Stochastic Local Search Techniques with Unimodal Continuous Distributions: A Survey

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. In continuous black-box optimization, various stochastic local search techniques are often employed, with various remedies for fighting the premature convergence. This paper surveys recent developments in the field (the most important from the author's perspective), analyzes the differences and similarities and proposes a taxonomy of these methods. Based on this taxonomy, a variety of novel, previously unexplored, and potentially promising techniques may be envisioned.

1 Introduction

Stochastic local search (SLS) methods originated in the field of combinatorial optimization and they are claimed to be among the most efficient optimizers. In [1] 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 [2], 'once randomized and appended or hybridized by 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 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) [4], and the multi-recombinant ES were born [5]. 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 [6]. Rather surprisingly, it is a kind of $(\mu/\mu, \lambda)$-ES—the computation of the center of the normal 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) [7]. 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. [8], [9]). EDAs using full covariance matrix [10] and EDAs using Bayesian factorization of the normal distribution [11] emerged shortly thereafter. These EDAs used almost exclusively maximum-likelihood estimates (MLE) of the distribution parameters—an approach that was very successful in case of discrete EDAs. However, it turned out very soon ([12], [13], [14], [15], [16]) 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 ([17], [18], [19] [20], [21]).

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 ([22], [23]), or a completely novel method of learning the distribution can be constructed [20].

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 Selected SLS Techniques

A detailed, thorough, and in-depth comparison of some algorithms relevant to this paper can be found in [24]. 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,
- model *adaptation* (the previous model, $\mathcal{M}^{(g-1)}$, enters the `Update` phase),
- *self-adaptation* (the model parameters can be part of the population $X^{(g-1)}$),

Algorithm 1: Continuous Stochastic Local Search

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, $\mathcal{M}^{(g-1)}$, $X^{(g-1)}$, $f^{(g-1)}$, $\mathcal{S}$, $\mathcal{D}$`)`
    $X_{\mathrm{Offs}} \leftarrow$ `Sample(`$\mathcal{M}^{(g)}$`)`
    $f_{\mathrm{Offs}} \leftarrow$ `Evaluate(`X_{Offs}`)`
    $\{X^{(g)}, f^{(g)}\} \leftarrow$ `Replace(`$X^{(g-1)}$, X_{Offs}, $f^{(g-1)}$, f_{Offs}`)`
    $g \leftarrow g + 1$
end

– *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 (SHCL-VND) [8] 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, $\boldsymbol{\mu}^{(g)} = \boldsymbol{\mu}^{(g-1)} + \mu_{\mathrm{move}}(\bar{X}_{\mathcal{S}} - \boldsymbol{\mu}^{(g-1)})$, 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 $\boldsymbol{\sigma}^{(g)} = c_{\mathrm{reduce}}\boldsymbol{\sigma}^{(g-1)}$, $c_{\mathrm{reduce}} \in (0, 1)$.

Univariate marginal distribution algorithm for continuous domains (UMDA$_C$) [9] 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 $\boldsymbol{\Sigma}^{(g)} = \mathrm{covmat}(X_{\mathcal{S}})$.

This algorithm is highly prone to premature convergence ([12], [13], [14], [15], [16]). To improve the algorithm, several approaches were suggested.

Variance scaling (VS) [13] 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 [25] 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) [17] coupled with ML-G-EDA 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 ($c_{\text{AVS}}^{(g)} = \eta \cdot c_{\text{AVS}}^{(g-1)}$) if the best solution in X_S is of better quality than f_{BSF}, or decreased ($c_{\text{AVS}}^{(g)} = c_{\text{AVS}}^{(g-1)}/\eta$), otherwise.

Correlation trigger (CT) was introduced in the same article [17] 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).

Standard-deviation ratio (SDR) [18] 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) [21] 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.

Other than normal distributions were explored in several works. In [26], anisotropic Cauchy distribution was used to fasten ES, but the algorithm actually exploits separability of the problem as shown in [27],[28]. In [19], 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) [6] 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 [22], and further discussed in [23]. 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.

Optimization via classification was explored in the context of single-Gaussian-based SLS in [20]. 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 [29], or in continuous spaces [30], 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) [31] 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 [32].

3 Continuous SLS Taxonomy

The algorithms that were just described can be categorized from many points of view. These features are (to a great extent) independent of each other and new algorithms can be constructed just by deciding on each feature.

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

$$\boldsymbol{z}_i \sim \mathcal{P}, \tag{1}$$
$$\boldsymbol{x}_i = \boldsymbol{\mu} + c \cdot \boldsymbol{R} \times \mathrm{diag}(\boldsymbol{\sigma}) \times \boldsymbol{z}_i. \tag{2}$$

Here it is assumed that single-peak origin-centered base distributions $\mathcal{P}$ (non-parametric, or with fixed parameters) are 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:

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

The majority of algorithms use standard Gaussian distribution. In [19], 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.

Feature 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 [9]. 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:

Feature 3. *Is the model re-estimated from scratch each generation?*

The model parameters are set *only* on the basis of the individuals in the current population (like in ML-EDA).

Feature 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.

Feature 5. *How to determine the model center for 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 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.

Feature 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).

[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.

Feature 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 related. 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 $\boldsymbol{R}$.

Feature 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

$$[\boldsymbol{\sigma}^2, \boldsymbol{R}] \leftarrow \mathrm{eig}(X_{\mathcal{S}}, \bar{X}_{\mathcal{S}}), \text{ where } \mathrm{eig}(X, x_r) \stackrel{\mathrm{def}}{=} \mathrm{eig}\left(\frac{1}{N-1}(X - x_r)(X - x_r)^T\right)$$

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 New Possibilities for 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. Due to the space limitations, let us very briefly discuss only some possibilities for the reference point of the distribution shape estimate (feature 8, F8) if we allow the model building phase to use the selected as well as discarded solutions (F4). In the following, the Gaussian distribution is used (F1) in all generations (F2). 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 (F3). A conservative setting was chosen for the distribution center in this article: $\bar{X}_{\mathcal{S}}$ is used similarly to ML-EDA (F5). The distribution is not enlarged, $c = 1$ (F6), and the shape is estimated by eigendecomposition of XX^T matrix of certain vectors in X (F7).

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 $\bar{X}_{\mathcal{S}}$.

[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.

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.

It is, of course, impossible to draw some far-reaching conclusions based on the presented pictures. 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. In general, however, there are many choices better then the ML-EDA approach.

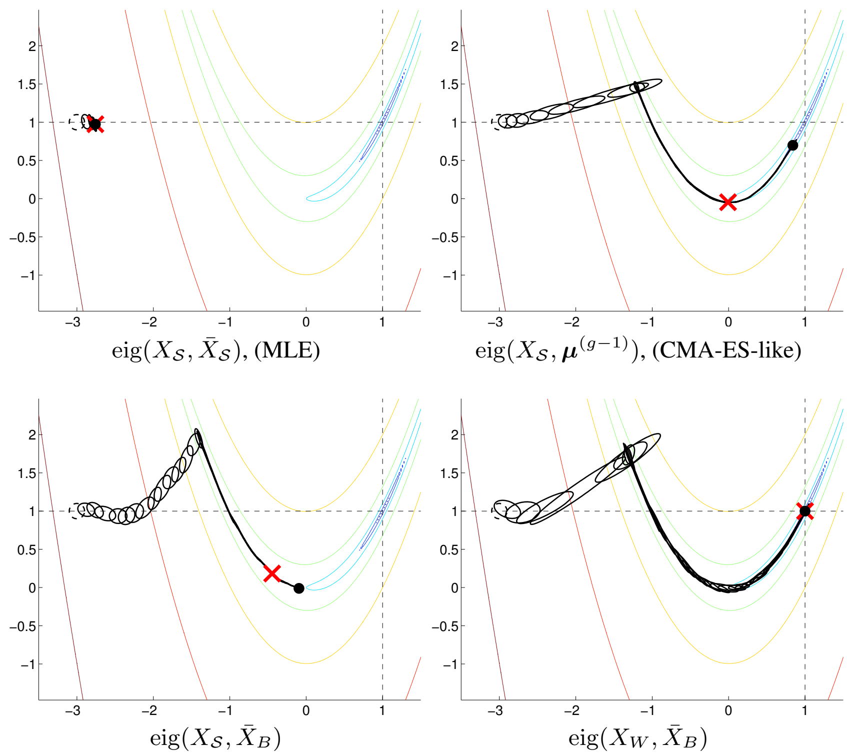

Fig. 1. SLS with various approaches of estimating the shape of the distribution on the 2D Rosenbrock function. Red cross: $\boldsymbol{\mu}^{(50)}$, black dot: $\boldsymbol{\mu}^{(100)}$. Initialization: $\boldsymbol{\mu}^{(0)} = (-3, 1)$, $\boldsymbol{\sigma}^{(0)} = (0.1, 0.1)$. No enlargement of the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

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 as a taxonomy to classify various SLS techniques. The taxonomy offers also many previously unexplored feature combinations that can result in potentialy successful algorithms. Exploring these various possibilities remains as a future work.

Acknowledgements

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”.

References

1. Hoos, H.H., Stützle, T.: Stochastic Local Search : Foundations & Applications. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann (2004)
2. Voss, S.: Book review: H. H. Hoos and T. Stützle: Stochastic local search: foundations and applications (2005). Mathematical Methods of Operations Research **63**(1) (2006) 193–194
3. Beyer, H.G., Schwefel, H.P.: Evolution strategies – a comprehensive introduction. Natural Computing **1**(1) (2002) 3–52
4. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
5. Rechenberg, I.: Evolutionsstrategien. In Schneider, ed.: Simulationsmethoden in der Medizin and Biologie, Berlin, Germany, Springer Verlag (1978) 83–113
6. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation **9**(2) (2001) 159–195
7. Larrañaga, P., Lozano, J.A., eds.: Estimation of Distribution Algorithms. GENA. Kluwer Academic Publishers (2002)
8. Rudlof, S., Köppen, M.: Stochastic hill climbing by vectors of normal distributions. In: First Online Workshop on Soft Computing, Nagoya, Japan (1996)
9. Larrañaga, P., Etxeberria, R., Lozano, J.A., Sierra, B., Inza, I., Peña, J.M.: A review of the cooperation between evolutionary computation and probabilistic graphical models. In Rodriguez, A.A.O., Ortiz, M.R.S., Hermida, R.S., eds.: CIMAF 99, Second Symposium on Artificial Intelligence. Adaptive Systems, La Habana (1999) 314–324
10. Larrañaga, P., Lozano, J.A., Bengoetxea, E.: 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)
11. Bosman, P.A., Thierens, D.: Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In: Workshop Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000). (2000) 197–200
12. Bosman, P.A.N., Thierens, D.: 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, London, UK, Springer-Verlag (2000) 767–776
13. Yuan, B., Gallagher, M.: On the importance of diversity maintenance in estimation of distribution algorithms. In Beyer, H.G., O’Reilly, U.M., eds.: Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2005. Volume 1., New York, NY, USA, ACM Press (2005) 719–726
14. Ocenasek, J., Kern, S., Hansen, N., Koumoutsakos, P.: A mixed bayesian optimization algorithm with variance adaptation. In Yao, X., ed.: Parallel Problem Solving from Nature – PPSN VIII, Springer-Verlag, Berlin (2004) 352–361

15. Grahl, J., Minner, S., Rothlauf, F.: Behaviour of UMDAc with truncation selection on monotonous functions. In: IEEE Congress on Evolutionary Computation, CEC 2005. Volume 3. (2005) 2553–2559
16. Gonzales, C., Lozano, J.A., Larrañaga, P.: Mathematical modelling of UMDAc algorithm with tournament selection. International Journal of Approximate Reasoning **31**(3) (2002) 313–340
17. Grahl, J., Bosman, P.A.N., Rothlauf, F.: The correlation-triggered adaptive variance scaling IDEA. In: Proceedings of the 8th annual conference on Genetic and Evolutionary Computation Conference – GECCO 2006, New York, NY, USA, ACM Press (2006) 397–404
18. Bosman, P.A.N., Grahl, J., Rothlauf, F.: 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, New York, NY, USA, ACM Press (2007) 492–499
19. Pošík, P.: Preventing premature convergence in a simple EDA via global step size setting. In: Parallel Problem Solving from Nature - PPSN X. Volume 5199 of Lecture Notes in Computer Science. Springer (2008) 549–558
20. Pošík, P., Franc, V.: Estimation of fitness landscape contours in EAs. In: Genetic and evolutionary computation conference – GECCO '07, New York, NY, USA, ACM Press (2007) 562–569
21. Bosman, P., Grahl, J., Thierens, D.: Enhancing the performance of maximum-likelihood Gaussian EDAs using anticipated mean shift. Volume 5199 of LNCS., Springer (2008) 133–143
22. Jastrebski, G.A., Arnold, D.V.: Improving evolution strategies through active covariance matrix adaptation. In: IEEE Congress on Evolutionary Computation – CEC 2006. (2006) 2814–2821
23. Arnold, D.V., Van Wart, D.C.S.: Cumulative step length adaptation for evolution strategies using negative recombination weights. In: EvoWorkshops 2008. Volume 4974 of LNCS., Springer (2008) 545–554
24. Kern, S., Müller, S.D., Hansen, N., Büche, D., Očenášek, J., Koumoutsakos, P.: Learning probability distributions in continuous evolutionary algorithms– a comparative review. Natural Computing **3**(1) (2004) 77–112
25. Pošík, P.: Truncation selection and Gaussian EDA: Bounds for sustainable progress in high-dimensional spaces. In: EvoWorkshops 2008. Volume 4974 of LNCS., Springer (2008) 525–534
26. Yao, X., Liu, Y.: Fast evolution strategies. Control and Cybernetics **26** (1997) 467–496
27. Obuchowicz, A.: Multidimensional mutations in evolutionary algorithms based on real-valued representation. Int. J. Systems Science **34**(7) (2003) 469–483
28. Hansen, N., Gemperle, F., Auger, A., Koumoutsakos, P.: When do heavy-tail distributions help? In: Parallel Problem Solving from Nature – PPSN IX, Springer (2006) 62–71
29. Miquèlez, T., Bengoetxea, E., Mendiburu, A., Larrañaga, P.: Combining Bayesian classifiers and estimation of distribution algorithms for optimization in continuous domains. Connection Science **19**(4) (December 2007) 297–319
30. Wojtusiak, J., Michalski, R.S.: 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)
31. Hansen, N.: Adaptive encoding: How to render search coordinate system invariant. In: Parallel Problem Solving from Nature - PPSN X. Volume 5199 of LNCS., Springer (2008) 205–214
32. Pošík, P.: 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)