Stochastic Local Search in Continuous Domains

Questions to be Answered When Designing A Novel Algorithm

Petr Pošík
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Technická 2, 166 27 Prague 6, Czech Republic
posik@labe.felk.cvut.cz

ABSTRACT

Several population-based methods (with origins in the world of evolutionary strategies and estimation-of-distribution algorithms) for black-box optimization in continuous domains are surveyed in this article. The similarities and differences among them are emphasized and it is shown that they all can be described in a common framework of *stochastic local search*—a class of methods previously defined mainly for combinatorial problems. Based on the lessons learned from the surveyed algorithms, a set of algorithm features (or, questions to be answered) is extracted. An algorithm designer can take advantage of these features and by deciding on each of them, she can construct a novel algorithm. A few examples in this direction are shown.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Black-box optimization, Estimation-of-distribution algorithm, Evolutionary strategy, Covariance matrix adaptation, Premature convergence, Probabilistic modeling, Gaussian distribution, Cauchy distribution

1. INTRODUCTION

Estimation-of-distribution algorithms (EDAs) [25, 32] are established among the most successful optimization algorithms for discrete and combinatorial problems. They allow to reasonably balance the global view of the search space (exploration) with the local view (exploitation) by decomposing the optimization problem, by modeling the dependencies among the design variables. Bayesian networks (BN) [20] are de-facto standard model used to encode the joint probability density function (p.d.f.), allowing the user to apply some apriori structural constraints. The basic method of BN fitting is the maximum likelihood estimation (MLE). In discrete domains, MLE esentially works, although it is often complemented with simple remedies ensuring that the right model can be found even when the population is not large enough, or when the initialization happens to create the initial finite population not properly.

In real-valued optimization, the situation is more complex. The probability density functions can have many different forms and the dependencies among variables may be hard to find. It is very hard to find a general model, that would encompass all possible density functions, and in the same time would be sufficiently simple allowing for easy parametrization and learning. A direct analogy to Bayesian network from discrete domain is the Gaussian network (GN) [24, 7]; GN has similar structure and similar parametrization as BN, yet it describes a single-peak Gaussian distribution only. Despite that, there are several attempts to transfer the principles of EDAs from discrete to continuous domains [5, 24, 30, 31, 1, 33, 22, 26]. But many of these approaches use some kind of MLE learning and have huge problems to find the solution if it lies beyond the convex envelope of the population members when it is needed to shift the whole population towards the optimum. Results of several benchmark studies (see e.g. the Black-box Optimization Benchmarking workshop at GECCO) suggest that for a broad range of functions it is better to have a fast local optimizer and restart it often than to have an EDA algorithm with a sophisticated probabilistic model.

This article is an updated version of the author's previously published work [36] and deals with certain class of evolutionary algorithms that lie on the boundary of EDAs and evolutionary strategies (ES) that use unimodal probabilistic models and thus exhibit a kind of local search behaviour.

Stochastic local search (SLS) methods originated in the field of combinatorial optimization and they are claimed to be among the most efficient optimizers. In [18] they are informally described as 'local search algorithms that make use of randomized choices in generating or selecting candidate solutions for a given combinatorial problem instance.' As noted in [42], 'once randomized and appended or hybridized with a local search component, these (SLS techniques) include a wealth of methods such as simulated annealing, iterated local search, greedy randomized adaptive search, variable neighbourhood search, ant colony optimization, among

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
Copyright 2010 ACM 978-1-4503-0073-5/10/07 ...$10.00.

others,' which are usually classified as global search heuristics. The *locality* is usually induced by the perturbation operators used to generate new solutions.

In this paper, the term *stochastic local search* is used to describe algorithms for continuous optimization as well. The local neighborhood is often not given by a perturbation operator, but rather by a single-peak probability distribution function (p.d.f.) with decaying tails (very often Gaussian). Even though generally the value of p.d.f. is non-zero for any point in the feasible space, the offspring are concentrated 'near' the distribution center. Due to this fact, many of these algorithms exhibit a kind of hill-climber behaviour, which is, according to the author, sufficient to describe them as SLS.

The algorithms discussed in this article arised mainly from two sources: evolutionary strategies and estimation of distribution algorithms. Evolution strategies (ES) (see e.g. [3] for recent introduction) were among the first algorithms for continuous black-box optimization which employed a stochastic component. Their first versions were purely mutative and used $(1+1)$, or $(1 \overset{+}{,} \lambda)$ selection operators. The individual solutions were coupled with the distribution parameters which underwent the evolution along with the candidate solutions. Later, they were generalized to $(\mu \overset{+}{,} \lambda)$-ES which were still only mutative, but allowed the search to take place in several places of the search space in parallel. With μ parents, it became possible to introduce the crossover operator which is considered to be the main evolutionary search operator in the field of genetic algorithms (GA) [8], and the multi-recombinant ES were born [38]. The notation $(\mu/\rho \overset{+}{,} \lambda)$-ES means that out of the μ parents, ρ of them were recombined to become a center for one of the λ generated offspring. After another two decades of research, the state-of-the-art evolutionary strategy with covariance matrix adaptation (CMA-ES) was developed [17]. Rather surprisingly, it is a kind of $(\mu/\mu, \lambda)$-ES—the computation of the center of thenormal distribution is based on all selected parents, i.e. the same distribution is used to generate all candidate solutions. Even though the CMA-ES is a multi-recombinant ES, in each generation it searches the neighborhood of 1 point and thus exhibits local search behaviour (given the step size is small compared to the size of the search space which is often the case).

The second source of SLS lies in estimation-of-distribution algorithms (EDAs) [25]. There are many variants that use multimodal distributions, but here we are concerned with unimodal ones. The first continuous EDAs modeled all variables independently (e.g. [39, 23]). EDAs using full covariance matrix [24] and EDAs using Bayesian factorization of the normal distribution [5] emerged shortly thereafter. These EDAs used almost exclusively maximum-likelihood estimation (MLE) of the distribution parameters—a method that was very successful in case of discrete EDAs. However, it turned out very soon [7, 46, 29, 11, 9] that MLE leads in case of normal distribution to premature convergence even if the population is situated on the slope of the fitness function! Various remedies of this problem emerged [10, 6, 34, 37, 4].

In both areas, ES and EDAs, articles discussing the use of solutions discarded by the selection can be found. The discarded solutions can be used to speed up the adaptation of covariance matrix [19, 2], or a completely novel method of learning the distribution can be constructed [37].

Algorithm 1: Continuous Stochastic Local Search

```
input  : The type of model $\mathcal{M}$
output: The best solution found do far
begin
    $\mathcal{M}^{(0)} \leftarrow$ InitializeModel()
    $X^{(0)} \leftarrow$ Sample($\mathcal{M}^{(0)}$)
    $f^{(0)} \leftarrow$ Evaluate($X^{(0)}$)
    $g \leftarrow 1$
    while not TerminationCondition() do
        $\{\mathcal{S}, \mathcal{D}\} \leftarrow$ Select($X^{(g-1)}$, $f^{(g-1)}$)
        $\mathcal{M}^{(g)} \leftarrow$ Update($g$, $\mathcal{M}^{(g-1)}$, $X^{(g-1)}$, $f^{(g-1)}$, $\mathcal{S}$, $\mathcal{D}$)
        $X' \leftarrow$ Sample($\mathcal{M}^{(g)}$)
        $f' \leftarrow$ Evaluate ($X'$)
        $\{X^{(g)}, f^{(g)}\} \leftarrow$ Replace($X^{(g-1)}$, $X'$, $f^{(g-1)}$, $f'$)
        $g \leftarrow g + 1$
```

As already stated, all the above mentioned algorithms can be described as instances of stochastic local search. In the next section, these key works in this field are surveyed in greater detail. In section 3 the taxonomy of these methods is constructed based on their commonalities and differences. Section 4 proposes a few new possibilities offered by the taxonomy and section 5 concludes the paper.

2. OVERVIEW OF SLS TECHNIQUES

A detailed, thorough, and in-depth comparison of some algorithms relevant to this paper can be found in [21]. This article, on the other hand, is aimed especially at describing the main distinguishing features of more recent algorithms.

A general continuous SLS algorithm with single-peak distribution can be described in high-level as Alg. 1. This formulation can accommodate

- *comma* (generational) and *plus* (steady-state) evolutionary schemes,
- use of *selected and/or discarded* solutions in the model adaptation phase (thanks to the `Selection` operator that returns indices of selected individuals, $\mathcal{S}$, as well as indices of discarded individuals, $\mathcal{D}$),
- model *adaptation* (the previous model, $\mathcal{M}^{(g-1)}$, enters the `Update` phase),
- *self-adaptation* (the model parameters can be part of the population $X^{(g-1)}$),
- *deterministic* model adaptation (a predefined schedule depending on the generation counter g can be used for the model parameters), and
- *feedback* model adaptation (the information on the current population state can be used to adapt the model).

Individual algorithms mentioned in the introduction can all be described in this framework. They will generally differ in the definition of the model, $\mathcal{M}$, and in the operations `Update` and `Sample`.

Stochastic hill-climbing with learning by vectors of normal distributions (SHCLVND) [39] uses normal distribution for sampling. The model has the form of $\mathcal{M} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$. In the update phase, it uses a Hebbian learning rule to adapt the center of the distribution, $\boldsymbol{\mu}^{(g)} = \boldsymbol{\mu}^{(g-1)} + \mu_{\text{move}}(\bar{X}_{\mathcal{S}} - \boldsymbol{\mu}^{(g-1)})$, so that the new center is between the old center and the mean of the selected individuals, $\bar{X}_{\mathcal{S}}$, or possibly behind the mean. The spread of the distribution is adapted using deterministic schedule as $\boldsymbol{\sigma}^{(g)} = c_{\text{reduce}}\boldsymbol{\sigma}^{(g-1)}$, $c_{\text{reduce}} \in (0, 1)$.

Univariate marginal distribution algorithm for continuous domains (UMDA$_C$) [23] also does not take into account any dependencies among variables. This algorithm performs some statistical tests in order to determine which of the theoretical density functions fits the particular variable best.

Maximum-likelihood Gaussian EDA (ML-G-EDA) uses a Gaussian distribution with full covariance matrix $\boldsymbol{\Sigma}$ to generate new candidate solutions. Its model has the form of $\mathcal{M} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$. Model is not adapted, it is created from scratch as ML estimate based on the selected individuals only. The update step is thus $\boldsymbol{\mu}^{(g)} = \bar{X}_{\mathcal{S}}$ and $\boldsymbol{\Sigma}^{(g)} = \text{covmat}(X_{\mathcal{S}})$.

This algorithm is highly prone to premature convergence [7, 46, 29, 11, 9]. To improve the algorithm, several approaches were suggested.

Variance scaling (VS) [46] is the most simple approach for ML-G-EDA improvement. The change is in the sampling phase: the covariance matrix $\boldsymbol{\Sigma}$ is substituted with enlarged one, $c\boldsymbol{\Sigma}$, $c \geq 1$. In [35] it was shown that this approach with multivariate normal distribution leads in higher-dimensional spaces either to premature convergence on the slopes of the fitness function, or to the divergence of the distribution in the neighborhood of the optimum.

Adaptive variance scaling (AVS) [10] is also a method to enlarge the covariance matrix in ML-G-EDA. It uses the model in the following form: $\mathcal{M} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, c_{\text{AVS}}, f_{\text{BSF}}\}$. The covariance matrix enlargement factor c_{AVS} becomes part of the model and is adapted on the basis if the best-so-far (BSF) solution was improved. In the update phase, c_{AVS} is increased ($c^{(g)}_{\text{AVS}} = \eta \cdot c^{(g-1)}_{\text{AVS}}$) if the best solution in $X_{\mathcal{S}}$ is of better quality than f_{BSF}, or decreased ($c^{(g)}_{\text{AVS}} = c^{(g-1)}_{\text{AVS}}/\eta$), otherwise.

Correlation trigger (CT) was introduced in the same article [10] as AVS. It was observed that the AVS slows down the algorithm convergence in situations when the distribution is centerd around the optimum of the fitness function. In that cases, the c_{AVS} multiplier is too high and it takes several generations to decrease it to reasonable values. A better approach is to trigger the AVS only when the population is on the slope, otherwise pure MLE of variance is used. The rank correlation between the values of probability density function (p.d.f.) and fitness values was used as the AVS trigger. If the population is on the slope, the correlation will be low, while in the valley the absolute value of correlation will be high (assuming minimization, with decreasing value of p.d.f. the fitness increases).

It is important to note, that this approach can be in principle used to scale individual main axes (instead the whole distribution) which would effectively change the shape of the distribution learned by MLE. However, difficulties in higher-dimensional spaces can be expected.

Standard-deviation ratio (SDR) [6] was later used instead of CT which fails in higher-dimensional spaces. SDR triggers AVS in cases when the improvements are found far away from the distribution center (if the distance of the average of all improving solutions in the current generation is larger than a threshold).

Anticipated mean shift (AMS) [4] is another scheme for fighting the premature convergence. In fact, it belongs to this article only partially: the parameters of the distribution are estimated as in the case of single-peak Gaussian, however, in the sampling phase 2-peak distribution is used (with the same shape parameters, but different means). This way, part of the offspring is artificially moved in the direction of estimated gradient (anticipated mean shift). It is assumed that if this prediction was right, then the shifted solutions will be selected along with some of the non-shifted solutions which in turn increases the variance in the direction of the gradient.

Reweighting is another simple modification of the Gaussian EDA introduced in [41]. It is aimed at the fact that when population shift towards the optimum is needed, its estimated position (given as the average of selected points) is biased towards the center of the distribution used to sample the points. The reweighting ensures that the selected points with lower values of p.d.f.have higher weight in the estimation of position of the optimum. This way the algorithm is able to make larger steps.

Other than normal distributions were explored in several works. In [45], anisotropic Cauchy distribution was used to fasten ES, but the algorithm actually exploits separability of the problem as shown in [28, 16]. In [34], the Gaussian, isotropic Gaussian, and isotropic Cauchy distributions were compared from the point of view if non-adaptive variance scaling (VS) is sufficient to preserve the needed diversity. The isotropic Cauchy distribution was the most promising. The shape and the center of the distribution were estimated using MLE for the Gaussian distribution in all cases. In subsequent experiments it turned out that this approach fails e.g. on ridge functions.

Evolutionary strategy with covariance matrix adaptation (CMA-ES) [17] is currently considered the state-of-the-art technique in numerical optimization. This algorithm nowadays exists in several variants, but all have some common features. Detailed description of CMA-ES is beyond the scope of this paper; note that its model is given by $\mathcal{M} = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}, c, \boldsymbol{p}_c, \boldsymbol{p}_\sigma\}$. CMA-ES differs from other approaches

1. by using a kind of aggregated memory, the so called evolution paths $\boldsymbol{p}_c$ and $\boldsymbol{p}_\sigma$, which are cumulated over generations and used to adapt $\boldsymbol{\Sigma}$ and c, and
2. by estimating the distribution of selected mutation steps, rather than distribution of selected individuals.

CMA-ES using also the discarded individuals was proposed in [19], and further discussed in [2]. The covariance matrix adaptation mechanism uses also the discarded individuals with negative weights. The covariance matrix $\boldsymbol{\Sigma}$ might lose its positive definiteness, but in practice it does not happen. Speedups of the order of 2 were observed using this strategy in high-dimensional spaces with large populations.

Natural Evolution Strategies (NES) were suggested in [43] and later improved to efficient NES (eNES) in [40]. This algorithm uses the Gaussian distribution as the model. Its unique feature is the learning algorithm that uses the so-called natural gradient, i.e. gradient in the space of the distribution parameters, and follows it towards better expected fitness. The authors claim that this approach is more principled and needs lower number of parameters than the update step in CMA-ES.

Optimization via classification was explored in the context of single-Gaussian-based SLS in [37]. Both the selected and the discarded individuals are used to train a classifier that distinguishes between them. If the classifier has a suitable structure (quadratic discriminant function), it can be transformed into a probabilistic model (Gaussian). Similar idea was used before in discrete space [27], or in continuous spaces [44], but the classifier in that case was not transformable to a single peak distribution and is thus out of the scope of this article.

Adaptive encoding (AE) [12] is not directly an optimization algorithm. In continuous domain, the ability to find the right rotation of the search space is crucial. AE decouples the space transformation part from the CMA-ES and makes it available for any search algorithm, especially for the single-peak SLS. To decouple the transformation part from the optimization algorithm was proposed also in other works, e.g. in [33].

3. QUESTIONS FOR CONTINUOUS SLS

The algorithms that were just described can be categorized from many points of view which are (to some extent) independent of each other. These points of view can be represented as a set of questions and novel algorithms can be constructed just by answering them.

3.1 Model Sampling

One of the most important aspects of any algorithm is the choice of the sampling distribution $\mathcal{P}$. The sampling process then reads

$$\boldsymbol{z}_i \sim \mathcal{P}, \tag{1}$$

$$\boldsymbol{x}_i = \boldsymbol{\mu} + \boldsymbol{R} \times \mathrm{diag}(\boldsymbol{\sigma}) \times (c \cdot \boldsymbol{z}_i). \tag{2}$$

Here it is assumed that single-peak origin-centered base distributions $\mathcal{P}$ (non-parametric, or with fixed parameters) is used to sample new raw candidate solutions, $\boldsymbol{z}_i$. The distribution is enlarged as a whole by multiplying it with a global step size c and elongated along the coordinate axes by multiplying each dth coordinate with the respective multiplicator σ_d ($\mathrm{diag}(\boldsymbol{\sigma})$ is a diagonal matrix with entries σ_d on the diagonal). The sampled points are rotated by using the rotation matrix $\boldsymbol{R}$ with orthonormal columns. The origin of the distribution is then changed by adding the position vector $\boldsymbol{\mu}$. The model parameters $\boldsymbol{\mu}$, c, $\boldsymbol{R}$, and $\boldsymbol{\sigma}$ are created in the model building phase. The base distribution $\mathcal{P}$ is usually fixed during the whole evolution.

Regarding the model sampling, the individual algorithms can differ in the following aspects:

Question 1. *What kind of base distribution $\mathcal{P}$ is used for sampling?*

The majority of algorithms use standard Gaussian distribution. In [34], scaled versions of isotropic Gaussian and isotropic Cauchy distributions[1] were analyzed. The distributions were scaled by a constant so that if the distribution was centered around the optimum of a sphere function, then after selecting τN best individuals, the distance of the furthest is expected to be 1. This is done by dividing the points sampled from the base distribution by the critical point of the inverse cumulative distribution function, e.g. in case of isotropic Cauchy distribution the sampling was modified as follows: $\boldsymbol{x} \sim \mathcal{C}^{\mathrm{iso}}$, $\boldsymbol{x}_m = \boldsymbol{x}/CDF_{\mathcal{C}}^{-1}(\frac{1+\tau}{2})$.

Question 2. *Is the type of distribution fixed during the whole evolution?*

Switching the types of probabilistic models is not quite common, but was already employed on the basis of individual axes [23]. It is also possible to change the type of model as a whole.

3.2 Model Building

In the phase of model building, there are two main tasks

1. set the model parameters $\boldsymbol{\mu}$, c, $\boldsymbol{R}$, and $\boldsymbol{\sigma}$ directly used in sampling, and
2. set the auxiliary strategy-specific parameters (cumulation paths, best-so-far solution, or other statistics describing the past evolution).

Again, several distinctive features can be observed:

Question 3. *Is the model re-estimated from scratch each generation? Or is it updated incrementaly?*

The model parameters can be set *only* on the basis of the individuals in the current population (like in ML-EDA) which is usually the simpler choice. On the other hand, if one knows an efficient method for updating the model incrementaly, it usually results in more stable behaviour of the algorithm.

Question 4. *Does the model-building phase use selected and/or discarded individuals?*

The discarded individuals are used in only a few works even though they offer various possibilities.

Question 5. *Where do you place the sampling distribution in the next generation?*

Answer to this question amounts to defining the equation for setting $\boldsymbol{\mu}^{(g)}$. The next sample can be centered around the best individual in the current population, around the best-so-far individual, around the mean of the selected individuals (ML-EDA), around the weighted mean of the best individuals (CMA-ES), around a place where we anticipate that the mean should move (AMS), etc.

Question 6. *How much should the distribution be enlarged?*

In other words, what should the global step-size setting be? The global step-size c can be 1 (ML-EDA), a constant (VS), or it can be adapted (AVS), or eventually used only sometimes (CT, SDR).

Question 7. *What should the shape of the distribution be?*

The answer lies in the way of computing the rotation matrix $\boldsymbol{R}$ and scaling factors $\boldsymbol{\sigma}$. These are usually closely re-

[1] These isotropic distributions are sampled so that (1) a direction vector is selected uniformly by selecting a point on hypesphere, and (2) this direction vector is multiplied by a radius sampled from 1D Gaussian or Cauchy distribution, respectively.

lated. They can be set e.g. by eigendecomposition of the covariance matrix of the selected data points (ML-EDA). In that case the $\boldsymbol{\sigma}$ is a vector of standard deviations in the main axes and $\boldsymbol{R}$ is a matrix of vectors pointing in directions of the main axes. AE offers an alternative way of estimating the $\boldsymbol{\sigma}$ and $\boldsymbol{R}$.

Question 8. *What should the reference point*[2] *be?*

Closely related to the previous feature, it has a crucial impact on the algorithm behaviour. If we take the selected data points $X_{\mathcal{S}}$, subtract their mean $\bar{X}_{\mathcal{S}}$, and perform the eigendecomposition

$$[\boldsymbol{\sigma}^2, \boldsymbol{R}] \leftarrow \mathrm{eig}(X_{\mathcal{S}}, \bar{X}_{\mathcal{S}})$$

where

$$\mathrm{eig}(X, x_r) \overset{\mathrm{def}}{=} \mathrm{eig}\left(\frac{1}{N-1}(X - x_r)(X - x_r)^T\right)$$

we arive at the approach ML-EDAs are using. If we change the reference point x_r from $\bar{X}_{\mathcal{S}}$ to $\boldsymbol{\mu}^{(g-1)}$ (so that $[\boldsymbol{\sigma}^2, \boldsymbol{R}] \leftarrow \mathrm{eig}(X_{\mathcal{S}}, \boldsymbol{\mu}^{(g-1)})$), then we get the principle behind CMA-ES—we estimate the distribution of selected mutation steps.

4. DESIGNING NOVEL SLS

Since many of the features are independent of each other, it makes sense to create new algorithms by combining previously unexplored combinations, and asses the quality of the newly constructed algorithms. Let us briefly discuss some possibilities for the reference point of the distribution shape estimate (question 8, Q8) if we allow the model building phase to use the selected as well as discarded solutions (Q4). In the following, the Gaussian distribution is used (Q1) in all generations (Q2). The model is re-estimated from scratch each generation with the exception of CMA-ES-like configuration where $\boldsymbol{\mu}^{(g-1)}$ is used as the reference point (Q3). A conservative setting was chosen for the distribution center in this article: $\bar{X}_{\mathcal{S}}$ is used similarly to ML-EDA (Q5). The distribution is not enlarged, $c = 1$ (F6), and the shape is estimated by eigendecomposition of XX^T matrix of certain vectors in X (Q7).

Interesting configurations can be envisioned in setting the reference point for estimation of the distribution shape. Let $\bar{X}_B$ and $\bar{X}_W$ be (possibly weighted) averages of several best selected and several best discarded solutions in the current population, respectively. $X_{\mathcal{S}}$ and $X_{\mathcal{D}}$ are solutions selected and discarded, respectively, by the selection operator. Furthermore, $X_B \subset X_{\mathcal{S}}$ and $X_W \subset X_{\mathcal{D}}$. Instead of using $\mathrm{eig}(X_{\mathcal{S}}, \bar{X}_{\mathcal{S}})$ (which is ML-EDA and converges prematurely, see Fig. 1, upper left) or $\mathrm{eig}(X_{\mathcal{S}}, \boldsymbol{\mu}^{(g-1)})$ (which is successful CMA-ES-like approach, see Fig. 1, upper right), we can use $\mathrm{eig}(X_{\mathcal{S}}, \bar{X}_B)$ (see Fig. 1, lower left). Compared to ML-EDA, this might result in greater spread in the gradient direction on the slope, but as a whole the estimates are still too low and the algorithm converges prematurely. In the neighborhood of the optimum, the MLE is recovered since $\bar{X}_B$ is expected to be the same as $\bar{X}_{\mathcal{S}}$.

Another option is to use $\mathrm{eig}(X_W, \bar{X}_B)$ (see Fig. 1, lower right). As can be seen, it might give the algorithm additional burst, since it located the optimum after 50 generations which is not the case for any other configuration.

4.1 BBOB: Stage for Algorithm Comparisons

It is, of course, impossible to draw some far-reaching conclusions based on the pictures presented in Fig. 1. Statistical analysis on broad class of problems and dimensionalities is needed and it is questionable if some of these methods can beat the finely tuned CMA-ES.

At GECCO 2010, the black-box optimization benchmarking workshop (BBOB) is held for the second time. It provides a well-thought methodology [13] of comparing various algorithms for numerical optimization on a reasonably chosen set of noise-free [14] and noisy [15] fitness functions. It also provides post-processing scripts (which produce a bunch of tables and graphs) and LaTeX article templates, so that everything that is left to the experimenter is the description of the algorithm and the discussion of the results. The experimenter can also freely choose among the algorithms that were benchmarked in the past and use any of them for comparison with her own algorithm.

The BBOB methodology has high chances to become a standard algorithm comparison tool. Constructing various algorithms suggested in this paper and comparing them with others using BBOB is a valuable direction for future work.

5. CONCLUSIONS

This paper surveyed recent contributions in the area of SLS techniques using single-peak search distribution. A broad set of methods and tweaks exist in this field—various similarities and differences were pointed out. Based on the lessons learned from these methods, a set of rather independent features was compiled; these features can be used to categorize various SLS techniques from many points of view. This schema offers also many previously unexplored feature combinations that can result in potentialy successful algorithms. Exploring these various possibilities remains as a future work.

6. ACKNOWLEDGMENTS

The author is supported by the Grant Agency of the Czech Republic with the grant no. 102/08/P094 entitled “Machine learning methods for solution construction in evolutionary algorithms”.

7. REFERENCES

[1] C. W. Ahn, R. S. Ramakrishna, and D. E. Goldberg. Real-coded bayesian optimization algorithm: Bringing the strength of BOA into the continuous world. In K. Deb, editor, *Proceedings of the Genetic and Evolutionary Computation Conference Ű GECCO 2004*, pages 840–851. Springer-Verlag, Berlin, 2004.

[2] D. V. Arnold and D. C. S. Van Wart. Cumulative step length adaptation for evolution strategies using negative recombination weights. In M. G. et al., editor, *EvoWorkshops 2008*, volume 4974 of *Lecture Notes in Computer Science*, pages 545–554. Springer, 2008.

[3] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, May 2002.

[4] P. Bosman, J. Grahl, and D. Thierens. Enhancing the performance of maximum-likelihood Gaussian EDAs

[2] Note the difference between the model center (see feature 5) and the reference point. We can e.g. estimate the shape of the distribution based on selected points when the worst point is taken as the reference, and then center the learned distribution around the best point.

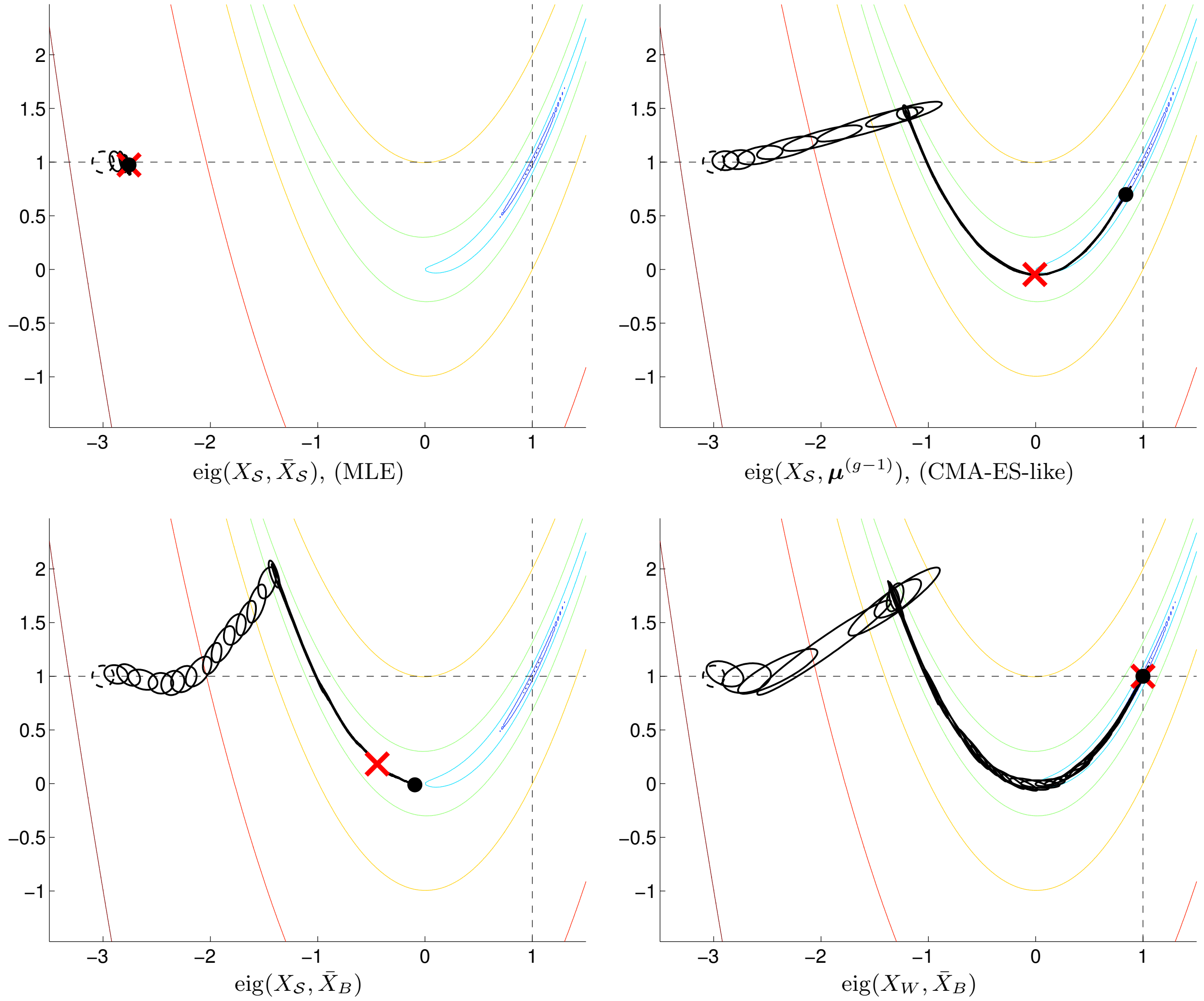

Figure 1: SLS with various approaches of estimating the shape of the distribution on the 2D Rosenbrock function. Red cross: $\boldsymbol{\mu}^{(50)}$, black dot: $\boldsymbol{\mu}^{(100)}$. Initialization: $\boldsymbol{\mu}^{(0)} = (-3, 1)$, $\boldsymbol{\sigma}^{(0)} = (0.1, 0.1)$. No enlargement of the estimated shape takes place, $c = 1$. Population size 200 and truncation selection with selection proportion $\tau = 0.3$ was used for all pictures.

using anticipated mean shift. In G. R. et al., editor, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *LNCS*, pages 133–143. Springer, 2008.

[5] P. A. Bosman and D. Thierens. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 197–200, 2000.

[6] P. A. N. Bosman, J. Grahl, and F. Rothlauf. SDR: A better trigger for adaptive variance scaling in normal EDAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, pages 492–499, New York, NY, USA, 2007. ACM Press.

[7] P. A. N. Bosman and D. Thierens. Expanding from discrete to continuous estimation of distribution algorithms: The IDEA. In *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, pages 767–776, London, UK, 2000. Springer-Verlag.

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

[9] C. Gonzales, J. A. Lozano, and P. Larrañaga. Mathematical modelling of UMDAc algorithm with tournament selection. *International Journal of Approximate Reasoning*, 31(3):313–340, 2002.

[10] J. Grahl, P. A. N. Bosman, and F. Rothlauf. The correlation-triggered adaptive variance scaling IDEA. In *Proceedings of the 8th annual conference on Genetic and Evolutionary Computation Conference – GECCO 2006*, pages 397–404, New York, NY, USA, 2006. ACM Press.

[11] J. Grahl, S. Minner, and F. Rothlauf. Behaviour of UMDAc with truncation selection on monotonous functions. In *IEEE Congress on Evolutionary Computation, CEC 2005*, volume 3, pages 2553–2559, 2005.

[12] N. Hansen. Adaptive encoding: How to render search coordinate system invariant. In G. R. et al., editor, *Parallel Problem Solving from Nature Ű- PPSN X*, volume 5199 of *LNCS*, pages 205–214. Springer, 2008.

[13] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

[14] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.

[15] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009. Updated February 2010.

[16] N. Hansen, F. Gemperle, A. Auger, and P. Koumoutsakos. When do heavy-tail distributions help? In *Parallel Problem Solving from Nature – PPSN IX*, pages 62–71. Springer, 2006.

[17] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[18] H. H. Hoos and T. Stützle. *Stochastic Local Search : Foundations & Applications*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, 2004.

[19] G. A. Jastrebski and D. V. Arnold. Improving evolution strategies through active covariance matrix adaptation. In *IEEE Congress on Evolutionary Computation – CEC 2006*, pages 2814–2821, 2006.

[20] F. B. Jensen. *Bayesian Networks and Decision Graphs*. Springer New York, December 2009.

[21] S. Kern, S. D. Müller, N. Hansen, D. Büche, J. Očenášek, and P. Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms– a comparative review. *Natural Computing*, 3(1):77–112, 2004.

[22] P. Lanzi, L. Nichetti, K. Sastry, D. Voltini, and D. Goldberg. Real-coded extended compact genetic algorithm based on mixtures of models. In Y.-p. Chen and M.-H. Lim, editors, *Linkage in Evolutionary Computation*, volume 157 of *Studies in Computational Intelligence*, chapter 14, pages 335–358. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[23] P. Larrañaga, R. Etxeberria, J. A. Lozano, B. Sierra, I. Inza, and J. M. Peña. A review of the cooperation between evolutionary computation and probabilistic graphical models. In A. A. O. Rodriguez, M. R. S. Ortiz, and R. S. Hermida, editors, *CIMAF 99, Second Symposium on Artificial Intelligence*, Adaptive Systems, pages 314–324, La Habana, 1999.

[24] P. Larrañaga, J. A. Lozano, and E. Bengoetxea. Estimation of distribution algorithms based on multivariate normal distributions and Gaussian networks. Technical Report KZZA-IK-1-01, Dept. of Computer Science and Artificial Intelligence, University of Basque Country, 2001.

[25] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms*. GENA. Kluwer Academic Publishers, 2002.

[26] M. Li, D. Goldberg, K. Sastry, and T.-L. Yu. Real-coded ecga for solving decomposable real-valued optimization problems. In *Linkage in Evolutionary Computation*, pages 61–86. 2008.

[27] T. Miquèlez, E. Bengoetxea, A. Mendiburu, and P. Larrañaga. Combining Bayesian classifiers and estimation of distribution algorithms for optimization in continuous domains. *Connection Science*, 19(4):297–319, December 2007.

[28] A. Obuchowicz. Multidimensional mutations in evolutionary algorithms based on real-valued representation. *Int. J. Systems Science*, 34(7):469–483, 2003.

[29] J. Očenášek, S. Kern, N. Hansen, and P. Koumoutsakos. A mixed bayesian optimization algorithm with variance adaptation. In X. Yao, editor, *Parallel Problem Solving from Nature – PPSN VIII*, pages 352–361. Springer-Verlag, Berlin, 2004.

[30] J. Očenášek and J. Schwarz. Estimation of distribution algorithm for mixed continuous-discrete optimization problems. In *2nd Euro-International Symposium on Computational Intelligence*, pages 227–232, Košice, Slovakia, 2002. IOS Press. ISBN 1-58603-256-9, ISSN 0922-6389.

[31] T. Paul and H. Iba. Optimization in continuous

domain by real-coded estimation of distribution algorithm, 2003.

[32] M. Pelikan. *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*. Studies in Fuzziness and Soft Computing. Springer, 1 edition, March 2005.

[33] P. Pošík. *On the Use of Probabilistic Models and Coordinate Transforms in Real-Valued Evolutionary Algorithms*. PhD thesis, Czech Technical University in Prague, Prague, Czech Republic, 2007.

[34] P. Pošík. Preventing premature convergence in a simple EDA via global step size setting. In G. Rudolph, editor, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 549–558. Springer, 2008.

[35] P. Pošík. Truncation selection and Gaussian EDA: Bounds for sustainable progress in high-dimensional spaces. In M. Giacobini, editor, *EvoWorkshops 2008*, volume 4974 of *LNCS*, pages 525–534. Springer, 2008.

[36] P. Pošík. Stochastic local search techniques with unimodal continuous distributions: A survey. In M. Giacobini, A. Brabazon, S. Cagnoni, G. A. Caro, A. Ekárt, A. I. Esparcia-Alcázar, M. Farooq, A. Fink, and P. Machado, editors, *Applications of Evolutionary Computing*, volume 5484, chapter 78, pages 685–694. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[37] P. Pošík and V. Franc. Estimation of fitness landscape contours in EAs. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 562–569, New York, NY, USA, 2007. ACM Press.

[38] I. Rechenberg. Evolutionsstrategien. In Schneider, editor, *Simulationsmethoden in der Medizin and Biologie*, pages 83–113, Berlin, Germany, 1978. Springer Verlag.

[39] S. Rudlof and M. Köppen. Stochastic hill climbing by vectors of normal distributions. In *First Online Workshop on Soft Computing*, Nagoya, Japan, 1996.

[40] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient natural evolution strategies. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 539–546, New York, NY, USA, 2009. ACM.

[41] F. Teytaud and O. Teytaud. Why one must use reweighting in estimation of distribution algorithms. In *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 453–460, New York, NY, USA, 2009. ACM.

[42] S. Voss. Book review: H. H. Hoos and T. Stützle: Stochastic local search: foundations and applications (2005). *Mathematical Methods of Operations Research*, 63(1):193–194, 2006.

[43] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural evolution strategies. In *Proceedings of CEC 2008*, pages 3381–3387. IEEE, IEEE Press.

[44] J. Wojtusiak and R. S. Michalski. The LEM3 system for non-darwinian evolutionary computation and its application to complex function optimization. Reports of the Machine Learning and Inference Laboratory MLI 04-1, George Mason University, Fairfax, VA, February 2006.

[45] X. Yao and Y. Liu. Fast evolution strategies. *Control and Cybernetics*, 26:467–496, 1997.

[46] B. Yuan and M. Gallagher. On the importance of diversity maintenance in estimation of distribution algorithms. In H. G. Beyer and U. M. O'Reilly, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2005*, volume 1, pages 719–726, New York, NY, USA, 2005. ACM Press.