increase the runtime given to ϵ-SEMO. For all iterations where ϵ_n has fallen below $\epsilon_n^*/4$ and the runtime of ϵ-SEMO has exceeded $2h_n$, analogous estimations as in the proof of Theorem 1 become applicable. Possibly, this will allow an extension of Theorem 1 to the described self-adapting variant.

Remark 2. Our presentation of the SEMO algorithm has been based on the original version of SEMO as introduced in (Laumanns et al. 2002b). In the literature, also a slightly modified version of SEMO has been analyzed, see, e.g., (Neumann and Wegener 2006). This version, which we shall denote by SEMO', differs from the original one by the fact that if to an element x' considered for insertion, there exists already an element $y \in \mathcal{P}$ with the same objective function values, then instead of keeping y and disregarding x', the new x' is inserted into $\mathcal{P}$, and y is omitted. More formally, the two last statements in the loop of SEMO (Fig. 2) are replaced by

if $(\nexists y \in \mathcal{P} : f(y) \succ f(x'))$ **then** $\mathcal{P} = (\mathcal{P} \setminus \{y \in \mathcal{P} \mid f(x') \succ f(y) \vee f(x') = f(y)\}) \cup \{x'\}$;

By obvious analogous replacements, we obtain corresponding procedures ϵ-SEMO' and ϵ-TRIM' from ϵ-SEMO and ϵ-TRIM, respectively. Going through our proofs, it is easily seen that Theorem 1 remains valid if in APS/ϵ-SEMO, subprocedure ϵ-SEMO is replaced by ϵ-SEMO', and subprocedure ϵ-TRIM is replaced by ϵ-TRIM'.

We immediately obtain from Theorem 1:

Corollary 5. If $\epsilon^*(n) = \Omega(1)$ and $\sigma_n = O(1)$, the expected optimization time of APS/ϵ-SEMO grows at most by a factor of order $O(n^3)$ faster in n than that of SEMO applied to the noise-free problem.

Example 1 (continued). For the SLOTZ problem introduced in subsection 4.1, we have $\epsilon_n^* = 1 \ \forall n$. A well-known result by Laumanns et al. (2002b) states that the expected optimization time required by SEMO to find the efficient set for LOTZ is of order $\Theta(n^3)$, with an upper bound $h_n = (1/2)n^3 - (1/2)n^2$. By applying our Theorem 1, we derive from this result that the expected optimization time for finding the efficient set for SLOTZ is at most of order $O(n^6)$.

Example 2. The article (Neumann and Wegener 2006) analyzes the expected optimization time of SEMO' (cf. Remark 2) for a bi-objective Minimum Spanning Tree (MST) problem. In an edge-weighted graph with r nodes and n edges, to each subset x (representable as a binary string) of the set of edges, two objective functions are assigned: The first objective $c(x)$ is the number of connected components of the subgraph defined by x, the second objective $w(x)$ is the total weight of the edges in x. Both objectives are to be minimized. Weights are assumed to be integers with a maximum value of w_{max}. Neumann and Wegener show that the expected optimization time of SEMO' is of order $O(nr(\log r + \log w_{max}) + nr^2)$.

In a first noise model, we extend the problem to the stochastic case in an analogous way as it has been done for SLOTZ by adding random Gaussian noise to each objective:

$$(f_1(x,\omega), f_2(x,\omega)) = (c(x) + \xi_1(x,\omega),\ w(x) + \xi_2(x,\omega)),$$

with independent noise variables $\xi_1(x,\omega)$ and $\xi_2(x,\omega)$ distributed according to (6). For the standard deviations, we assume $\sigma_n = O(1)$. Similarly as in the SLOTZ example, it is supposed that the solution algorithm is "blind" with respect to the underlying MST structure of the problem. Because of the integrality of the objectives, we have $\epsilon_n^* \geq 1 \ \forall n$. Let us assume that $w_{max} \leq 2^r$, and that the relation between the number of nodes and the number of edges satisfies $r = O(n^\alpha)$ with some $\alpha \geq 1/2$. (The case $\alpha = 1/2$ represents dense graphs.) Then $h_n = O(n^{2\alpha+1})$, and Corollary 5

yields an expected optimization time of order $O(n^{2\alpha+4})$ on the stochastic problem.

As an alternative, let us also consider a second noise model where the noise $\xi_2(x,\omega)$ results from imprecise measurement of the weights of the edges on the assumption that an estimate of $w(x)$ is obtained by summing up the individual weight estimates of the edges contained in x. Suppose that the individual noise terms for the measurement of the edges are independent and normally distributed with a variance smaller or equal to some constant independent of n, and that the number $c(x)$ of components can be determined without noise. Then $\sigma_n = O(n^{1/2})$, and again on the assumptions $w_{max} \leq 2^r$ and $r = O(n^\alpha)$, Theorem 1 yields an expected optimization time of order $O(n^{2\alpha+7})$.

Corollary 5 shows that in many cases, the transition from a MOCO problem to a SMOCO problem only generates a polynomial-time overhead factor. Although this is a helpful insight, the bounds presented in Corollary 5 and the SLOTZ example are still rather weak. We conjecture that stronger bounds hold, and below we shall outline a situation where stronger bounds can be rigorously derived indeed. Before turning to this issue, a generalization of Theorem 1 will be demonstrated. We start with a definition.

Definition 3 (Birnbaum, 1984). For two random variables ξ and ξ', the variable ξ is called *more peaked about zero* than the variable ξ' if

$$Pr\{|\xi| \geq z\} \;\leq\; Pr\{|\xi'| \geq z\} \quad \forall z > 0. \tag{21}$$

Corollary 6. The assertion of Theorem 1 remains still valid if the $\mathcal{N}(0,\sigma^2)$-distributed noise $\xi_i(x,\omega)$ is replaced by independent noise $\bar{\xi}_i(x,\omega)$ following some other distribution (which is allowed to depend on i and x), provided that for all $x \in X$ and $i = 1, 2$,

(i) the density of $\bar{\xi}_i(x,\omega)$ is continuous, symmetric (i.e., the density in $-z$ is equal to the density in z) and non-increasing for positive arguments z, and

(ii) $\bar{\xi}_i(x,\omega)$ is more peaked about zero than a random variable $Z_n \sim \mathcal{N}(0,\sigma_n^2)$.

Proof. Theorem 1 in (Birnbaum, 1984) states that under the indicated conditions, the corresponding sample averages inherit the "more peaked" relation, i.e., $\bar{\xi}_i(x) = \frac{1}{k}\sum_{\nu=1}^{k} \bar{\xi}(x,\omega_\nu)$ is more peaked about zero than $\xi_i(x) = \frac{1}{k}\sum_{\nu=1}^{k} \xi_i(x,\omega_\nu)$ with $\xi_i(x) \sim \mathcal{N}(0,\sigma_n^2)$. A re-inspection of the proof of our Theorem 1 above and the preceding lemmas (especially Lemma 2 and Corollaries 2–4) shows that this property is all that is needed to make the used upper bound estimations still valid for the case where the $\xi_i(x,\omega)$ are replaced by the $\bar{\xi}_i(x,\omega)$. □

Corollary 6 above allows it to include also the case of *bounded* noise, e.g., noise following a triangular distribution or a truncated normal distribution[7]. Our next result shows that in the case of bounded noise, a considerably stronger runtime bound than that of Theorem 1 can be derived:

Theorem 2. Let the conditions of Corollary 6 be satisfied, and let for all n,

(i) $F(X) \subseteq [a_n, b_n] \times [a_n, b_n] \quad (a_n > 0,\ b_n > 0)$,

(ii) $|\xi_i(x,\omega)| \leq c_n$ with probability 1 $\quad (i = 1, 2,\ x \in X) \quad$ with $c_n > 0$.

[7]Strictly speaking, in the last case, the truncation must be defined in such a way that the density function remains continuous, which can of course approximate a non-continuously truncated normal distribution to any desired degree of accuracy.

Then the expected optimization time of APS/ϵ-SEMO on an instance of size n in an implementation where each call of ϵ-SEMO performs $2h_n$ iterations is bounded from above by

$$\left[2h_n + \left(\frac{8(b_n - a_n + 2c_n)}{\epsilon_n^*} + 2\right) m + 1\right] \left(\frac{1}{2}\, k_0^2 + \frac{1}{2}\, k_0 + 3\right)$$

with k_o as in Theorem 1.

Proof. First of all, we compute the maximal length of an ϵ-antichain in a square with side length γ ($\gamma > 0$). An ϵ-antichain is a set $\mathcal{A} \subseteq \mathbb{R}^2$ of points with the property that two different points $z, z' \in \mathcal{A}$ are ϵ-incomparable, i.e., that neither $z \succ_\epsilon z'$ nor $z' \succ_\epsilon z$ nor $z =_\epsilon z'$. In particular, this means that two elements z and z' in an ϵ-antichain differ by at least ϵ in *each* of their two coordinates. By projection to one of the two coordinate axes, one immediately sees that in the considered square, not more than $\gamma/\epsilon + 1$ mutually ϵ-incomparable elements can exist. We choose $\epsilon = \epsilon_n^*/4$ as in Theorem 1. Then we have that an ϵ-antichain can contain not more than $4\gamma/\epsilon_n^* + 1$ elements.

In iteration k, both the solution $S^{(k)}$ delivered by ϵ-SEMO and the solution $L^{(k-1)}$ delivered in the previous iteration by ϵ-TRIM are ϵ-antichains. Since by assumption, the modulus of the noise is smaller or equal to c_n, also

$$\left|\frac{1}{s}\sum_{\nu=1}^{s} \xi_i(x, \omega_\nu)\right| \leq \frac{1}{s}\sum_{\nu=1}^{s} |\xi_i(x, \omega_\nu)| \leq \frac{1}{s} \cdot s c_n = c_n$$

holds, and therefore both the sample average estimates used for the elements of $S^{(k)}$ in the call of ϵ-SEMO in iteration k and those that have been used for the elements of $L^{(k-1)}$ in the *solution evaluation* part of iteration $k-1$ lie all in the square $[a_n - c_n, b_n + c_n] \times [a_n - c_n, b_n + c_n]$, i.e., in a square with side length $\gamma_n = b_n - a_n + 2c_n$. Using the bound above, we conclude that $|S^{(k)}| \leq 4\gamma_n/\epsilon_n^* + 1$ and $|L^{(k-1)}| \leq 4\gamma_n/\epsilon_n^* + 1$, so $|L^{(k-1)} \cup S^{(k)}| \leq 8\gamma_n/\epsilon_n^* + 2$.

Now we go through the proof of Theorem 1, resp. that of Corollary 6 which allows a generalization to noise that is more peaked than Gaussian noise. In these proofs, we used an upper bound of $k \cdot (2h_n + 1)$ for the number of elements in $|L^{(k-1)} \cup S^{(k)}|$ that have to be re-evaluated in the *solution evaluation* part of the APS iteration. On the current premises, this set can contain at most $8\gamma_n/\epsilon_n^* + 2$ elements. For iteration k of APS, this yields an upper runtime bound of $(2h_n + 1)k + mk \cdot (8\gamma_n/\epsilon_n^* + 2) = \bar{C}k$ with

$$\bar{C} = 2h_n + 1 + m\left(\frac{8\gamma_n}{\epsilon_n^*} + 2\right) = 2h_n + \left(\frac{8(b_n - a_n + 2\epsilon_n)}{\epsilon_n^*} + 2\right) m + 1.$$

Therefore, the total runtime in iterations 1 to $k_0 - 1$ of APS is bounded from above by

$$\bar{C}\sum_{k=1}^{k_0-1} k = \bar{C}\left(\frac{1}{2}k_0^2 - \frac{1}{2}k_0\right),$$

and the expected total runtime in iterations k_0, $k_0 + 1$ etc. until $F(X^*)$ has been identified is bounded from above by

$$\bar{C}k_0 \cdot \frac{1}{4} + \bar{C}(k_0 + 1) \cdot \frac{3}{4} \cdot \frac{1}{4} + \bar{C}(k_0 + 2) \cdot \left(\frac{3}{4}\right)^2 \cdot \frac{1}{4} + \ldots$$

$$= \frac{\bar{C}}{4}\sum_{k=k_0}^{\infty} k\, \xi^{k-k_0} = \frac{\bar{C}}{4}\sum_{k=0}^{\infty}(k + k_0)\, \xi^k = \frac{\bar{C}}{4} \cdot (12 + 4k_0) = \bar{C}(k_0 + 3)$$

with $\xi = 3/4$. Hence $\bar{C} \cdot \left(\frac{1}{2}\, k_0^2 + \frac{1}{2}\, k_0 + 3\right)$ is an upper bound for the total expected optimization time. Insertion for $\bar{C}$ gives the result. □

Example 1 (continued). Replace in the SLOTZ problem introduced in subsection 4.1 the Gaussian noise $\xi_i(x,\omega) \sim \mathcal{N}(0,\sigma^2)$ by noise $\bar{\xi}_i(x,\omega)$, where the distribution of $\bar{\xi}_i(x,\omega)$ is more peaked about zero than $\mathcal{N}(0,\sigma^2)$, and where $\bar{\xi}_i(x,\omega) \leq c$ for all $x \in X$ with some constant $c > 0$.[8] Note that for $a_n = 0$, $b_n = n$ and $c_n = c$ the conditions of Theorem 2 are satisfied, and that $b_n - a_n + 2c_n = n + 2c = \Theta(n)$. Then Theorem 2 yields an upper bound for the expected optimization time of

$$[O(n^3) + (O(n) + 2)m + 1] \cdot O(n^2) = O(n^5), \tag{22}$$

which is only by a factor of order $O(n^2)$ worse than the expected optimization time in the deterministic boundary case of LOTZ. With a view at the proof of Theorem 2, it can also be seen from (22) that now, the term of order $O(n) \cdot m$ referring to the *solution evaluation* part of APS is negligible compared to the term of order $O(n^3)$ referring to the *solution proposal* part. This is a typical situation also for applications to more complex problems, since the current approximation $L^{(k-1)} \cup S^{(k)}$ of the efficient set is usually rather small, such that the *solution evaluation* part is computationally unproblematic.

Remark 3. Obviously, the $O(n^2)$ bound for the ratio of expected optimization times between stochastic and deterministic problem in the bounded-noise case can be generalized from SLOTZ to a broader class of problems, analogously as in Corollary 5. However, it is not quite general, as the following example shows.

Example 2 (continued). For the stochastic bi-objective MST problem of Example 2, let us perform a similar replacement of the Gaussian noise $\xi_i(x,\omega) \sim \mathcal{N}(0,\sigma^2)$ by noise $\bar{\xi}_i(x,\omega)$ with a distribution of $\bar{\xi}_i(x,\omega)$ more peaked about zero than $\mathcal{N}(0,\sigma^2)$ as in Example 1 above. Let all other assumptions be the same as before. For $w_{max} \leq 2^r$, Theorem 2 yields in this case an exponential bound. However, if $w_{max} = O(n^\beta)$ with some $\beta > 0$, Theorem 2 (with $a_n = 0$, $b_n = w_{max}$ and $c_n = c$) gives a bound for the expected optimization time of order $O(n^{2\alpha+3} + mn^{\beta+2})$.

Remark 4. Also for a normal distribution, in the case of sufficiently large $c_n > 0$, absolute noise values larger than c_n are very unlikely. Therefore, it can be conjectured that for Gaussian noise, the expected optimization time is bounded by an order that even conforms to the often stronger bound of Theorem 2 instead of only to the bound of Theorem 1. We leave a rigorous proof of this conjecture as an open question.

Remark 5. We do not claim that the sample size increment scheme $s_k = \Theta(k)$ and $\bar{s}_k = \Theta(k)$ is optimal for the expected optimization time of APS/ϵ-SEMO. Concerning the *upper bound* for the expected optimization time, it can even be shown that it becomes the better the faster the sample size is increased: Consider a scheme where $s_k = \bar{s}_k = k^p$ with some $p \geq 0$. For the sake of simplicity, let us restrict the discussion to the SLOTZ special case. Then by analogous derivations as those in Theorems 1 and 2, we find an expected optimization time bound of order $O(n^{4+2/p})$ and of $O(n^{4+1/p})$ in the case of Gaussian and of bounded noise, respectively, which decreases for increasing p. Nevertheless, it should be kept in mind that these are only upper bounds. It is easily possible that the actual expected optimization time of the linear scheme is better than $\Theta(n^4)$, since in our proofs, we used rather coarse bound estimations, assuming that the efficient set has to be identified as a whole in one of the APS iterations, whereas APS can also find the elements of the efficient set one after another or subset by subset, which considerably speeds up the process. This

[8] The special case of large c and a continuously truncated normal distribution produces a very close approximation to the Gaussian model.

gradual identification of the efficient set, however, is prevented if the sample size is increased to fast.[9]

6 Conclusions

In this paper, an algorithm for stochastic multi-objective combinatorial optimization (SMOCO) has been analyzed. The algorithm is obtained by inserting the Simple Evolutionary Multiobjective Optimizer (SEMO) as a subprocedure into Adaptive Pareto Sampling (APS). Normally distributed noise as well as more general noise distributions have been investigated. It has been shown that if SEMO applies a specific ϵ-dominance relation instead of the ordinary dominance relation, the expected optimization time of the overall algorithm can be bounded from above, provided that ϵ is chosen small enough. The bound depends on the expected optimization time of SEMO on the corresponding *deterministic* counterpart problem. An explicit expression for this dependence has been given. In the special case where the variance of the noise is bounded from above by a constant independent of the problem size n, and ϵ is bounded from below by a constant independent of n, the ratio between the expected optimization times for the stochastic and the corresponding deterministic problem, respectively, is of order $O(n^3)$. For bounded noise, the ratio reduces to $O(n^2)$.

Since the underlying deterministic single- or multi-objective optimization problems of practically relevant SMOCO problems are often NP-hard, such that the optimization time for them must be anticipated to grow exponentially fast in the instance size, the results show that in typical cases, *stochastic* MOCO problems are not essentially harder than their deterministic counterparts.

Future research should address several topics. A question of practical relevance is whether the results presented here can also be transferred from the use of SEMO to the use of a more elaborated multi-objective metaheuristic such as NSGA-II or SPEA2. This question is especially important in view of well-known drawbacks of SEMO, e.g., the fact that SEMO can end up with prohibitively large populations. Furthermore, a generalization of the results presented here to the case of more than two objective functions would be of interest.

Expected optimization time is not the only relevant criterion for the performance of an evolutionary optimization algorithm; often, the immediate goal is not to find the exact solution (in the multi-objective case: the efficient set), but rather to identify a good approximate solution within short time. For investigating this question in the multi-objective situation, the behavior of the hypervolume or of another quality indicator during the run of a combined algorithm of the type considered here could be studied.

The bounds derived here are rather coarse, and it is possible that tighter bounds may be found if the way how APS gradually adapts the currently proposed solution set is taken into more detailed consideration. Moreover, in the present work, we focused on *upper* bounds for the expected optimization time. A most interesting question would be how good lower bounds look like and whether there are cases where it can be shown that their order matches that of the found upper bounds. Based on such results, also the question how the sample size scheme in APS can be organized in order to obtain the fastest possible convergence to the Pareto front could be addressed.

Acknowledgment. The author thanks Lothar Thiele for a discussion on ϵ-dominance concepts during the Dagstuhl seminar 2010 on "Theory of Evolutionary Algorithms", and three anonymous

[9] In the case where instead of SEMO, a solver based on Mathematical Programming is used for the solution of the deterministic subproblem, as it has been done in (Gutjahr, 2009), a rapid increase of the sample size is especially detrimental, since the runtime of such a solver typically increases exponentially fast in the sample size, such that high sample sizes have to be avoided at any price.

reviewers for their very helpful comments on the first version of this paper.

References.

Amodeo, L., Prins, C., Sanchez, D. (2009). Comparison of metaheuristic approaches for multi-objective simulation-based optimization in supply chain inventory management. Applications of Evolutionary Computing, Lecture Notes in Computer Science 5484, Springer, 798–807.

Baesler, F.F., Sepúlveda, J.A. (2001). Multi-objective simulation optimization for a cancer treatment center. In: Proc. Winter Simulation Conference 2001, 1405-1411.

Basseur, M., Zitzler, E. (2006). Handling uncertainty in indicator-based multiobjective optimization. International Journal of Computational Intelligence Research 2, 255–272.

Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing 8, 239-287.

Birnbaum, Z.W. (1984). On random variables with comparable peakedness. Annals of Mathematical Statistics 19, 76–81.

Birge, J.R., Louveaux, F. (1997). Intoduction to Stochastic Programming, Springer.

Brockhoff, D. (2011). Theoretical aspects of evolutionary multiobjective optimization—a review. Theory of Randomized Search Heuristics (eds.: A. Auger, B. Doerr), World Scientific Publishing, 101–139.

Caballero, R., Cerda, E., Muños, M.M., Rey, L. (2004). Stochastic approach versus multiobjective approach fro obtaining efficient solutions in stochastoc multiobjective programming problems. European Journal of Operations Research 158, 633–648.

Cardona-Valdés, Y., Álvarez, A., Ozdemir, D. (2011). A bi-objective supply chain design problem with uncertainty. Transportation Research Part C 19, 821–832.

Claro, J., Sousa J.P. (2010a). A multiobjective metaheuristic for mean-risk multistage capacity investment problem. Journal of Heuristics 16, 85–115.

Claro, J., Sousa, J.P. (2010b). A multiobjective metaheuristic for a mean-risk static stochastic knapsack problem. Computational Optimization and Applications 46, 427–450.

Coello Coello, C.A. (2006). Evolutionary multi-objective optimization: a historical view of the field. Computational Intelligence Magazine 1, 28–36.

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

Franca, R.B., Jones, E.C., Richards, C.N., Carlson, J. (2010). Multi-objective stochastic supply chain modeling to evaluate tradeoffs between profit and quality. International Journal of Production Economics 127, 292–299.

Grosan, C. (2004). Improving the performance of evolutionary algorithms for multi-objective 0/1 knapsack problems using ϵ-dominance. Congress on Evolutionary Computation (CEC), G. Greenwood et al. (eds.), IEEE Press, 1958–1963.

Guillén, G., Mele, F.D., Bagajewicz, M.J., Espuña, A., Puigjaner, L. (2005). Multiobjective supply chain design under uncertainty. Chemical Engineering Science 60, 1535–1553.

Gutjahr, W.J. (2005). Two metaheuristics for multiobjective stochastic combinatorial optimization. Proc. SAGA 2005 (Stochastic Algorithms: Foundations and Applications), LNCS 3777, Springer, 116–125.

Gutjahr, W.J. (2009). A provably convergent heuristic for stochastic bicriteria integer programming. Journal of Heuristics 15, 227–258.

Gutjahr, W.J., Reiter, P. (2010). Bi-objective project portfolio selection and staff assignment under uncer-

tainty. Optimization 59, 417–445.

Hassan-Pour, H.A., Mosadegh-Khah, M., Tavakkoli-Moghaddam, R. (2009). Solving a multi-objective multi-depot stochastic location-routing problem by a hybrid simulated annealing algorithm. In: Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 1045–1054.

Hnaien, F., Delorme, X., Dolgui, A. (2010). Multi-objective optimization for inventory control in two-level assembly systems under uncertainty of lead times. Computers and Operations Research 37, 1835–1843.

Homem-de-Mello, T. (2003). Variable-sample methods for stochastic optimization. ACM Transactions on Modeling and Computer Simulation 13, 108–133.

Horoba, C., Neumann, F. (2008). Benefits and drawbacks for the use of epsilon-dominance in evolutionary multi-objective optimization. Proc. GECCO '08 (Genetic And Evolutionary Computation Conference), ACM, 641–648.

Hughes, E.J. (2001). Evolutionary multi-objective ranking with uncertainty and noise: Proc. EMO '01 (Evolutionary Multicriterion Optimization), Springer, 329–343.

Jin, Y., Branke, J. (2005). Evolutionary optimization in uncertain environments – a survey. IEEE Transactions on Evolutionary Computation 9, 303–317.

Laumanns, M., Thiele, L., Deb, K., Zitzler, E. (2002a). Combining convergence and diversity in evolutionary multi-objective optimization. Evolutionary Computation 10, 263–282.

Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., Deb, K. (2002b). Running time analysis of multi-objective evolutionary algorithms on pseudo-boolean functions. In: Conference on Parallel Problem Solving from Nature (PPSN VII), LNCS 2439, Springer, 44–53.

Laumanns, M., Thiele, L., Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. European Journal of Operational Research 169, 932–942.

Liefooghe, A., Basseur, M., Jourdan, L., Talbi E.-G. (2007). Combinatorial Optimization of Stochastic Multi-objective Problems: An application to the flow-shop scheduling problem. Proc. EMO 2007 (Conference on Evolutionary Multi-Criterion Optimization), LNCS 4403, Springer, 457–471.

Neumann, F., Wegener, I. (2006). Minimum spanning trees made easier via multi-objective optimization. Natural Computing 5, 305–319.

Schuetze, O., Laumanns, M., Tantar, E., Coello Coello, C.A., Talbi, E.-G. (2007). Convergence of stochastic search algorithms to gap-free Pareto front approximations. Proc. GECCO '07, London, ACM, 892-899.

Spronk, J, Steuer, R.E., Zopounidis, C. (2005). Multicriteria decision analysis/aid in finance. In: Multiple Criteria Decision Analysis: State of the Art Surveys. International Series in Operations Research and Management Science 78, Springer New York, 799–848.

Teich, J. (2001). Pareto-front exploration with uncertain objectives. Proc. EMO '01 (Evolutionary Multi-Criterion Optimization), Springer, 314–328.

APPENDIX

Proof of Corollary 2: From (17), we see that for $k \geq k^+$, the lower bound in (16) is nonnegative. Considering (9), (6) and the independence of the random variables $\xi_i(x, \omega_\nu)$ for different $x \in X$, and observing that the power function u^p (p even) is nondecreasing as long as $u \geq 0$, we obtain from Lemma 2:

$$\eta_k = \pi_k^{2^n} \geq \left[1 - \frac{4\sigma}{r\sqrt{2\pi k}} \exp\left(-\frac{r^2 k}{2\sigma^2}\right)\right]^{2^n} \quad (k \geq k^+).$$

From this inequality, the statement follows by means of the inequality $(1-z)^p \geq 1 - pz$ ($z \in \mathbb{R}$, p even). □

Proof of Corollary 3: If the size of a sample of i.i.d. random noise variables is increased, the variance of the sample average decreases. As a consequence, the probability that the error terms of the sample average estimates over a sample of size $\bar{s}_k = mk$ with $m > 1$ all lie in $Q_r(\mathbf{0})$ is larger or equal to the corresponding probability for the case $m = 1$. Therefore, the choice $m = 1$ is conservative w.r.t. the lower bound on η_k^*. For $m = 1$, the only difference to Corollary 2 is that now, not only in 2^n, but in $2^n + 2^n = 2^{n+1}$ random experiments, the error terms have to be sufficiently small. This immediately yields the indicated formula. □

Proof of Corollary 4: By Corollary 3,

$$1 - \eta_k^* \;\leq\; \frac{\sigma}{r\sqrt{2\pi k}} \exp\left((n+3)\,\ln 2 - \frac{r^2 k}{2\sigma^2}\right) \;\leq\; \frac{\sigma}{r\sqrt{2\pi}} \exp\left((n+3)\,\ln 2 - \frac{r^2 k}{2\sigma^2}\right)$$

for each $k \geq k^+$. The assertion follows since if $k \geq (2\sigma^2/r^2) \cdot ((n+3)\ln 2 + C)$ in addition to $k \geq k^+$, then the argument of "exp" in the rightmost expression above is smaller or equal to $-C$. □