Comparison-Based Optimizers Need Comparison-Based Surrogates

Ilya Loshchilov, Marc Schoenauer, Michèle Sebag

Comparison-Based Optimizers Need Comparison-Based Surrogates

Ilya Loshchilov[1,2], Marc Schoenauer[1,2], and Michèle Sebag[2,1]

[1] TAO Project-team, INRIA Saclay - Île-de-France**
[2] Laboratoire de Recherche en Informatique (UMR CNRS 8623)
Université Paris-Sud, 91128 Orsay Cedex, France
FirstName.LastName@inria.fr

Abstract. Taking inspiration from approximate ranking, this paper investigates the use of rank-based Support Vector Machine as surrogate model within CMA-ES, enforcing the invariance of the approach with respect to monotonous transformations of the fitness function. Whereas the choice of the SVM kernel is known to be a critical issue, the proposed approach uses the Covariance Matrix adapted by CMA-ES within a Gaussian kernel, ensuring the adaptation of the kernel to the currently explored region of the fitness landscape at almost no computational overhead. The empirical validation of the approach on standard benchmarks, comparatively to CMA-ES and recent surrogate-based CMA-ES, demonstrates the efficiency and scalability of the proposed approach.

1 Introduction

The importance of invariances in science has long been acknowledged. In computer science in particular, the invariance of an algorithm with respect to a given transformation of the problem domain is a source of robustness, as any theoretical or empirical result that is demonstrated for a given problem instance can be extended to the whole class of problems obtained by applying the transformation. For instance, many bio-inspired optimization algorithms such as tournament-based EAs, PSO, or DE only rely on comparisons of the fitness function, making them invariant under any monotonous transformation of the fitness. From a theoretical perspective, this invariance property is a source of robustness [5]; from an algorithmic perspective, it removes the need to tune the algorithm hyper parameters according to some (generally unknown) scale of the fitness function. In the realm of continuous optimization, the state-of-the-art CMA-ES [8] is known to achieve invariance with respect to orthogonal transformations of the search space. CMA-ES extreme robustness with respect to internal parameter tuning, and its outstanding performances for many types of fitness functions [7] are attributed in part to this invariance property, the importance of which is witnessed by the variability of other algorithm performances depending on e.g. the separability or condition number of the fitness function [1].

** Work partially funded by FUI of System@tic Paris-Region ICT cluster through contract DGT 117 407 *Complex Systems Design Lab* (CSDL).

Meta-model assisted optimization is used to decrease the number of evaluations of computationally expensive fitness functions in the framework of continuous optimization: a *surrogate model* of the fitness is built on the fly, and used *in lieu* of the actual fitness. Such method, also known as "Response Surface Method", has been used for long in the Numerical Engineering community [2]. Because Evolutionary Algorithms (EAs) require a lot of fitness evaluations, much work has been devoted in the last decade to specifically tune meta-model assisted approaches to EAs (see e.g. [9] for a survey – and section 2).

Unfortunately, building a surrogate model from the fitness values gathered during the search definitely obliterates any invariance by monotonous transformation of the fitness. Preserving such invariance in a surrogate-based approach requires the surrogate model to only comply with the ranks of the sample points with respect to the fitness function. In the realm of statistical Machine Learning, rank-based Support Vector Machines (SVMs) precisely aims at learning a model from the only ordering of the sample points [15]; such a rank-based surrogate could be used *in lieu* of a value-based surrogate, enforcing the comparison-based invariance of the underlying optimization algorithm. To the best of our knowledge, the only work investigating the use of rank-based SVM as surrogate model within a meta-model assisted EA is Runarsson's [14]; while this approach was reported to bring small improvements over a CMA-ES baseline, a major issue regards the choice of the kernel, the Achilles heel of all SVM-based methods.

Following the path opened by [14], this paper investigates the use of rank-based SVM surrogate models within CMA-ES. It further borrows [11] the use of the Covariance Matrix adapted by CMA-ES, viewed as the proper metric to look at the region of the fitness landscape currently explored. Finally, the paper contribution is to integrate a rank-based Support Vector Machine surrogate within CMA-ES, where the SVM kernel is set to the covariance matrix adapted by CMA-ES. Section 2 surveys evolutionary model-assisted approaches, focussing on rank-based surrogates and CMA-ES. Section 3 introduces the proposed algorithm, called ACM-ES for (alphabetically) ranked CMA-ES; it details how a rank-based SVM is tightly coupled with CMA-ES, using the change of representation induced by the current covariance matrix to derive an adaptive kernel with almost no computational overhead. The experimental validation of the approach is reported and discussed in section 4, and directions for further work are sketched in section 5.

2 Surrogate Models and Ranking

2.1 Approximate Ranking

Most approaches to evolutionary meta-model assisted optimization build and use the surrogate model in a way similar to that of classical optimization. The surrogate model is trained by regression, depending on the underlying model space (from mostly quadratic polynomials to neural networks, kriging aka Gaussian Processes or SVMs); the surrogate model (possibly taking into account its

uncertainty) is used *in lieu* of the actual fitness function, or it is used to pre-screen promising solutions; and the model is updated based on computations of the true fitness on those promising solutions (see [9] for a detailed survey).

To our best knowledge, the first work acknowledging the fact that EAs "only" require accurate ranking information, as opposed to accurate approximation of the fitness function, is that of Th. Runarsson [13]. This work introduced the idea of *approximate ranking*: a simple weighted nearest neighbor regression model is used as surrogate model, and its validity is assessed based on whether it preserves the (objective function-based) ordering of points. The most promising individuals according to the surrogate model are evaluated with the objective function until the ranking of the best individuals stabilizes. Significant savings are reported on test functions compared to the baseline algorithm (an ES variant tailored to constrained optimization), although they do not allow comparison with state-of-the-art results. Moreover, the surrogate model is probably too simple to lead to competitive results.

Approximate ranking however inspired *Local Meta-Model CMA-ES* (lmm-CMA), proposed by Kern et al. [11]: a local quadratic model is build anew for each offspring generated with the usual CMA-ES procedure, and approximate ranking is used to adaptively determine the number of actual objective evaluations to be run at each generation. lmm-CMA significantly outperforms the original CMA-ES with a speed-up factor of circa 2-3, making it competitive with state-of-the-art approaches. The requirement on approximate ranking was later relaxed by another variant, nlmm-CMA [3], using the rank stability of the set of μ best offspring as stopping criterion (as opposed to, the rank of each offspring), and thus improving over lmm-CMA on most benchmark functions (detailed results will be given in section 4.2 for the sake of comparative validation).

There are however a few drawbacks with lmm-CMA algorithms, apart from the fact that they use a regression surrogate model and hence depart from the comparison-based invariance of CMA-ES. Firstly, they rely on quadratic approximations, and thus their performances decrease when the objective function is far from being quadratic (see section 4.2). Secondly, they must use the full quadratic model [11], and hence the surrogate model must be of order d^2, where d denotes the problem dimension; the regression problem thus is of order d^6, which makes it hardly scalable for medium size problems ($d > 20$).

2.2 Rank-Based Surrogate Model with Rank-SVMs

Another seminal idea regarding the comparison-based issue in meta-model assisted EAs is again due to Th. Runarsson [14], using rank-based learning to train the surrogate model. Let us briefly recall rank-based Support Vector Machines, assuming the reader's familiarity with SVM first principles [15].

Let $(x_1, \ldots, x_N)$ denote an N-sample in instance space X, assuming with no loss of generality that point x_i has rank i. Rank-based SVM learning [10] aims at a real-valued function $\mathcal{F}$ on X such that $\mathcal{F}(x_i) < \mathcal{F}(x_j)$ for all pairs i, j such that $i < j$. In the SVM framework, this goal is formalized through minimizing the norm of $\mathcal{F}$ (regularization term) subject to the ordering constraints, thus

involving $N(N-1)/2$ constraints. A more tractable formulation [15] only involves the $N-1$ constraints related to consecutive points, $\mathcal{F}(x_i) < \mathcal{F}(x_{i+1})$ for $i = 1 \ldots N-1$. The latter formulation was used in [14] and will also be used in the presented approach.

Using the kernel trick[3], ranking function $\mathcal{F}$ is defined as a linear function w w.r.t. some feature space $\Phi(X)$, i.e. $\mathcal{F}(x) = \langle\ w, \Phi(x)\ \rangle$. With same notations as in [15], the primal optimization problem is defined as follows, where slack variable ξ_i and constant C_i respectively account for the violation of the i-th constraint, and the weight of the violation, to be minimized:

$$\begin{array}{l} \text{Minimize}_{\{w,\ \xi\}} \frac{1}{2}||w||^2 + \sum_{i=1}^{N} C_i \xi_i \\ \text{subject to } \begin{cases} \langle\ w, \Phi(x_i) - \Phi(x_{i+1})\ \rangle \geq 1 + \xi_i \ \ (i = 1 \ldots N-1) \\ \xi_i \geq 0 \ \ (i = 1 \ldots N-1) \end{cases} \end{array} \tag{1}$$

The corresponding dual problem, quadratic in the Lagrangian multipliers α, can be solved easily. Finally, the rank surrogate $\mathcal{F}$ is given as

$$\mathcal{F}(x) = \sum_{i=1}^{N-1} \alpha_i (K(x_i, x) - K(x_{i+1}, x))$$

Like for any SVM-based approach, the main critical issue behind rank-based SVMs remains the choice of the kernel, that is known to be highly problem-dependent [15]. Furthermore, as pointed out in [4], the kernel used within an SVM-based surrogate should adapt to the optimization process: the optimal kernel is likely to change as search proceeds, exploring different regions of the search space. Some results related to kernel adaptation within SVM-based surrogate in CMA-ES, using fixed kernels, have been obtained by updating the surrogate model using Kendall tests on ranks (although approximate ranking could also have been used). The computational gains in terms of number of function evaluations do depend on the kernel, as was expected; the gains however are reported in [14] to rapidly decrease with the dimension d of the problem.

3 Rank-SVM CMA-ES

The main contribution of the paper is to integrate the rank-based surrogate approach first proposed by [14] within the CMA-ES framework, taking advantage of the Covariance-Matrix Adaptation scheme to adaptively define the kernel of the rank-based surrogate.

3.1 From CMA-ES to Rank-SVM kernel

By construction, CMA-ES adapts the covariance matrix describing the local structure of the fitness landscape. After [6], CMA-ES proceeds by adapting the

[3] The so-called kernel trick supports the extension of the SVM approach from linear to non-linear functional spaces, by mapping instance space X onto some *feature space*. It only requires the scalar product in feature space to be computable on instance space X through a *kernel* function K: $K(x, x') =_{def} \langle\ \Phi(x), \Phi(x')\ \rangle$.

problem encoding, and performing a Cumulative Step-size Adaptation algorithm in the transformed space. The change of coordinates, defined from the current covariance matrix C and the current mean value m, reads:

$$x'_j = C^{-1/2}(x_j - m), \tag{2}$$

Notably, the CMA information was directly used in [11] to building quadratic surrogate models; when training a quadratic surrogate model centered on x^*, the weight of each sample x was set to $\sqrt{(x-x^*)^T C^{-1}(x-x^*)}$.

In the case of a kernel-based surrogate model, it thus comes naturally to set the Radius-Based (RBF) kernel directly to the covariance matrix, with $\sigma > 0$:

$$K_C(x_i, x_j) = e^{-\frac{(x_i - x_j)^T C^{-1} (x_i - x_j)}{2\sigma^2}} \tag{3}$$

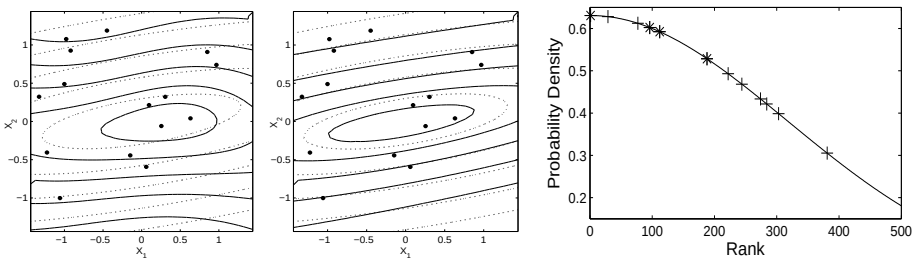
Fig. 1 (left) illustrates the potential gain of using such transformation for the simple case of the ellipsoid function, where the matrix C is exactly known. Interestingly, the change of coordinates is already computed within CMA-ES, therefore the transformation comes at almost no additional cost. Kernel width σ is set to the average distance between training points in the experiments. Another possibility, left for further work, could be to tie σ to CMA step size.

3.2 Overview of ACM-ES

Having chosen its kernel after Eq. (3), the integration of a rank-SVM as surrogate model within CMA-ES raises three main issues: i/ how to train the surrogate model, i.e. how to select the current training sample in the set of all points evaluated with the true objective function; ii/ how to use the model within CMA-ES, without perturbing the delicate adaptive mechanism thereof; and iii/ how to select the new points which will be evaluated with the true objective function.

Regarding the first issue, i.e. the selection of the training sample, several requirements have been identified. Firstly, the number $N_{training}$ of training samples must increase with the dimension d of the search space. Using statistical learning arguments, $N_{training}$ should be of the order of the VC dimension of the model space. Note that after transformation (Eq. 2) the decision space is a variant of the sphere function, in the best case, or a noisy multimodal variant thereof in the worst case. A second requirement is that the training samples should not lie too far from the current mean m of the distribution used by CMA-ES to generate its offspring, since the transformation defined by the current covariance matrix only aims at the local structure of the fitness landscape around m. Finally, the analysis of preliminary experiments on the d-dimensional sphere function shows that $N_{training}$ should increase proportionally to $\sqrt{d}$; the proportionality constant however remains problem-dependent as will be seen in section 4.1. These $N_{training}$ selected points are the best points evaluated with the true fitness function so far.

The second issue regards how to use the rank-based surrogate within CMA-ES. Using the surrogate model *in lieu* of the true fitness is a risky option due



(a) Contour plots of ellipsoid function (dotted lines) and Rank-SVM surrogates (solid lines) obtained with isotropic RBF kernel (left) or exact transformed RBF kernel (right).

(b) $\lambda = 12$ offspring ('+' and '$*$') and $\lambda' = 4$ candidates for true evaluation ('$*$') selected among 500 pre-children.

Fig. 1. (a) The transformed RBF kernel is more appropriate than the isotropic one. (b) Selecting offspring from pre-children: mapping the ranks to a normal distribution.

to the lack of guarantees about errors in regions outside the training sample. A more conservative option thus is to use the surrogate model to pre-screen the offspring [12], generating many more *pre-children* than required, and keeping the best ones after the surrogate model. Such an approach however rapidly looses the offspring diversity, hindering the CMA-ES adaptive mechanism used to adapt the covariance matrix. Some tradeoff between the optimization of the objective and the adaptation of the covariance matrix must thus be found.

The proposed approach finally is a two-step process. In order to prevent premature convergence, and interfere as little as possible with CMA-ES cumulative step-size adaptation, a large number N_{test} of pre-children is drawn using the standard CMA Gaussian distribution; let them be noted $x_1, \ldots x_{N_{test}}$, assuming with no loss of generality them to be ranked after the surrogate model. The λ offspring are obtained by iteratively drawing a real number $a < N_{test}$ from distribution $\mathcal{N}(0, \sigma^2_{sel0})$ (where σ_{sel0} is a parameter of the algorithm), and retaining the pre-child with rank $\lfloor a \rfloor$. The same procedure is followed to select the points to be evaluated according to the true objective function, with the same rationale: on the one hand, one should select the best points according to the current surrogate model; on the other hand, some diversity must be preserved. Finally, i/ the point with top rank is selected and always evaluated (as in the approximate ranking approach [13]); ii/ other (λ'-1) points selected among the pre-children using a rank distribution $\mathcal{N}(0, \sigma^2_{sel1})$ are evaluated, using the same process as for the offspring selection albeit with a larger standard deviation ($\sigma_{sel1} > \sigma_{sel0}$). A typical distribution of the ranks of the λ offspring is depicted on Fig. 1 (right), legend +, for $N_{test} = 500$, $\lambda = 12$, and $\sigma^2_{sel0} = 0.4$, while points that will be evaluated with the true fitness are represented by $*$ ($\lambda' = 4$ and $\sigma^2_{sel1} = 0.8$).

In ACM-ES, a fixed number λ' of points is evaluated in each generation, thus bounding the complexity in terms of true fitness evaluation. The choice of the ratio λ/λ' thus controls the efficiency of the approach and the speedup w.r.t. the standard CMA-ES (where λ offspring are evaluated in each generation).

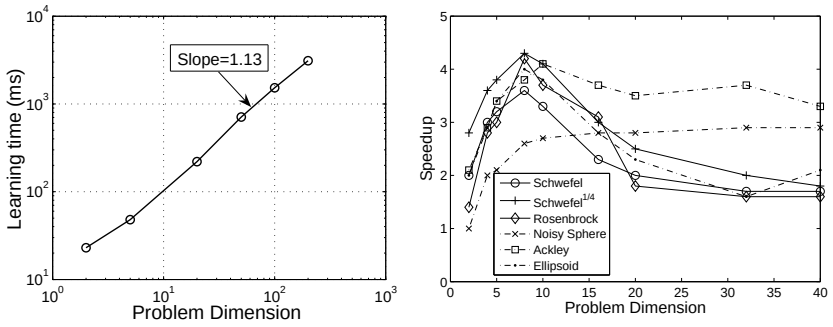


Fig. 2. Left: the cost of model learning/testing increases quasi-linearly with d. Right: the average speedup and speedup for all problems except Rastrigin.

4 Experimental validation

The experimental validation of ACM-ES investigates the performance of the approach comparatively to CMA-ES and nlmm-CMA, focussing on its scalability w.r.t. the problem dimension d, the robustness with respect to multi-modality, and with respect to the calibration of the surrogate training.

4.1 Experimental Settings

Seven uni- and multimodal benchmark functions have been considered (see Table 1, definitions in [11] and [3]), with dimension d ranging in $[2, 40]$ except for the Rastrigin function. Within ACM-ES, CMA-ES is used with its default parameters [8]. Reported results are based on 20 independent runs. The stopping criterion is reaching target value 10^{-10}, with a maximum of $1000d^2$ evaluations.

The rank-based surrogate was trained using $N_{training} = 30\sqrt{d}$ samples for all functions, except for Ellipsoid and Rosenbrock where it was set to $70\sqrt{d}$. The maximum number of iterations of the SVM learning algorithm was arbitrarily set to $50000\sqrt{d}$. The constraint weights C_i (Eq. 1) were set to $10^6(N_{training} - i)^{2.0}$, implying that the cost of constraint violation quadratically increases for top-ranked samples. For all functions except Rastrigin, $\lambda' = \frac{\lambda}{3}$, $\sigma^2_{sel0} = 0.4$, $\sigma^2_{sel1} = 2\sigma^2_{sel0} = 0.8$, $N_{test} = 500$. For Rastrigin function $\sigma^2_{sel0} = \sigma^2_{sel1} = 0.6$.

4.2 Results and Discussion

Firstly, experiments are conducted to estimate the empirical complexity of the surrogate training and using, using $100\sqrt{d}$ training points, stopping after $50000\sqrt{d}$ iterations and assessing the surrogated model on 500 test points. The empirical complexity with respect to dimension d (Fig. 2 (left) in log scale) is 1.13 (thus, slightly super-linear, contrasting with lmm-CMA complexity of $\mathcal{O}(d^6)$).

Table 1. Test functions, initialization intervals and initial std. dev. (from [11, 3]).

Noisy Sphere	$f_{NoisySphere}(\mathrm{x}) = (\sum_{i=1}^{d} x_i^2)\exp(\epsilon\mathcal{N}(0,1))$	$[-3,7]^d$	5
Ellipsoid	$f_{Elli}(\mathrm{x}) = \sum_{i=1}^{d} 10^{\frac{i-1}{d-1}} x_i^2$	$[1,5]^d$	2
Schwefel	$f_{Schwefel}(\mathrm{x}) = \sum_{i=1}^{d}(\sum_{j=1}^{i} x_j)^2$	$[-10,10]^d$	10
Schwefel$^{1/4}$	$f_{Schwefel^{1/4}}(\mathrm{x}) = (f_{Schwefel}(x))^{1/4}$	$[-10,10]^d$	10
Rosenbrock	$f_{Rosenbrock}(\mathrm{x}) = \sum_{i=1}^{d-1}\left(100.(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right)$	$[-5,5]^d$	0.5
Ackley	$f_{Ackley}(\mathrm{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) + \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i))$	$[1,30]^d$	14.5
Rastrigin	$f_{Rastrigin}(\mathrm{x}) = 10d + \sum_{i=1}^{d}(x_i^2 - 10.\cos(2\pi x_i))$	$[1,5]^d$	2

Secondly, the comparative validation of ACM-ES, nlmm-CMA and standard CMA-ES on all benchmark functions is reported in Table 2; lmm-CMA and nlmm-CMA results have been taken from original papers [11] and [3] when available; those of CMA-ES have been recomputed. Overall, ACM-ES outperforms lmm-CMA and nlmm-CMA algorithms on most problems, particularly so for problems with dimension $d > 4$. The invariance of ACM-ES w.r.t. monotonous transformations of the fitness is witnessed by its almost identical results on $f_{Schwefel}$ and $f_{Schwefel^{1/4}}$ functions, when the stopping criterion is adjusted accordingly (which is not the case for the results of Table 2). Likewise, the results on f_{Elli} confirm that ACM-ES also retains the good behavioral properties of CMA-ES with respect to the ill-conditioning of the fitness function. The speedup w.r.t CMA-ES is depicted on Fig. 2 (right) versus the problem dimension d. Interestingly, the speedup reaches its peak for d ranging in 8..10, then it decreases – except on the Noisy Sphere function. A possible explanation is that the noise level is comparatively less when the dimension increases (as in [3]), enabling the regularization involved in the model optimization to counteract the noise effects.

On the negative side, ACM-ES performs poorly on $f_{Rastrigin}$ function, and only solves it marginally for dimensions $d > 8$. This failure is attributed to the fact that ACM-ES does not handle well multi-modal diversity at the moment; it tends to accelerate the premature convergence to a local optimum, thus amplifying the weakness of CMA-ES on this benchmark problem: the best-performing versions of CMA-ES require an increasing population size [7]. Further work will consider the use of niching techniques to overcome this weakness.

5 Conclusion and Perspectives

The main contribution of the paper, ACM-ES, is a surrogate-based CMA-ES preserving invariance with respect to both monotonous transformations of the fitness function and orthogonal transformations of the search space. Comparison-based invariance is enforced by using rank-based Support Vector Machines to learn the surrogate model; coordinate invariance is enforced through using the covariance matrix adapted by CMA-ES as SVM kernel. Experimental validation confirms both invariance claims, and demonstrates the merits of the approach in terms of fitness evaluations and scalability w.r.t. the space dimension.

The main weakness of the approach is due to the failure of the surrogate model to account for multi-modal landscapes, as shown on the Rastrigin function; some improvements, e.g. related to niching, have been mentioned in the previous section and their validation is under way. Another issue regards the surrogate model hyper-parameters, which have been calibrated after preliminary experiments on the Sphere function conditionally to the carefully tuned hyper-parameters of CMA-ES [8]. A global approach, considering both sets of hyperparameters in an integrated way, would be appropriate. Another perspective, pointed out in [13], is to extend the approach to constrained optimization.

References

1. A. Auger, N. Hansen, J. Perez Zerpa, R. Ros, and M. Schoenauer. Experimental comparisons of derivative free optimization algorithms. In J. Vahrenhold, editor, *8th Intl Symp. Experimental Algorithms*, pages 3–15. LNCS 5526, Springer, 2009.
2. J.-F. M. Barthelemy and R. Haftka. Approximation concepts for optimial structural design – a review. *Structural Optimization*, 5:129–144, 1993.
3. Z. Bouzarkouna, A. Auger, and D. Ding. Investigating the local-meta-model CMA-ES for large population sizes. In C. Di Chio et al., editor, *Proc. EvoNUM'10*, pages 402–411. LNCS 6024, Springer-Verlag, 2010.
4. N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines*. Cambridge University Press, 2000.
5. S. Gelly, S. Ruette, and O. Teytaud. Comparison-based algorithms are robust and randomized algorithms anytime. *Evolutionary Computation*, 15(4):411–434, 2007.
6. N. Hansen. Adaptive encoding: How to render search coordinate system invariant. In G. Rudolph et al., editors, *PPSN X*, pages 205–214. LNCS 5199, Springer, 2008.
7. N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík. Comparing results of 31 algorithms from the BBOB-2009. In *GECCO Workshop Proc.* ACM Press, 2010.
8. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.
9. Y. Jin. A Comprehensive Survey of Fitness Approximation in Evolutionary Computation. *Soft Computing*, 9(1):3–12, 2005.
10. T. Joachims. A support vector method for multivariate performance measures. In L. D. Raedt and S. Wrobel, editors, *Proc. ICML'05*, volume 119 of *ACM International Conference Proceeding Series*, pages 377–384. ACM, 2005.
11. S. Kern, N. Hansen, and P. Koumoutsakos. Local meta-models for optimization using evolution strategies. In Th. Runarsson et al., editor, *PPSN IX*, pages 939–948. LNCS 4193, Springer Verlag, 2006.
12. K. Rasheed and H. Hirsh. Informed operators: Speeding up genetic-algorithm-based design optimization using reduced models. In D. Whitley et al., editor, *GECCO'2000*, pages 628–635. Morgan Kaufmann, 2000.
13. T. P. Runarsson. Constrained evolutionary optimization by approximate ranking and surrogate models. In Xin Yao et al., editor, *PPSN VIII*, pages 401–408. LNCS 3242, Springer Verlag, 2004.
14. T. P. Runarsson. Ordinal regression in evolutionary computation. In Th. Runarsson et al., editor, *PPSN IX*, pages 1048–1057. LNCS 4193, Springer Verlag, 2006.
15. J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

Table 2. Computational effort SP1 (i.e. average number of function evaluations of successful runs divided by proportion of successful runs), standard deviations and speedup performance (spu) of ACM-ES, (n)lmm-CMA-ES and CMA-ES. Results in the (n)lmm-CMA column are the best of those in [11] and [3] (marked with leading "n:" for the latter). Successful runs are those who reached the target fitness value of 10^{-10}. The proportion of successful runs is given in parentheses if less than 100%. ϵ is the noise level (when relevant).

Function	n	λ	λ'	ϵ	ACM-ES		spu	(n)lmm-CMA		spu	CMA-ES	
$f_{Schwefel}$	2	6	3		186±5		2.0	**81±5**		4.5	370±32	
	4	8	3		289±9		3.0	**145±7**		6.0	879±60	
	5	8	3		344±9		3.2				1112±72	
	8	10	3		558±18		3.6	**282±11**		7.1	2010±82	
	10	10	3		801±36		3.3				2667±87	
	16	11	3		2204±74		2.3	**626±17**		8.2	5156±161	
	20	12	4		3531±179		2.0				7042±172	
	32	14	4		8933±337		1.7				15072±377	
	40	15	5		13440±281		1.7				22400±289	
$f_{Schwefel^{1/4}}$	2	6	3		551±12		2.8	**n:413±25**		3.7	1527±76	
	4	8	3		**783±8**		3.6	n:971±36		2.9	2847±109	
	5	8	3		**914±15**		3.8	n:1302±31		2.7	3505±114	
	8	10	3		**1366±25**		4.3				5882±146	
	10	10	3		**1774±37**		4.1				7220±206	
	16	11	3		**4193±88**		3.0				12411±198	
	20	12	4		**6138±82**		2.5				15600±294	
	32	14	4		**14796±310**		2.0				29378±330	
	40	15	5		**22658±390**		1.8				41534±466	
$f_{Rosenbrock}$	2	6	3		511±84		1.4	**n:252±52**		2.8	700±194	
	4	8	3		775±108		2.8	**n:719±54**	(0.85)	3.0	2187±376	(0.85)
	5	8	3		**854±89**		3.0	n:1014±94	(0.90)	2.5	2526±308	(0.95)
	8	10	3		**1388±139**		4.2	2494±511	(0.90)	2.3	5769±547	(0.85)
	10	10	3		**2059±143**	(0.95)	3.7				7669±691	(0.90)
	16	11	3		**5255±560**		3.1	7299±1154		2.2	16317±1281	(0.90)
	20	12	4		**11793±574**	(0.75)	1.8				21794±1529	
	32	14	4		**32261±2165**	(0.8)	1.6				52671±5587	
	40	15	5		**49750±2412**	(0.9)	1.6				82043±3991	
$f_{NoisySphere}$	2	6	3	0.35	413±114		1.0	**n:109±12**		3.7	407±61	(0.95)
	4	8	3	0.25	428±46		2.0	**n:236±19**		3.6	844±141	
	5	8	3	0.22	480±66		2.1				1014±68	
	8	10	3	0.18	**630±76**		2.6	n:636±33		2.6	1663±140	
	10	10	3	0.15	**766±90**	(0.95)	2.7				2058±148	
	16	11	3	0.13	**1119±115**		2.8	n:2156±216		1.4	3120±168	
	20	12	4	0.11	**1361±212**		2.8				3777±127	
	32	14	4	0.09	**1997±247**		2.9				5767±162	
	40	15	5	0.08	**2409±120**		2.9				7023±173	
f_{Ackley}	2	6	3		352±39		2.1	**n:227±23**		3.2	735±55	
	4	8	3		**540±29**	(0.95)	2.9				1577±83	
	5	8	3		**566±33**		3.4	n:704±24	(0.90)	2.2	1904±122	(0.95)
	8	10	3		**800±22**	(0.95)	3.8				3066±114	
	10	10	3		**892±28**		4.1	n:2066±119	(0.95)	1.8	3641±154	
	16	11	3		**1530±39**		3.7				5672±151	
	20	12	4		**1884±50**		3.5	8150±196		0.8	6641±108	
	32	14	4		**2747±62**		3.7				10063±203	
	40	15	5		**3690±80**		3.3				12084±247	
f_{Elli}	2	6	3		**393±19**		2.0				774±73	
	4	8	3		**582±24**		2.9				1688±11	
	5	8	3		**683±33**		3.4				2342±162	
	8	10	3		**1142±53**		4.0				4542±155	
	10	10	3		**1628±95**		3.8				6211±264	
	16	11	3		**4706±148**		2.8				13177±341	
	20	12	4		**8250±393**		2.3				19060±501	
	32	14	4		**27281±753**		1.6				44562±530	
	40	15	5		**33602±548**		2.1				69642±644	
$f_{Rastrigin}$	2	50	25		1640±242	(0.6)	1.2	**n:528±48**	(0.95)	3.6	1970±418	(0.85)
	5	140	70		23293±1374	(0.3)	0.5	**n:4037±209**	(0.60)	3.0	12310±1098	(0.75)