Soft Comput
DOI 10.1007/s00500-012-0946-x

METHODOLOGIES AND APPLICATION

Computational results for an automatically tuned CMA-ES with increasing population size on the CEC'05 benchmark set

Tianjun Liao · Marco A. Montes de Oca · Thomas Stützle

Abstract In this article, we apply an automatic algorithm configuration tool to improve the performance of the CMA-ES algorithm with increasing population size (iCMA-ES), the best performing algorithm on the CEC'05 benchmark set for continuous function optimization. In particular, we consider a separation between tuning and test sets and, thus, tune iCMA-ES on a different set of functions than the ones of the CEC'05 benchmark set. Our experimental results show that the tuned iCMA-ES improves significantly over the default version of iCMA-ES. Furthermore, we provide some further analyses on the impact of the modified parameter settings on iCMA-ES performance and a comparison with recent results of algorithms that use CMA-ES as a subordinate local search.

Keywords Automatic algorithm configuration · CMA-ES · Continuous optimization

1 Introduction

The special session on real parameter optimization of the 2005 IEEE Congress on Evolutionary Computation (CEC'05) initiated a series of research efforts on benchmarking continuous optimizers and the development of new, improved continuous optimization algorithms. Two noteworthy results of this session are the establishment of a benchmark set of 25 hard benchmark functions and the establishment of CMA-ES with increasing population size (iCMA-ES) (Auger and Hansen 2005) as the state-of-the-art continuous optimizer at least for what concerns the field of nature-inspired computation in the widest sense.

Here, we explore whether we can improve iCMA-ES's performance on the CEC'05 benchmark set by further fine-tuning iCMA-ES using automatic algorithm configuration tools. In fact, iCMA-ES has a number of parameters and hidden constants in its code that make it a parameterized algorithm. Although its designers have spent a considerable effort in the design choices and certainly also in the definition of its parameters, over the past few years evidence has arisen that many algorithms' performance can be improved by considering automatic algorithm configuration and tuning tools (Adenso-Diaz and Laguna 2006; Birattari et al. 2002; Balaprakash et al. 2007; Bartz-Beielstein 2006; Hutter et al. 2007, 2009a, b; Nannen and Eiben 2007). It is therefore a natural question to ask whether and by how much the performance of iCMA-ES could be further improved by such tools. Note that the answer to this question has also implications on methodological aspects in algorithm development. When trying to improve over an algorithm such as iCMA-ES, the design and tuning process of a new algorithm often starts by some new idea that is then iteratively refined manually until better performance on the considered benchmark set is obtained. One such idea is to embed CMA-ES as a local search into other algorithms. In fact, various authors have followed this path and have reported positive results, claiming better performance than iCMA-ES on the CEC'05

Communicated by A.-A. Tantar.

T. Liao (✉) · T. Stützle
IRIDIA, CoDE, Université Libre de Bruxelles,
Brussels, Belgium
e-mail: tliao@ulb.ac.be

T. Stützle
e-mail: stuetzle@ulb.ac.be

M. A. M. de Oca
Department of Mathematical Sciences,
University of Delaware, Newark, DE, USA
e-mail: mmontes@math.udel.edu

Published online: 11 December 2012

benchmark set (Molina et al. 2010; Müller et al. 2009). As an alternative approach, it is reasonable to simply try to improve directly iCMA-ES by fine-tuning it further. Hence, the question arises as to how iCMA-ES would perform against these "improved" algorithms if additional effort is put directly in iCMA-ES instead of the design of "new" algorithms. In this article, we also try to shed some light on this issue.

The present paper is not the first to try to further tune CMA-ES using automatic algorithm configuration tools. CMA-ES was used in the paper by Hutter et al. (2009a) as a benchmark algorithm to be tuned for evaluating SPO^+, their improved variant of SPO (Bartz-Beielstein 2006). Following earlier work on SPO, they tuned CMA-ES only on individual functions; thus, in this sense "overtuning" CMA-ES on individual functions. (One has to remark, however, that the interest of Hutter et al. (2009a) was to evaluate SPO and the improved variant SPO^+ rather than proposing a new, generally improved parameter setting for CMA-ES.) Another attempt of tuning CMA-ES was made by Smit and Eiben in the paper "Beating the World Champion Evolutionary Algorithm via REVAC Tuning" (Smit and Eiben 2010). However, rather than the full version of CMA-ES, which is this "world champion evolutionary algorithm," in their study they use a reduced version that has limitations on rotated functions (for details, see Sect. 2). They reported significant improvements of their tuned algorithm over the default settings across the full range of functions of the CEC'05 benchmark set. From a tuning perspective, it should be mentioned that they tuned their algorithm on the whole set of the CEC'05 benchmark functions. For the tuning, they allowed the CMA-ES variant they used on each 10 dimensional function a maximum of 100 000 function evaluations; then they were running the tests with the tuned algorithm on the same functions for 1,000,000 function evaluations.

In this article, we tune iCMA-ES on a set of functions that has no overlap with the functions of the CEC'05 benchmark set. In this sense, we try to avoid a bias of the results obtained due to potentially overtuning (Birattari 2009) the algorithm on the same benchmark functions as those on which the algorithm is tested. As such, this gives a better assessment of the potential for what concerns the tuning of continuous optimizers as we have a separation between tuning and test set. Note that such a separation is standard when studying tuning algorithms for combinatorial problems (Birattari 2009; Birattari et al. 2002; Hutter et al. 2009b). This separation of tuning and test sets for continuous functions is also different from our own previous applications, where we have tuned algorithms on small dimensional functions and later tested them on (much) larger dimensional variants of the same functions (Liao et al. 2011c). In this latter case, the training and testing functions differ only in their dimensionality, which may potentially lead to some biases in the tuning. The differences in the applied approaches to tuning CMA-ES with respect to the separation of tuning and test sets of functions is summarized in Fig. 1.

As the tuning set, we consider small dimensional benchmark functions from the recent special issue of the Soft Computing journal (Herrera et al. 2010; Lozano et al. 2011) on large-scale function optimization. This SOCO benchmark set contains 19 functions whose dimension is freely choosable. Four of these functions are the same as in the CEC'05 benchmark set, so we removed them from the tuning set. As tuner, we apply the irace software (López-Ibáñez et al. 2011) to automatically tune seven parameters of iCMA-ES on the 10 dimensional SOCO benchmark functions (we refer to this tuned version of iCMA-ES as iCMA-ES-tsc). Then, we benchmark iCMA-ES-tsc on the whole CEC'05 benchmark function suite for 10, 30, and 50 dimensions. The experimental results show that iCMA-ES-tsc improves over the default parameter setting of iCMA-ES (called iCMA-ES-dp), and, maybe surprisingly, also is competitive or even improves over a version of iCMA-ES that we have tuned on the 10-dimensional CEC'05 benchmark set (we refer to this tuned version of iCMA-ES as iCMA-ES-tcec). We also compare iCMA-ES-tsc with MA-LSch-CMA (Molina et al. 2010) and PS-CMA-ES (Müller et al. 2009), two state-of-the-art algorithms based on CMA-ES on CEC'05 benchmark function suite. Finally, we explore different possible choices of the tuning setup and, in particular, the choice of different sets of tuning functions.

2 Parameterized iCMA-ES

CMA-ES (Hansen and Ostermeier 1996, 2001; Hansen et al. 2003) is a (μ, λ)-evolution strategy that samples new candidate solutions based on a multivariate normal distribution that is adapted at execution time. In particular, CMA-ES adapts the full covariance matrix of a normal search distribution. It is shown to result in a search that is invariant against linear transformations of the search space (rotational invariance), which makes it particularly suited for rotated functions. Sep-CMA-ES (Ros and Hansen 2008; Ros 2009) is a modification of CMA-ES with lower time complexity that instead of the full covariance matrix uses a diagonal matrix (that is, the covariances are assumed to be zero); in a sense, in Sep-CMA-ES the step size for each variable is adapted independently of the other variables. Sep-CMA-ES is also the variant that was used in the paper by Smit and Eiben (2010), which was mentioned in the introduction. iCMA-ES (Auger and Hansen 2005) is a variant of the CMA-ES algorithm that uses a restart schema coupled with an increasing population size. An outline of

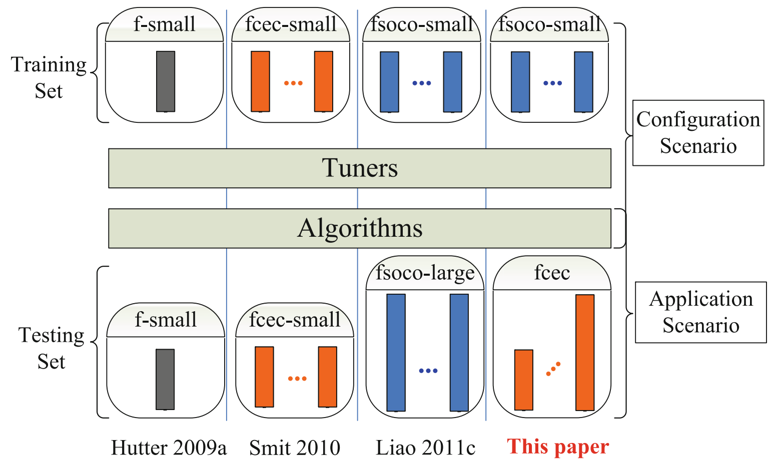

Fig. 1 Summary of the methodological approach to tuning CMA-ES over few recent articles (Hutter et al. 2009a; Smit and Eiben 2010; Liao et al. 2011c). The approaches differ in the usage of a single versus multiple functions and the degree of separation between tuning and test set

iCMA-ES is given in Algorithm 1. The core steps of iCMA-ES are the generation of new solutions for possible inclusion in the population, the selection of the best candidate solutions of the generated population, and the adaption of the step size and the covariance matrix. If the inner stopping criterion of CMA-ES triggers, then iCMA-ES restarts the CMA-ES algorithm with an increased population size. The execution of iCMA-ES terminates once the termination criterion such as a maximum number of function evaluations is met. For a detailed explanation of the optimization principles of CMA-ES we refer to Hansen (2010); Auger and Hansen (2005).

Algorithm 1 Outline of iCMA-ES

Input: μ, λ, $\sigma^{(0)}$, d, *stopTolFunHist*, *stopTolFun* and *stopTolX*.
  while termination criterion is not satisfied **do**
    Uniformly sample the initial point /* initialization */
    while stopping criterion for restart is not satisfied **do**
      /* CMA-ES iterations */
      Sample new population (λ, $\sigma^{(0)}$)
      Selection (μ)
      Adapt step size and covariance matrix
    end while (*stopTolFunHist*, *stopTolFun* and *stopTolX*)
    Increase population size (λ, d)
  end while

The default settings of iCMA-ES are as follows: The initial population size is $\lambda = 4 + \lfloor 3\ln(D) \rfloor$, where D is the number of dimensions of the function to be optimized. The number of selected search points in the parent population is $\mu = \lfloor 0.5\lambda \rfloor$. The initial step-size is $\sigma^{(0)} = 0.5(B - A)$, where $[A, B]^D$ is the initial search interval. At each restart, the population size is multiplied by a factor of two. Restarts occur if the stopping criterion is met. The three parameters *stopTolFunHist*, *stopTolFun* and *stopTolX* of the stopping criterion refer to the range of the improvement of the best objective function values in the last $10 + \lceil 30D/\lambda \rceil$ generations, all function values of the recent generation, and the standard deviation of the normal distribution in all coordinates, respectively. These internal parameter settings of iCMA-ES are set, as far as we are aware, by the experience of the developers of iCMA-ES.

For tuning iCMA-ES, we considered seven parameters related to the above-mentioned default settings. The parameters are given in Table 1. The first four parameters are actually used in a formula to compute some internal parameters of iCMA-ES and the remaining three are used to define the termination of CMA-ES. Note that if a run of iCMA-ES is terminated, CMA-ES is restarted with an increased population size λ. For the increase of the population size, we here introduce a parameter d we call IPOP factor. The first five columns of Table 1 give the parameters we use, the formula where they are used, their default values, and the range that we considered for tuning. The remaining two columns are explained later.

3 Experimental setup and tuning

We used the C version of iCMA-ES (last modification date 10/16/10) from Hansen's webpage http://www.lri.fr/~hansen/cmaesintro.html. We modified the code to handle bound constraints by clamping the variable values outside the bounds on the nearest bound value. (the issues about the effects of enforcing and ignoring bound constraints have been addressed by Liao et al. (2011a). Our test-suite consists of 25 CEC'05 benchmark functions (functions labeled as f_{cec*}) of dimensions $n \in \{10, 30, 50\}$. The training instances of iCMA-ES-tsc and iCMA-ES-tcec involve the 10-dimensional SOCO and CEC'05 benchmark functions, respectively. The SOCO and CEC'05 benchmark sets have four same functions (identical except for the shift vectors for moving the known optimum solution) and therefore we

Table 1 Parameters that have been considered for tuning

Parameters	Formulas	Factor	Default values	Range	Tuned configurations	
					f_{cec^*}	f_{soco^*}
Pop size (λ)	$4 + \lfloor a \ln(D) \rfloor$	a	3	[1, 10]	7.315	9.600
Parent size (μ)	$\lfloor \lambda / b \rfloor$	b	2	[1, 5]	3.776	1.452
Init step size ($\sigma^{(0)}$)	$c(B - A)$	c	0.5	(0, 1)	0.8297	0.6034
IPOP factor *(d)*	d	d	2	[1, 4]	2.030	3.292
stopTolFun	10^e	e	−12	[−20, −6]	−8.104	−8.854
stopTolFunHist	10^f	f	−20	[−20, −6]	−6.688	−9.683
stopTolX	10^g	g	−12	[−20, −6]	−13.85	−12.55

Given are the default values of the parameters and the continuous range we considered for tuning. The last two columns give for each set of tuning instances the found algorithm configurations

have removed these four functions from the SOCO benchmark set that we used as training set for tuning. In particular, we eliminated the four SOCO functions $f_{soco1}, f_{soco3}, f_{soco4}$ and f_{soco8}, which are the same as the CEC'05 functions $f_{cec1}, f_{cec6}, f_{cec9}$ and f_{cec2}, respectively.

The CEC'05 and SOCO benchmark functions are listed in Table 2. The two benchmark sets have common characteristics such as unimodality, multi-modality, and separability. Several of the functions in the two benchmark sets are also defined as compositions of other two functions; we refer to these also as hybrid functions in what follows: a major difference between the two benchmark sets is that in the CEC'05 benchmark 16 of the 25 functions are rotated functions, while all SOCO benchmark functions are unrotated.[1] For a more detailed explanation of the respective benchmark sets we refer to their original description (Suganthan et al. 2005; Herrera et al. 2010); for a more recent intent to develop specific, more low-level function features for their classification, we refer to Mersmann et al. (2011). We followed the protocol described in Suganthan et al. (2005) for the CEC'05 test-suite, that is, the maximum number of function evaluations was $10{,}000 \times D$ where $D \in \{10, 30, 50\}$ is the dimensionality of a function when using them as test set (or as training set in the case of iCMA-ES-tcec). The investigated algorithms were run 25 times on each function. We report error values defined as $f(x) - f(x^*)$, where x is a candidate solution and x^* is the optimal solution. Error values lower than 10^{-8} are clamped to 10^{-8}, which is the zero threshold defined in the CEC'05 protocol (Suganthan et al. 2005). Our analysis considers the median errors, mean errors and the solution quality distribution for each function.

For tuning the parameters of iCMA-ES, we employ Iterated F-Race (Birattari et al. 2010), a racing algorithm for algorithm configuration that is included in the irace package (López-Ibáñez et al. 2011). Iterated F-Race is an algorithm that repeatedly applies F-Race (Birattari et al. 2002) to a set of candidate configurations that are generated via a sampling mechanism that intensifies the search around the best found configurations. The generated candidate configurations then perform a "race". At each step of the race, each surviving candidate configuration is run on one benchmark function of the training set. Poor performing candidate configurations are eliminated from the race based on the result of statistical tests. To this aim, the results of each surviving configuration on the same training problem is ranked. Note that this ranking corresponds to blocking in statistical tests since ranks are determined on a same training problem. In fact, ranking is useful in the context of continuous function optimization to account for the different ranges of the values of the benchmark functions. Without ranking, few functions with large values would dominate the evaluation of the algorithm performance. Based on the obtained ranks, the Friedman test checks whether sufficient statistical evidence is gathered that indicates that some configurations behave differently from the rest. If the null hypothesis of the F-test is rejected, Friedman post-tests are used to eliminate the statistically worse performing candidates.

The performance measure is the fitness error value of each instance. In the automatic parameter tuning process, the maximum budget is set to 5,000 runs of iCMA-ES. The setting of Iterated F-Race we used is the default (López-Ibáñez et al. 2011). The input to Iterated F-Race are the ranges for each parameter, which are given in Table 1, and a set of training instances. When using the SOCO benchmark set of tuning, the 10-dimensional versions of $f_{soco1} - f_{soco19}$ (except $f_{soco1}, f_{soco3}, f_{soco4}$ and f_{soco8}) were sampled as training instances in a random order and the number of

[1] Recall that "rotational invariance" is an important feature of iCMA-ES. This feature of iCMA-ES means that its performance is not negatively affected by a rotation of a function with respect to the coordinate system. From the perspective of parameter tuning, this is an important property since it implies that we should be able to tune iCMA-ES on unrotated functions of the SOCO benchmark set. In fact, in the experimental part we consider as a control experiment also tuning iCMA-ES directly on the CEC'05 benchmark set with the rotated functions.

Table 2 Benchmark functions

ID	Name/description	Range $[X_{\min}, X_{\max}]^D$	Uni/multi-modal	Separable	Rotated
f_{cec1}	Shift.Sphere	$[-100, 100]^D$	U	Y	N
f_{cec2}	Shift.Schwefel 1.2	$[-100, 100]^D$	U	N	N
f_{cec3}	Shift.Ro.Elliptic	$[-100, 100]^D$	U	N	Y
f_{cec4}	Shift.Schwefel 1.2 Noise	$[-100, 100]^D$	U	N	N
f_{cec5}	Schwefel 2.6 Opt on Bound	$[-100, 100]^D$	U	N	N
f_{cec6}	Shift.Rosenbrock	$[-100, 100]^D$	M	N	N
f_{cec7}	Shift.Ro.Griewank No Bound	$[0, 600]^{D,\dagger}$	M	N	Y
f_{cec8}	Shift.Ro.Ackley Opt on Bound	$[-32, 32]^D$	M	N	Y
f_{cec9}	Shift.Rastrigin	$[-5, 5]^D$	M	Y	N
f_{cec10}	Shift.Ro.Rastrigin	$[-5, 5]^D$	M	N	Y
f_{cec11}	Shift.Ro.Weierstrass	$[-0.5, 0.5]^D$	M	N	Y
f_{cec12}	Schwefel 2.13	$[-\pi, \pi]^D$	M	N	N
f_{cec13}	Griewank plus Rosenbrock	$[-3, 1]^D$	M	N	N
f_{cec14}	Shift.Ro.Exp.Scaffer	$[-100, 100]^D$	M	N	Y
f_{cec15}	Hybrid Composition	$[-5, 5]^D$	M	N	N
f_{cec16}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec17}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec18}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec19}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec20}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec21}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec22}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec23}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec24}	Ro. Hybrid Composition	$[-5, 5]^D$	M	N	Y
f_{cec25}	Ro. Hybrid Composition	$[2, 5]^{D,\dagger}$	M	N	Y
f_{soco1}	Shift.Sphere	$[-100, 100]^D$	U	Y	N
f_{soco2}	Shift.Schwefel 2.21	$[-100, 100]^D$	U	N	N
f_{soco3}	Shift.Rosenbrock	$[-100, 100]^D$	M	N	N
f_{soco4}	Shift.Rastrigin	$[-5, 5]^D$	M	Y	N
f_{soco5}	Shift.Griewank	$[-600, 600]^D$	M	N	N
f_{soco6}	Shift.Ackley	$[-32, 32]^D$	M	Y	N
f_{soco7}	Shift.Schwefel 2.22	$[-10, 10]^D$	U	Y	N
f_{soco8}	Shift.Schwefel 1.2	$[-65.536, 65.536]^D$	U	N	N
f_{soco9}	Shift.Extended f_{10}	$[-100, 100]^D$	U	N	N
f_{soco10}	Shift.Bohachevsky	$[-15, 15]^D$	U	N	N
f_{soco11}	Shift.Schaffer	$[-100, 100]^D$	U	N	N
f_{soco12}	$f_{soco9} \oplus_{0.25} f_{soco1}$	$[-100, 100]^D$	M	N	N
f_{soco13}	$f_{soco9} \oplus_{0.25} f_{soco3}$	$[-100, 100]^D$	M	N	N
f_{soco14}	$f_{soco9} \oplus_{0.25} f_{soco4}$	$[-5, 5]^D$	M	N	N
f_{soco15}	$f_{soco10} \oplus_{0.25} f_{soco7}$	$[-10, 10]^D$	M	N	N
f_{soco16}	$f_{soco9} \oplus_{0.5} f_{soco1}$	$[-100, 100]^D$	M	N	N
f_{soco17}	$f_{soco9} \oplus_{0.75} f_{soco3}$	$[-100, 100]^D$	M	N	N
f_{soco18}	$f_{soco9} \oplus_{0.75} f_{soco4}$	$[-5, 5]^D$	M	N	N
f_{soco19}	$f_{soco10} \oplus_{0.75} f_{soco7}$	$[-10, 10]^D$	M	N	N

† Denotes initialization range without bound constraints. Its global optimum is outside of initialization range

function evaluations of each run is set equal to $5{,}000 \times D$ ($D = 10$). When using the CEC'05 benchmark set of tuning in some control experiments, the 10-dimensional variants of $f_{cec1} - f_{cec25}$ were sampled as training instances in a random order and the number of function evaluations of each run is equal to

Table 3 Overview of the abbreviations used in the article

Iterated F-Race	An algorithm for algorithm configuration that is included in the irace package (Birattari et al. 2010; López-Ibáñez et al. 2011)
CMA-ES	Covariance matrix adaptation evolution strategy
iCMA-ES	CMA-ES with increasing population size
iCMA-ES-dp	iCMA-ES with default parameter setting
iCMA-ES-tcec	iCMA-ES with parameters tuned on CEC'05 functions $f_{cec1} - f_{cec25}$
iCMA-ES-tsc	iCMA-ES with parameters tuned on SOCO functions $f_{soco1} - f_{soco19}$ (except $f_{soco1}, f_{soco3}, f_{soco4}$ and f_{soco8})
iCMA-ES-$\oplus$	iCMA-ES with parameters tuned on all hybrid functions of SOCO
iCMA-ES-uni	iCMA-ES with parameters tuned on uni-modal functions of SOCO except f_{soco1}, f_{soco8}
iCMA-ES-multi	iCMA-ES with parameters tuned on all non-hybrid multi modal functions of SOCO except f_{soco3}, f_{soco4}
iCMA-ES-uni+	Adding uni-modal functions f_{soco1} and f_{soco8} to respective training set
iCMA-ES-multi+	Adding multi modal functions f_{soco3} and f_{soco4} to respective training set
Sep-CMA-ES	A modification of CMA-ES that uses a diagonal matrix instead of the full covariance matrix (Ros and Hansen 2008)
Sep-iCMA-ES	Sep-CMA-ES with increasing population size (Ros 2009; smit and Eiben 2010)
Sep-iCMA-ES-tsc	Results of a tuned Sep-iCMA-ES from (Liao et al. 2011c)
iCMA-ES-05	Results of the Matlab version of iCMA-ES with a sophisticated bound handling mechanism from the CEC'05 special session (Auger and Hansen 2005)
MA-LSch-CMA	A memetic algorithm integrating CMA-ES as a local search algorithm (Molina et al. 2010)
PS-CMA-ES	A particle swarm optimization algorithm integrating CMA-ES as a local search algorithm (Müller et al. 2009)
CEC'05	Special session on real parameter optimization of the 2005 IEEE Congress on Evolutionary Computation (Suganthan et al. 2005)
SOCO	Benchmark set of a special issue of the Soft Computing journal on the scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems (Herrera et al. 2010)

$10{,}000 \times D$ ($D = 10$). The differences of the lengths of iCMA-ES runs on the SOCO and CEC'05 benchmark sets are due to the different termination criteria used in the definition of these benchmark sets. The default and tuned settings of iCMA-ES' parameters are presented in Table 1.

Comparing the tuned parameter settings to the default settings, maybe the most noteworthy difference is that the tuned settings imply a more explorative search behavior. This is true least for the initial phases of the iCMA-ES search at the algorithm start and after each restart. This more explorative search behavior is due to the larger initial population size (through parameter a), a larger initial step size and a larger factor for the increase of the population size at restarts (at least for configuration iCMA-ES-tsc, which, as we will see later, is the best performing one). The performance of iCMA-ES-dp and iCMA-ES-tsc will be compared in Sect. 4.1.

The significance of the differences of the algorithmic variants is assessed using statistical tests in two ways. First, on an instance level, we use a two-sided Wilcoxon signed-rank test at the 0.05 α-level to check whether the performance of two algorithms is statistically significantly different. Recall that each algorithm is run 25 independent times on each benchmark function. Second, across all benchmark functions, we apply a two-sided Wilcoxon matched-pairs signed-rank test at the 0.05 α-level to check whether the differences in the mean or median results obtained by two algorithms on each of the 25 CEC'05 benchmark functions is statistically significant.

In Table 3, we give an overview of the abbreviations that are used the experimental analysis section.

4 Experimental study

In this section, we compare the performance of iCMA-ES-tsc to the default paramter settings, to iCMA-ES-dp, and to other algorithms that use CMA-ES as a local search operator and that have been proposed with the aim of improving over iCMA-ES.

4.1 iCMA-ES-tsc versus iCMA-ES-dp

First, we focus on the improvement iCMA-ES-tsc obtains over iCMA-ES-dp. Table 4 shows the performance of iCMA-ES-dp (default parameters) and iCMA-ES-tsc (tuned on SOCO benchmark functions) on the CEC'05 benchmark function suite. Considering the differences on individual functions, we first observe that iCMA-ES-dp and iCMA-ES-tsc reach on surprisingly many functions similar performance at least from the perspective of the applied Wilcoxon test: on 19, 17, and 14 functions for 10, 30, and 50 dimensions, respectively, no statistically significant differences could be observed. On the one hand, this may be due to the relatively small number of 25 independent runs of the algorithms; on the other hand, this is also caused by the fact that several benchmark functions are very easy to solve and therefore introduce floor effects. For example, functions $f_{cec1}, f_{cec2}, f_{cec3}, f_{cec6}$, and f_{cec7} are solved by iCMA-ES-dp and iCMA-ES-tsc in all runs and in all dimensions to the zero threshold. In the other functions, iCMA-ES-tsc performs better than iCMA-ES-dp except in 2, 1, and 1 cases for dimensions 10, 30, and 50, respectively, indicating superior performance of iCMA-ES-tsc over iCMA-ES-dp.

When comparing the means and medians obtained across each of the benchmark functions, we can observe that iCMA-ES-tsc reaches statistically better results on dimension 50 according to the Wilcoxon test, while on dimensions 10 and 30 the observed differences are not statistically significant. However, on a function-by-function basis iCMA-ES-tsc is statistically better than iCMA-ES-dp on more functions than vice-versa.

On few functions, the differences in the solution qualities are very strong. As an example, consider function f_{cec4}, where iCMA-ES-dp stagnated at very high mean error values of 6.58E+02 and 1.43E+04 for dimensions 30 and 50, respectively, while iCMA-ES-tsc reached in each trial a solution better than the zero threshold. On other functions of dimension 50, such as functions f_{cec12}, f_{cec16}, and f_{cec17}, iCMA-ES-tsc more than halved the error values that were reached by iCMA-ES-dp.

Figure 2 shows correlation plots where each point has as x and y coordinate the mean error obtained with iCMA-ES-tsc and iCMA-ES-dp on a same function. The plots of Fig. 2 show the mean errors for the 10, 30, and 50 dimensional problems, respectively. Clearly, on some functions iCMA-ES-tsc reaches results that are of much better solution quality than those of iCMA-ES-dp (indicated by the circles that are above the diagonal). This is the case especially on functions $f_{cec4}, f_{cec5}, f_{cec11}, f_{cec12}, f_{cec16}$ and f_{cec17}. f_{cec4} and f_{cec17} are the two noisy functions of the CEC'05 benchmark. The other four are multi-modal functions; among these, f_{cec5} has the optimum on the bounds. Next, we focus on these six functions.

Figure 3 shows the development of the mean error for iCMA-ES-dp and iCMA-ES-tsc over the number of function evaluations on functions $f_{cec4}, f_{cec5}, f_{cec11}, f_{cec12}, f_{cec16}$ and f_{cec17} of dimension 50. We observe that iCMA-ES-tsc and iCMA-ES-dp perform similar up to about 1.00E+04 function evaluations. As the number of function evaluation increases, the advantage of iCMA-ES-tsc over iCMA-ES-dp starts to become apparent. At the stopping criterion of 5.00E+05 function evaluations from the CEC'05 competition rules (indicated by the dotted, vertical lines in the plots), iCMA-ES-tsc shows generally much lower mean errors. Especially on f_{cec4} and f_{cec5}, iCMA-ES-tsc converges fast to the zero threshold after 1.00E+04 function evaluations. Looking at what happens beyond the termination criterion of 5.00E+06 function evaluations (right from the dotted vertical lines in the plots), we can see that iCMA-ES-dp catches up with the lower mean errors of iCMA-ES-tsc on functions f_{cec5}, f_{cec11}, and f_{cec16}. Hence, on these functions the tuned parameter settings appear to result in a faster convergence towards near-optimal solutions. On functions f_{cec4} and f_{cec12} the advantage of iCMA-ES-tsc with respect to the mean error remains substantial. These general conclusions are also backed up by a more detailed analysis of the algorithms' qualified run-length distributions (RLDs) (Hoos and Stützle 2004). Qualified RLDs give the distribution of the number of function evaluations to reach specific bounds on the errors. For details on the qualified RLDs, which are measured across 100 independent algorithm trials, we refer to this articles's supplementary information pages (Liao et al. 2011b).

Finally, we consider qualified RLDs for iCMA-ES-dp and iCMA-ES-tsc on functions $f_{cec1}, f_{cec2}, f_{cec3}, f_{cec6}$ and f_{cec7} on dimension 50. On these functions each trial of iCMA-ES-dp and iCMA-ES-tsc reaches the zero threshold within the termination criterion of the CEC'05 protocol. Figure 4 shows the qualified RLDs for reaching the zero threshold over 100 independent runs for iCMA-ES-dp and iCMA-ES-tsc on these five functions. We observe that on f_{cec1}, f_{cec2} and f_{cec3}, both iCMA-ES-dp and iCMA-ES-tsc converge very fast to the zero threshold in each trial without recurring to restarts. For these three, relatively easy functions, iCMA-ES-tsc converges slightly more slowly than iCMA-ES-dp mainly because of its larger initial population size. On f_{cec6} and f_{cec7}, in contrast, iCMA-ES-tsc reaches a 100 % success rate faster than iCMA-ES-dp, although there is no dominance relationship among the RLDs.

As said at the end of the previous section, the parameter settings of iCMA-ES-tsc imply a larger exploration at the beginning of the search. For example, for dimension 50, the population size is 15 for the default settings but 41 for the tuned settings, that is, almost three times larger.[2] The experimental results are somehow in accordance with this interpretation of a higher exploration. In fact, on the hard noisy functions, most multi-modal and especially the hybrid functions the larger exploration apparently leads to better final performance. However, the larger initial exploration also leads to a slightly slower convergence to the optimum on several, relatively easily solved unimodal functions such as f_{cec1}, f_{cec2} and f_{cec3}, as we have shown through qualified RTDs in Fig. 4.

[2] Note that a different interpretation of the population size is that not the parameter setting for factor a should be changed but possibly the scaling function with the problem dimension. In fact, the scaling by a logarithmic function may be too weak and other scaling laws may be examined leading to overall better behavior of iCMA-ES.

Table 4 Results of the comparison between iCMA-ES-dp and iCMA-ES-tsc over 25 independent runs for CEC'05 functions

f_{cec}	10 dimensions				30 dimensions				50 dimensions			
	iCMA-ES-dp		iCMA-ES-tsc		iCMA-ES-dp		iCMA-ES-tsc		iCMA-ES-dp		iCMA-ES-tsc	
	Mean and median		Mean and median		Mean and median		Mean and median		Mean and median		Mean and median	
f_1	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_2	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_3	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_4	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	6.58E+02	1.75E+00 >	1.00E−08	1.00E−08	1.43E+04	1.27E+04 >	1.00E−08	1.00E−08
f_5	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	7.41E−02	8.61E−08 >	1.00E−08	1.00E−08
f_6	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_7	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_8	2.00E+01	2.00E+01 ≈	2.02E+01	2.00E+01	2.04E+01	2.00E+01 <	2.08E+01	2.10E+01	2.09E+01	2.11E+01 ≈	2.10E+01	2.11E+01
f_9	1.59E−01	1.00E−08 ≈	4.81E−02	1.00E−08	1.87E+00	1.99E+00 ≈	1.99E+00	1.99E+00	4.36E+00	3.98E+00 ≈	4.18E+00	3.98E+00
f_{10}	3.18E−01	1.00E−08 ≈	3.73E−03	1.00E−08	1.44E+00	9.95E−01 ≈	1.59E+00	9.95E−01	2.89E+00	1.99E+00 ≈	2.71E+00	2.98E+00
f_{11}	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	7.17E−02	1.00E−08 >	5.09E−05	1.00E−08	9.94E−02	1.00E−08 ≈	6.03E−02	1.00E−08
f_{12}	4.07E+03	1.00E−08 >	1.00E−08	1.00E−08	1.19E+04	4.84E+03 >	4.22E+02	5.96E+01	4.25E+04	2.78E+04 >	4.69E+03	3.39E+03
f_{13}	6.49E−01	6.37E−01 ≈	7.14E−01	6.94E−01	2.63E+00	2.71E+00 ≈	2.53E+00	2.57E+00	4.44E+00	4.37E+00 <	4.70E+00	4.62E+00
f_{14}	1.96E+00	1.98E+00 ≈	2.03E+00	2.09E+00	1.26E+01	1.26E+01 >	1.10E+01	1.13E+01	2.28E+01	2.30E+01 >	2.09E+01	2.12E+01
f_{15}	2.15E+02	2.00E+02 <	3.32E+02	4.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02
f_{16}	9.04E+01	9.11E+01 >	8.86E+01	9.03E+01	1.48E+01	1.51E+01 >	1.11E+01	1.07E+01	1.10E+01	1.14E+01 >	5.34E+00	5.47E+00
f_{17}	1.17E+02	1.09E+02 >	9.34E+01	9.43E+01	2.52E+02	1.80E+02 >	2.08E+02	5.60E+01	1.91E+02	1.62E+02 >	6.36E+01	4.99E+01
f_{18}	3.16E+02	3.00E+02 <	3.60E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.13E+02	9.16E+02 >	9.13E+02	9.13E+02
f_{19}	3.20E+02	3.00E+02 ≈	3.20E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.13E+02	9.15E+02 >	9.13E+02	9.13E+02
f_{20}	3.20E+02	3.00E+02 ≈	3.40E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.15E+02	9.15E+02 >	9.13E+02	9.13E+02
f_{21}	5.00E+02	5.00E+02 ≈	5.00E+02	5.00E+02	5.00E+02	5.00E+02 ≈	5.00E+02	5.00E+02	6.64E+02	5.00E+02 ≈	7.05E+02	5.00E+02
f_{22}	7.28E+02	7.28E+02 ≈	7.28E+02	7.28E+02	8.10E+02	8.11E+02 ≈	8.17E+02	8.24E+02	8.19E+02	8.18E+02 ≈	8.19E+02	8.21E+02
f_{23}	5.86E+02	5.59E+02 ≈	5.59E+02	5.59E+02	5.34E+02	5.34E+02 ≈	5.34E+02	5.34E+02	6.97E+02	5.40E+02 ≈	7.30E+02	5.40E+02
f_{24}	2.33E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02
f_{25}	4.34E+02	4.04E+02 >	4.03E+02	4.03E+02	2.10E+02	2.10E+02 >	2.09E+02	2.09E+02	2.14E+02	2.14E+02 >	2.13E+02	2.13E+02
Mean	$f_1 - f_{25}(<, =, >)$: (6, 11, 8)				$f_1 - f_{25}(<, =, >)$: (4, 13, 8)				$f_1 - f_{25}(<, =, >)$: (4, 10, 11)†			
Median	$f_1 - f_{25}(<, =, >)$: (3, 19, 3)				$f_1 - f_{25}(<, =, >)$: (2, 16, 7)				$f_1 - f_{25}(<, =, >)$: (3, 12, 10)†			
By Func	$f_1 - f_{25}(<, \approx, >)$: (2, 19, 4)				$f_1 - f_{25}(<, \approx, >)$: (1, 17, 7)				$f_1 - f_{25}(<, \approx, >)$: (1, 14, 10)			
	10, 30 and 50 dimensional $f_1 - f_{25}(<, >)$:(5%, 28%)											

Symbols <, ≈, and > denote whether the performance of iCMA-ES-dp is statistically better, indifferent, or worse than that of iCMA-ES-tsc according to the two-sided Wilcoxon matched-pairs signed-rank test at the 0.05 α-level. The numbers in parenthesis represent the times of <, ≈, and >, respectively. The numbers in parenthesis for (<, =, >) represent the times we have <, =, and >, respectively, when iCMA-ES-tcec is compared with iCMA-ES-tsc based on the mean or median errors

† Denotes there is a significant difference over the distribution of mean or median errors between iCMA-ES-tcec with iCMA-ES-tsc by a two-sided Wilcoxon matched-pairs signed-ranks test at the 0.05 α-level

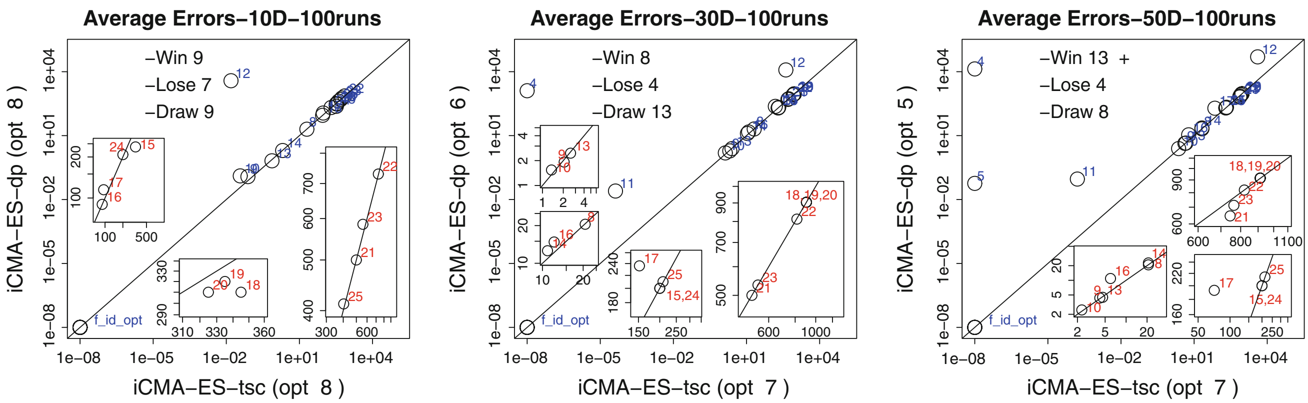

Fig. 2 Correlation plots of iCMA-ES-dp and iCMA-ES-tsc on dimensions 10, 30, and 50, respectively. Each point represents the mean error value obtained by either of the two algorithms. A point on the *upper triangle* delimited by the diagonal indicates better performance for the algorithm on the *x*-axis; a point on the *lower right triangle* indicates better performance for the algorithm on the *y*-axis. The *number labeled* beside some outstanding points represent the index of the corresponding function. The comparison is conducted based on mean error values and the comparison results of the algorithm on the *x*-axis are presented in form of -win, -draw, -lose, respectively. We marked with a + symbol those cases in which there is a statistically significant difference at the 0.05 α-level between the algorithms. The number of opt on the axes shows the number of means lower than the zero threshold by the corresponding algorithm

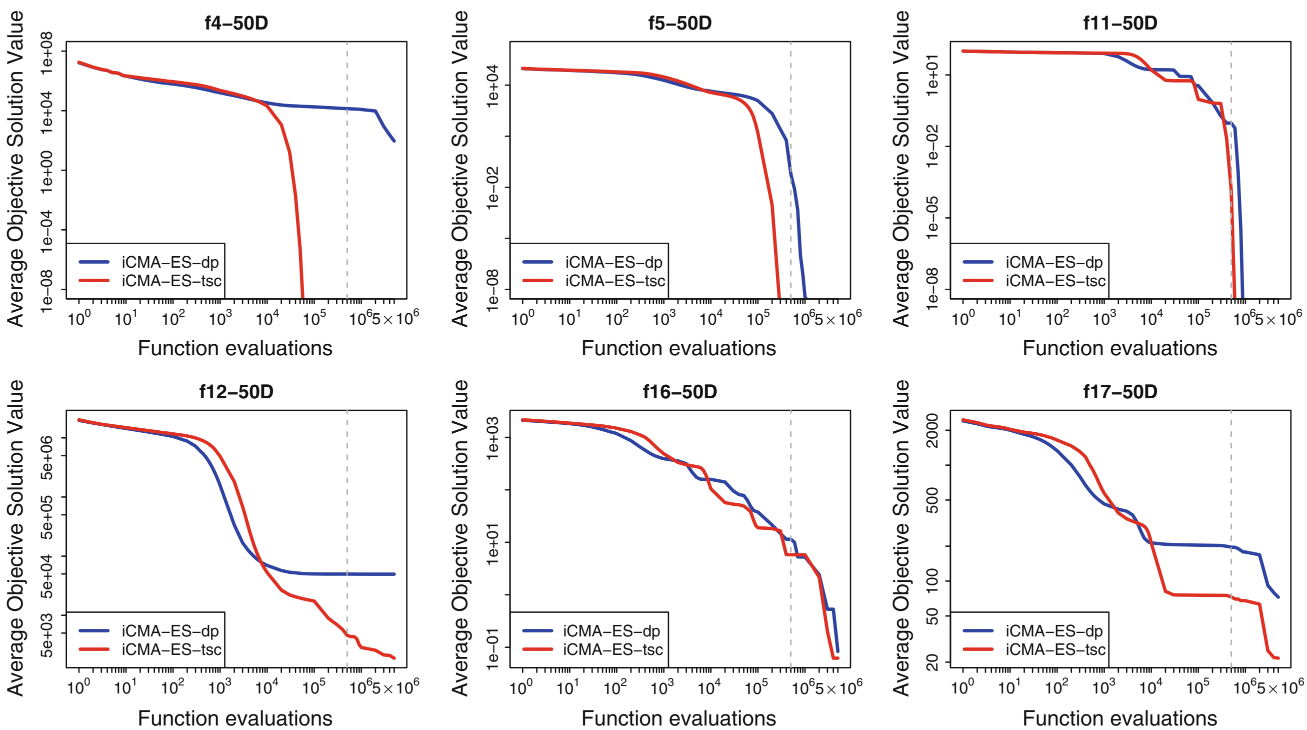

Fig. 3 The development of the mean error of the fitness values across 100 independent runs of iCMA-ES-dp and iCMA-ES-tsc over the number of function evaluations on functions $f_{cec4}, f_{cec5}, f_{cec11}, f_{cec12}, f_{cec16}$ and f_{cec17} of 50 dimensions. The *vertical, dotted line* in each plot indicates 5.00E+05 function evaluations, which is the termination criterion for the number of function evaluations in the CEC'05 protocol

4.2 iCMA-ES-tsc versus iCMA-ES-tcec

One may wonder whether tuning iCMA-ES on the CEC'05 benchmark suite directly incurs better final performance on this set of functions. To explore this question, we compare in Table 5 the performance of iCMA-ES-tcec and iCMA-ES-tsc on the CEC'05 benchmark set. iCMA-ES-tcec and iCMA-ES-tsc are mutually statistically better than each other on four functions of dimension 10, respectively. On the 10-dimensional functions, iCMA-ES-tsc is slightly worse than iCMA-ES-tcec with respect to the distribution of mean or median errors. This may be due to the fact that

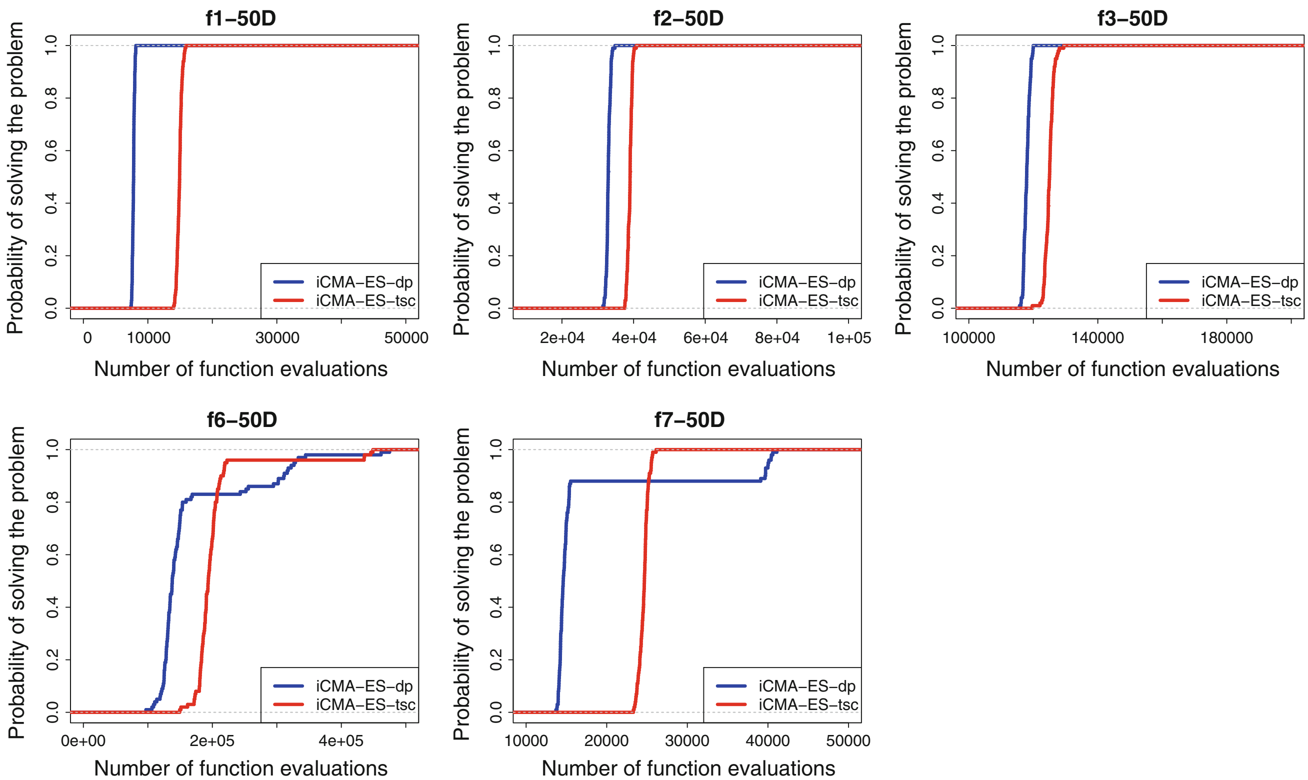

Fig. 4 The qualified run-length distributions (RLDs, for short) over 100 independent runs obtained by iCMA-ES-dp and iCMA-ES-tsc on the 50 dimensional versions of functions $f_{cec1}, f_{cec2}, f_{cec3}, f_{cec6}$ and f_{cec7}. The solution quality demanded is 1.00E−08 for each function

iCMA-ES-tcec is tuned on the 10-dimensional CEC'05 benchmark set, the same functions on which it is tested. Interestingly, this slight superiority of iCMA-ES-tcec on the 10-dimensional functions does not generalize to higher dimensions. As an example, consider functions f_{cec18}, f_{cec19}, f_{cec20} of dimension 10. On these, iCMA-ES-tcec obtains an error value of 3.00E+02 in all independent 25 runs which is the lowest value reported in the literature for these functions as far as we aware. However, on the 50-dimensional version of these functions, iCMA-ES-tcec is significantly worse than iCMA-ES-tsc. Moreover, considering the differences on all functions of dimension 50, iCMA-ES-tsc statistically significantly improves upon iCMA-ES-tcec on 12 functions while it performs statistically significantly worse than iCMA-ES-tcec on only three functions. Considering the distribution of the mean or median error values of the 50-dimensional functions, iCMA-ES-tsc statistically significantly improves upon iCMA-ES-tcec.

It should also be mentioned that tuning on the SOCO benchmark functions is much faster than on the CEC'05 benchmark set. In fact, the difference in computation time amounts to a factor of about 50. This difference is mainly due to the fact that 16 of the 25 CEC'05 functions of each dimension are rotated functions, which requires more costly computations in the evaluation such as multiplication operations on a rotated matrix.

4.3 Comparison to state-of-the-art methods that exploit CMA-ES

At least two recent, newly designed state-of-the-art algorithms exploit CMA-ES as an underlying local search method; these are a memetic algorithm with local search chains based on CMA-ES (MA-LSch-CMA) (Molina et al. 2010) and a hybridization of a PSO algorithm with CMA-ES (PS-CMA-ES) (Müller et al. 2009). We compare iCMA-ES-tsc to these following the experimental analysis used in Molina et al. (2010) and Müller et al. (2009), that is by (1) statistically analyzing for the distribution of the mean errors as in Molina et al. (2010) and, (2) ranking the mean errors as in Müller et al. (2009). As the results of MA-LSch-CMA and PS-CMA-ES we use those reported in Liao et al. (2011a). Note that the original results reported in Molina et al. (2010) and Müller et al. (2009) did not necessarily satisfy the bound constraints of the CEC'05 benchmark functions, which is corrected in the results reported in (Liao et al. 2011a). Table 6 shows that iCMA-ES-tsc performs statistically significantly better than MA-LSch-CMA in all dimensions; iCMA-ES-tsc performs statistically significantly better than PS-CMA-ES on the 50-dimensional functions and it reaches better performance than PS-CMA-ES on more functions for dimensions 10 and 30. Table 6 also shows that iCMA-ES-tsc obtains the best

Table 5 Comparison between iCMA-ES-tcec and iCMA-ES-tsc over 25 independent runs for CEC'05 functions

f_{cec}	10 dimensions				30 dimensions				50 dimensions			
	iCMA-ES-tcec		iCMA-ES-tsc		iCMA-ES-tcec		iCMA-ES-tsc		iCMA-ES-tcec		iCMA-ES-tsc	
	Mean and median		Mean and median		Mean and median		Mean and median		Mean and median		Mean and median	
f_1	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_2	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_3	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_4	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	2.85E+02	1.00E−08 >	1.00E−08	1.00E−08
f_5	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	2.69E−08	2.72E−08>	1.00E−08	1.00E−08	5.02E−08	4.85E−08 >	1.00E−08	1.00E−08
f_6	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	2.79E−02	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_7	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08
f_8	2.01E+01	2.00E+01 ≈	2.02E+01	2.00E+01	2.08E+01	2.09E+01 ≈	2.08E+01	2.10E+01	2.08E+01	2.11E+01 <	2.10E+01	2.11E+01
f_9	2.64E−01	1.00E−08 >	4.81E−02	1.00E−08	2.15E+00	1.99E+00 ≈	1.99E+00	1.99E+00	5.49E+00	5.97E+00 >	4.18E+00	3.98E+00
f_{10}	1.59E−01	1.00E−08 >	3.73E−03	1.00E−08	1.67E+00	1.99E+00 >	1.59E+00	9.95E−01	3.54E+00	3.98E+00 >	2.71E+00	2.98E+00
f_{11}	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	8.69E−02	1.00E−08 >	5.09E−05	1.00E−08	1.83E−01	1.00E−08 ≈	6.03E−02	1.00E−08
f_{12}	1.00E−08	1.00E−08 ≈	1.00E−08	1.00E−08	4.88E+02	1.92E+02 ≈	4.22E+02	5.96E+01	6.41E+03	4.59E+03 ≈	4.69E+03	3.39E+03
f_{13}	6.14E−01	6.46E−01 <	7.14E−01	6.94E−01	2.48E+00	2.50E+00 ≈	2.53E+00	2.57E+00	4.42E+00	4.47E+00 <	4.70E+00	4.62E+00
f_{14}	8.07E−01	6.66E−01 <	2.03E+00	2.09E+00	9.93E+00	9.94E+00 >	1.10E+01	1.13E+01	2.02E+01	1.98E+01 <	2.09E+01	2.12E+01
f_{15}	2.96E+02	3.00E+02 <	3.32E+02	4.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02
f_{16}	8.83E+01	8.94E+01 ≈	8.86E+01	9.03E+01	1.06E+01	1.04E+01 ≈	1.11E+01	1.07E+01	9.46E+00	9.09E+00 >	5.34E+00	5.47E+00
f_{17}	1.20E+02	1.17E+02 >	9.34E+01	9.43E+01	2.14E+02	1.52E+02 ≈	2.08E+02	5.60E+01	9.55E+01	8.06E+01 >	6.36E+01	4.99E+01
f_{18}	3.00E+02	3.00E+02 <	3.60E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.15E+02	9.16E+02 >	9.13E+02	9.13E+02
f_{19}	3.00E+02	3.00E+02 ≈	3.20E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.16E+02	9.16E+02 >	9.13E+02	9.13E+02
f_{20}	3.00E+02	3.00E+02 ≈	3.40E+02	3.00E+02	9.04E+02	9.04E+02 ≈	9.04E+02	9.04E+02	9.15E+02	9.14E+02 >	9.13E+02	9.13E+02
f_{21}	5.00E+02	5.00E+02 ≈	5.00E+02	5.00E+02	5.00E+02	5.00E+02 ≈	5.00E+02	5.00E+02	9.68E+02	1.01E+03 >	7.05E+02	5.00E+02
f_{22}	7.27E+02	7.26E+02 ≈	7.28E+02	7.28E+02	8.10E+02	8.07E+02 ≈	8.17E+02	8.24E+02	8.15E+02	8.14E+02 ≈	8.19E+02	8.21E+02
f_{23}	5.59E+02	5.59E+02 ≈	5.59E+02	5.59E+02	5.34E+02	5.34E+02 ≈	5.34E+02	5.34E+02	9.04E+02	1.02E+03 >	7.30E+02	5.40E+02
f_{24}	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02	2.00E+02	2.00E+02 ≈	2.00E+02	2.00E+02
f_{25}	4.04E+02	4.04E+02 >	4.03E+02	4.03E+02	2.09E+02	2.09E+02 >	2.09E+02	2.09E+02	2.13E+02	2.13E+02 >	2.13E+02	2.13E+02
Mean	$f_1 - f_{25}$ (<, =, >): (9, 12, 4)				$f_1 - f_{25}$(<, =, >): (4, 14, 7)				$f_1 - f_{25}$(<, =, >): (4, 8, 14)†			
Median	$f_1 - f_{25}$ (<, =, >): (5, 18, 2)				$f_1 - f_{25}$(<, =, >): (5, 16, 4)				$f_1 - f_{25}$(<, =, >): (3, 11, 11)†			
By Func	$f_1 - f_{25}$ (<, ≈, >): (4, 17, 4)				$f_1 - f_{25}$(<, ≈, >): (1, 20, 4)				$f_1 - f_{25}$(<, ≈, >): (3, 10, 12)			
	10, 30 and 50 dimensional $f_1 - f_{25}$ (<, >):(11%, 27%)											

Symbols <, ≈, and > denote whether the performance of iCMA-ES-tcec is statistically better, indifferent, or worse than that of iCMA-ES-tsc according to the two-sided Wilcoxon matched-pairs signed-rank test at the 0.05 α-level. The numbers in parenthesis represent the times of <, ≈, and >, respectively. The numbers in parenthesis for (<, =, >) represent the times we have <, = and >, respectively, when iCMA-ES-tcec is compared with iCMA-ES-tsc based on the mean or median errors

† denotes there is a significant difference over the distribution of mean or median errors between iCMA-ES-tcec with iCMA-ES-tsc by a two-sided Wilcoxon matched-pairs signed-ranks test at the 0,05 α-level

Table 6 The mean errors obtained by MA-LSch-CMA, PS-CMA-ES and iCMA-ES-tsc (MA, PS, iCMAESt for their abbreviations, respectively, in this table) over 25 independent runs for CEC'05 functions

f_{cec}	10 dimensions			30 dimensions			50 dimensions		
	Mean errors			Mean errors			Mean errors		
	MA	PS	iCMAESt	MA	PS	iCMAESt	MA	PS	iCMAESt
f_1	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08
f_2	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	1.00E−08	3.06E−02	7.36E−06	1.00E−08
f_3	1.00E−08	1.45E−01	1.00E−08	2.75E+04	2.96E+04	1.00E−08	3.21E+04	9.10E+04	1.00E−08
f_4	5.54E−03	1.00E−08	1.00E−08	3.02E+02	4.56E+03	1.00E−08	3.23E+03	2.17E+04	1.00E−08
f_5	6.75E−07	1.00E−08	1.00E−08	1.26E+03	2.52E+01	1.00E−08	2.69E+03	1.79E+03	1.00E−08
f_6	3.19E−01	1.00E−08	1.00E−08	1.12E+00	1.15E+01	1.00E−08	4.10E+00	2.91E+01	1.00E−08
f_7	1.43E−01	1.00E−08	1.00E−08	1.75E−02	1.00E−08	1.00E−08	5.40E−03	1.00E−08	1.00E−08
f_8	2.00E+01	2.00E+01	2.02E+01	2.00E+01	2.00E+01	2.08E+01	2.00E+01	2.00E+01	2.10E+01
f_9	1.00E−08	3.98E−02	4.81E−02	1.00E−08	8.76E−01	1.99E+00	1.00E−08	5.45E+00	4.18E+00
f_{10}	2.67E+00	1.00E−08	3.73E−03	2.25E+01	5.57E−01	1.59E+00	5.01E+01	5.33E+00	2.71E+00
f_{11}	2.43E+00	8.51E−01	1.00E−08	2.15E+01	7.10E+00	5.09E−05	4.13E+01	1.59E+01	6.03E−02
f_{12}	1.14E+02	1.10E+00	1.00E−08	1.67E+03	8.80E+02	4.22E+02	1.39E+04	6.90E+03	4.69E+03
f_{13}	5.45E−01	3.67E−01	7.14E−01	2.03E+00	2.05E+00	2.53E+00	3.15E+00	4.15E+00	4.70E+00
f_{14}	2.25E+00	3.40E+00	2.03E+00	1.25E+01	1.24E+01	1.10E+01	2.22E+01	2.15E+01	2.09E+01
f_{15}	2.24E+02	8.67E+01	3.32E+02	3.00E+02	1.37E+02	2.00E+02	3.72E+02	1.25E+02	2.00E+02
f_{16}	9.18E+01	9.28E+01	8.86E+01	1.26E+02	1.59E+01	1.11E+01	6.90E+01	1.62E+01	5.34E+00
f_{17}	1.01E+02	1.12E+02	9.34E+01	1.83E+02	9.15E+01	2.08E+02	1.47E+02	9.13E+01	6.36E+01
f_{18}	8.84E+02	3.60E+02	3.60E+02	8.98E+02	9.05E+02	9.04E+02	9.41E+02	8.70E+02	9.13E+02
f_{19}	8.78E+02	3.25E+02	3.20E+02	9.01E+02	8.85E+02	9.04E+02	9.38E+02	9.13E+02	9.13E+02
f_{20}	8.63E+02	3.43E+02	3.40E+02	8.96E+02	9.05E+02	9.04E+02	9.28E+02	9.09E+02	9.13E+02
f_{21}	7.94E+02	4.71E+02	5.00E+02	5.12E+02	5.00E+02	5.00E+02	5.00E+02	6.62E+02	7.05E+02
f_{22}	7.53E+02	7.46E+02	7.28E+02	8.80E+02	8.43E+02	8.17E+02	9.14E+02	8.63E+02	8.19E+02
f_{23}	8.88E+02	5.58E+02	5.59E+02	5.34E+02	5.34E+02	5.34E+02	5.39E+02	8.12E+02	7.30E+02
f_{24}	2.28E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02	2.00E+02
f_{25}	4.55E+02	4.00E+02	4.03E+02	2.14E+02	2.10E+02	2.09E+02	2.21E+02	2.14E+02	2.13E+02
V.S.	(3, 4, 18)†	(8, 8, 9)		(7, 4, 14)†	(7, 6, 12)		(5, 2, 18)†	(6, 4, 15)†	
Optima	4	7	**9**	3	3	**7**	2	2	**7**
Rank	2.52	1.78	**1.7**	2.26	1.98	**1.76**	2.42	2.02	**1.56**

The numbers in parenthesis represent the times of <, =, and >, respectively, when the corresponding algorithms are compared with iCMA-ES-tsc based on the mean errors. The number of means below the zero-threshold found by each algorithm (indicated by "Optima" and the average ranking of each algorithm are also given

† Denotes there is a significant difference over the distribution of mean errors between the corresponding algorithm with iCMA-ES-tsc by a two-sided Wilcoxon matched-pairs signed-ranks test at the 0.05 α-level.

Bold values indicate the algorithm with the best average rank and the largest number of optimal solutions

average ranking in all dimensions and most often the zero threshold in all dimensions. Clearly, it would be interesting to also automatically tune MA-LSch-CMA and PS-CMA-ES to exploit possibly more their potential. Nevertheless, this comparison indicates that a feasible way to go for improving the performance of iCMA-ES-dp is to further fine-tune iCMA-ES parameters (or maybe other design choices of iCMA-ES) instead of embedding CMA-ES into other algorithms. Recall that this also justifies the effort in tuning iCMA-ES because when designing new hybrid algorithms, often also a substantially large effort flows into the further, often manual fine-tuning of algorithm parameters and algorithm designs.

5 Additional experiments

5.1 Comparison with other results by iCMA-ES

We also compared iCMA-ES-tsc with Sep-iCMA-ES-tsc (Liao et al. 2011c), the algorithm used in the article by Smit and Eiben (2010), on the full CEC'05 benchmark set.

Figure 5 shows correlation plots that illustrate the relative performance for Sep-iCMA-ES-tsc and iCMA-ES-tsc on dimensions 10, 30, and 50, respectively. Each point represents the mean error value obtained by either of the two algorithms. These plots indicate superior performance of iCMA-ES-tsc over Sep-iCMA-ES-tsc, which is confirmed by the table of full results available at Liao et al. (2011b). We verified that iCMA-ES-tsc reaches statistically significantly better performance than Sep-iCMA-ES-tsc on the distribution of mean error values on all dimensions. This comparison confirms our expectation of iCMA-ES-tsc's superiority over Sep-iCMA-ES-tsc on the CEC'05 benchmark set, where 16 of 25 functions are rotated functions. The most significant example is f_{cec3}, a unimodal rotated high conditional function, where Sep-iCMA-ES-tsc stagnated at very high mean error values for all dimensions, while iCMA-ES-tsc reached in each trial the zero threshold. However, Sep-iCMA-ES-tsc obtains better performance than iCMA-ES-tsc on f_{cec10}, a rotated Rastrigin function, over all dimensions. This case gives an indication that we can only conclude that iCMA-ES's rotational invariance plays a pivotal role to handle most but not all rotated functions.

Next, we take the mean errors reported for iCMA-ES in the CEC'05 special session as a reference and refer to these results as iCMA-ES-05. Note that the results of iCMA-ES-05 were obtained with a different implementation (using the Matlab and not the C code we use) and with a much more sophisticated bound handling mechanism. We summarize the comparison with iCMA-ES-tsc in Table 7. iCMA-ES-tsc performs statistically significantly better than iCMA-ES-05 on dimension 30 and reaches on more functions statistically significantly better results (on a per function basis). This confirms the high performance of iCMA-ES-tsc.

5.2 Tuning setup

In this section, we examine different choices for the composition of the training set to obtain an indication which types of functions are important for high performance of the tuned iCMA-ES. We also explore alternative settings of the irace tool.

In what follows, we define training sets that are composed of different subsets of the SOCO benchmark functions and evaluate the tuned performance of iCMA-ES using the mean errors on the 50-dimensional CEC'05 benchmark functions. We summarize here our main findings and for detailed numerical results we refer to the supplementary page (Liao et al. 2011b).

The convention we use for labeling the training function sets is introduced first:

- $\oplus$ — all hybrid functions of SOCO
- uni — uni-modal functions of SOCO except f_{soco1}, f_{soco8}
- multi — all non-hybrid multi modal functions of SOCO except f_{soco3}, f_{soco4}
- $+$ — adds the uni-modal (f_{soco1}, f_{soco8}) or multi modal (f_{soco3}, f_{soco4}) functions to respective training

For example, iCMA-ES-$\oplus$ denotes iCMA-ES tuned using only the eight hybrid functions, iCMA-ES-uni,multi denotes iCMA-ES tuned with the seven uni-modal and (non-hybrid) multi-modal functions of SOCO, and

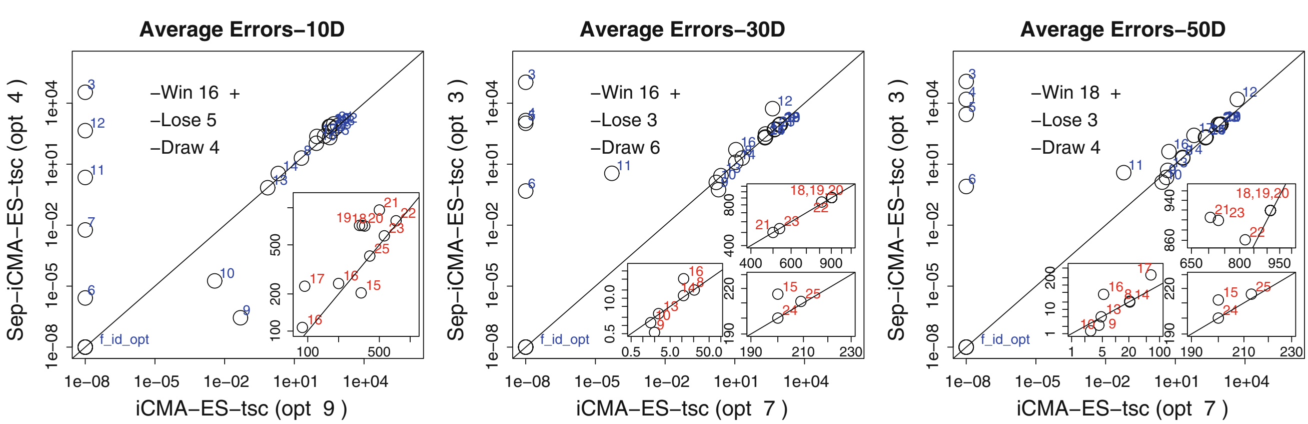

Fig. 5 Correlation *plots* of iCMA-ES-tsc and Sep-iCMA-ES-tsc on dimensions 10, 30, and 50, respectively. Each point represents the mean error value over 25 independent runs obtained by either of the two algorithms. A point on the upper triangle delimited by the diagonal indicates better performance for the algorithm on the *x*-axis; a point on the lower right triangle indicates better performance for the algorithm on the *y*-axis. The number labeled beside some outstanding points represent the index of the corresponding function. The comparison is conducted based on mean error values and the comparison results of the algorithm on the *x*-axis are presented in the form of -win, -draw, -lose, respectively, using iCMA-ES-tsc as the reference. We marked with a + symbol those cases in which there is a statistically significant difference at the 0.05 α-level between the algorithms. The number of opt on the axes shows the number of means that is lower than the zero threshold, obtained by the corresponding algorithm

Table 7 Summary of the comparison with iCMA-ES-tsc on 10, 30 and 50 dimensions with respect to mean error values: (better, equal, worse)

	iCMA-ES-05 vs.iCMA-ES-tsc	Sep-iCMA-ES-tsc vs.iCMA-ES-tsc
10 Dim	(6, 10, 9)	(5, 4, 16)†
30 Dim	(4, 11, 10)†	(3, 6, 16)†
50 Dim	(7, 6, 12)	(3, 4, 18)†

Error values lower than 10^{-8} are approximated to 10^{-8}

† Denotes there is a significant difference over the distribution of mean errors between the corresponding algorithm and iCMA-ES-tsc according to a two-sided Wilcoxon matched-pairs signed-rank test at the 0.05 α-level

Table 8 Given are for each algorithm the number of optima reached and the average rank on the CEC'05 benchmark problems of dimension 50

ΔR_α	Algs	OptNum	Rank	ΔR
19.75	iCMA-ES-tsc	7	2.9	0
	iCMA-ES-⊕	4	2.9	0
	iCMA-ES-uni	7	3.2	7.5
	iCMA-ES-dp	5	3.7	20.0
	iCMA-ES-uni,multi	6	3.8	22.5
	iCMA-ES-multi	4	4.5	40.0

We use the Friedman test at significance level $\alpha = 0.05$ is used. ΔR_α is the minimum significant difference between the ranks of algorithms. The numbers in the last column are the differences of the sum of ranks relative to the best algorithm; if a difference is larger than ΔR_α, it is statistically significant

iCMA-ES-uni+ denotes iCMA-ES tuned with all seven uni-modal functions.

In Table 8 we summarize the average ranking and the statistical analysis of several parameter settings of iCMA-ES that were obtained with the various training set compositions that we considered. In Table 9 are given the parameter setting obtained using these training set compositions; the parameter settings for iCMA-ES-tsc and iCMA-ES-dp are given in Table 1. The main conclusions we can obtain are the following:

1. The usage of the hybrid functions in the training set is a key to high tuned performance. iCMA-ES-⊕ obtains about the same performance as iCMA-ES-tsc and if the hybrid functions are not part of the training set, the performance of the tuned iCMA-ES degrades considerably. Interestingly, the configuration iCMA-ES-⊕ obtains on all 50-dimensional hybrid functions of CEC'05 significantly better results than iCMA-ES-tsc-uni,multi, indicating that there maybe some common aspects between the hybrid functions of the SOCO and the CEC'05 benchmark set.
2. The usage of the multi-modal functions only, that is, configuration iCMA-ES-multi, leads to significantly worse performance than iCMA-ES-⊕. One may object that the set multi contains only two training functions; however, adding the two multi-modal functions f_{soco3} and f_{soco4} to the training set does not lead to much improved performance (configuration iCMA-ES-multi+, see Liao et al. (2011b)).
3. Configuration iCMA-ES-uni leads to, at first sight, surprisingly high performance on the CEC'05 functions, and it has only a slightly worse mean rank than iCMA-ES-⊕. At a second glance, it is noteworthy that uni-modal functions such as those in the set "uni" can actually be quite difficult to optimize; for example, the default parameter setting of iCMA-ES has poor performance on uni-modal functions f_{cec4} and f_{cec5} (see Table 3). Configurations iCMA-ES-uni+, which uses also functions f_{soco1} and f_{soco8} in the training set, is, however, worse than iCMA-ES-uni (and significantly worse performing than iCMA-ES-⊕). This is possibly caused by floor effects obtained due to adding functions that are easily solved by iCMA-ES.

A common pattern among the best performing parameter settings, which are iCMA-ES-tsc, iCMA-ES-⊕, and iCMA-ES-uni, is that they tend to increase the exploration performed by iCMA-ES. In fact, the commonalities of these parameter settings are a higher population size, a (slightly) larger initial step size, and a faster increase of the population size upon a restart than the default parameter settings. Since these parameter settings improve performance, in particular, on the hardest benchmark problems, it may be that the default settings were possibly biased by experiments on too simple benchmark functions.

Considering the tuning setup, we also made tests (1) replacing the F-test with a t-test (that is, using the Student t-test for the race) and (2) increasing the tuning budget to 25000 runs. Similar to the results reported previously by iCMA-ES-tsc, the resulting configurations improved upon iCMA-ES-dp, being statistically significantly better than iCMA-ES-dp on the distribution of the mean or the median error values. These experiments also indicate that the observation of the superior performance of iCMA-ES-tsc over iCMA-ES-dp is relatively stable with respect to some (minor) changes in the tuning setup. The detailed data of these trials are available at Liao et al. (2011b).

6 Conclusions and future work

In this article, we tuned iCMA-ES to improve its performance on the CEC'05 benchmark set. We did so by using a

Table 9 Given are the tuned parameter settings for different subsets of the training functions taken from the SOCO benchmark set

Factor	iCMA-ES			
	−⊕	−uni,multi	−uni	−multi
a	8.349	5.977	8.419	3.235
b	1.647	3.749	2.324	4.537
c	0.5325	0.4666	0.5955	0.4582
d	3.809	2.571	2.499	1.618
e	−13.01	−10.87	−8.583	−8.178
f	−6.217	−8.703	−11.57	−14.83
g	−8.37	−17.46	−19.91	−19.61

separation between training and test set to avoid the bias of the results due to potentially overtuning the algorithm. Our experimental results showed that the tuned iCMA-ES improves significantly over the default parameter settings of iCMA-ES-dp. While on some individual functions the improvements observed from a solution quality perspective are rather large, on many other functions only minor though often statistically significant improvements are observed. iCMA-ES-tsc also performs competitive or superior to methods, such as MA-LSch-CMA (Molina et al. 2010) and PS-CMA-ES (Müller et al. 2009), which were developed with the goal of improving over iCMA-ES performance. This indicates that, instead of embedding CMA-ES into other algorithms to improve over its performance, a viable, alternative approach is to further fine-tune the parameter settings or maybe some design choices of iCMA-ES. This direction would involve to further parameterize choices that are currently fixed in the algorithm. Examples of such parameterizations are to treat further constants as parameters that are to be tuned or to consider alternative choices for specific functions. A concrete example could be the formula that is used to determine the initial population size, which is $4 + \lfloor a \ln(D) \rfloor$ (see also Table 1). Here, the constant 4 could be replaced by a real-valued parameter and different functions instead of + and ln may be considered.

It is also interesting to consider the impact the tuned parameter settings have on the behavior of iCMA-ES. In fact, a common pattern among the best performing tuned parameter settings we observed is that they lead to an increased exploration of the search space at least in the initial search phases and upon a restart of iCMA-ES. This more explorative behavior is implied by larger population sizes, larger step sizes, and a higher factor for the increase of the population size upon a restart of iCMA-ES. Interestingly, increasing search space exploration is also often the goal of hybrid algorithms such as the above-mentioned MA-LSch-CMA and PS-CMA-ES where CMA-ES is used as a local search. In fact, it seems that such increased exploration can be directly provided inside the iCMA-ES framework by modified parameter settings.

Our experimental results also indicate that using off-line automatic algorithm configuration to further improve adaptive algorithms is a viable approach—recall that iCMA-ES is such an adaptive algorithm where step sizes and search directions are adapted to the particular continuous optimization function under concern.

We have also presented initial results examining the role of the specific composition of a training set on the performance of the tuned parameter settings. On the one hand, these results indicated that the hybrid functions in the SOCO benchmark set alone are enough to derive high-performing tuned parameter settings. Maybe surprisingly, using only the uni-modal functions of the SOCO benchmark set resulted in a same level of performance of the tuned iCMA-ES on the CEC'05 benchmark set. Although it is known that uni-modal functions can be difficult to optimize, in future research the importance of the training set should be examined in much more detail. For this task, it is important to consider interactions between algorithm properties and properties of the training set. For example, the fact that iCMA-ES is rotationally invariant made it possible to use the SOCO benchmark set of functions which contains only unrotated functions—the rotational invariance implies that iCMA-ES's performance should be unaffected by rotations of the functions. For algorithms that are not invariant with respect to rotations, the usage of the SOCO benchmark set as training set may actually lead to poor performance. An interesting direction here would be to consider other benchmark set such as the BBOB benchmark function suite (see http://coco.gforge.inria.fr/doku.php?id=bbob-2012) or newly designed benchmark suites containing functions with specific properties for the tuning of continuous optimizers.

Acknowledgments The authors would like to thank Dr. Nikolaus Hansen for making publicly available the iCMA-ES codes. The authors also would like to thank Dr. Daniel Molina and Dr. Christian L. Müller for their updated results of MA-LSch-CMA and PS-CMA-ES on the CEC'05 benchmark function suite (taken from Liao et al. (2011a)). This work was supported by the Meta-X project funded by the Scientific Research Directorate of the French Community of Belgium. Thomas Stützle acknowledges support from the Belgian F.R.S.-FNRS, of which he is a Research Associate. Tianjun Liao acknowledges a fellowship from the China Scholarship Council.

References

Adenso-Diaz B, Laguna M (2006) Fine-tuning of algorithms using fractional experimental designs and local search. Operations Res 54:99–114

Auger A, Hansen N (2005) A restart CMA evolution strategy with increasing population size. In: Proceeding of IEEE Congress on Evolutionary Computation, CEC'05. IEEE Press, Piscataway, pp 1769–1776

Balaprakash P, Birattari M, Stützle T (2007) Improvement strategies for the F-Race algorithm: sampling design and iterative refinement. In: Bartz-Beielstein et al. (eds) Proceedings of International Conference on Hybrid Metaheuristics, LNCS, vol 4771. Springer, Berlin pp 108–122

Bartz-Beielstein T (2006) Experimental research in evolutionary computation: the new experimentalism. Springer, Berlin

Birattari M (2009) Tuning metaheuristics: a machine learning perspective, 1st edn. Springer, Berlin

Birattari M, Stützle T, Paquete L, Varrentrapp K (2002) A racing algorithm for configuring metaheuristics. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'02. Morgan Kaufmann, San Francisco, pp 11–18

Birattari M, Yuan Z, Balaprakash P, Stützle T (2010) F-Race and Iterated F-Race: an overview. In: Bartz-Beielstein et al. (eds) Experimental methods for the analysis of optimization algorithms. Springer, Berlin, pp 311–336

Hansen N (2010) The CMA evolution strategy: a tutorial. http://www.lri.fr/~hansen/cmatutorial.pdf

Hansen N, Ostermeier A (1996) Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: Proceedings of IEEE International Conference on Evolutionary Computation, CEC'96. IEEE Press, Piscataway, pp 312–317

Hansen N, Ostermeier A (2001) Completely derandomized self-adaptation in evolution strategies. Evolutionary Comput 9(2):159–195

Hansen N, Muller SD, Koumoutsakos P (2003) Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evolutionary Comput 11(1):1–18

Herrera F, Lozano M, Molina D (2010) Test suite for the special issue of soft computing on scalability of evolutionary algorithms and other metaheuristics for large scale continuous optimization problems. http://sci2s.ugr.es/eamhco/

Hoos HH, Stützle T (2004) Stochastic local search: foundations applications. Morgan Kaufmann, San Francisco

Hutter F, Babic D, Hoos HH, Hu AJ (2007) Boosting verification by automatic tuning of decision procedures. In: Proceedings of Formal Methods in Computer Aided Design, FMCAD'07. IEEE Press, Piscataway, pp 27–34

Hutter F, Hoos HH, Leyton-Brown K, Murphy K (2009a) An experimental investigation of model-based parameter optimisation: SPO and beyond. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO'09. ACM, New York, pp 271–278

Hutter F, Hoos HH, Leyton-Brown K, Stützle T (2009b) ParamILS: an automatic algorithm configuration framework. J Artif Intell Res 36(1):267–306

Liao T, Molina D, Montes de Oca MA, Stützle T (2011a) A note on the effects of enforcing bound constraints on algorithm comparisons using the IEEE CEC'05 Benchmark Function Suite. Tech. Rep. TR/IRIDIA/2011-010, IRIDIA. Université Libre de Bruxelles, Belgium

Liao T, Montes de Oca MA, Stützle T (2011b) Computational results for an automatically tuned CMA-ES with increasing population size on the CEC'05 benchmark set. http://iridia.ulb.ac.be/supp/IridiaSupp2011-023

Liao T, Montes de Oca MA, Stützle T (2011c) Tuning parameters across mixed dimensional instances: a performance scalability study of Sep-G-CMA-ES. In: Proceedings of the Workshop on Scaling Behaviours of Landscapes, Parameters and Algorithms of the Genetic and Evolutionary Computation Conference. GECCO'11, ACM, NY, pp 703–706

López-Ibáñez M, Dubois-Lacoste J, Stützle T, Birattari M (2011) The irace package, iterated race for automatic algorithm configuration Tech. Rep. TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium

Lozano M, Molina D, Herrera F (2011) Editorial scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems. Soft computing—a Fusion of Foundations. Methodol Appl 15:2085–2087

Mersmann O, Bischl B, Trautmann H, Preuss M, Weihs C, Rudolph G (2011) Exploratory landscape analysis. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 11. ACM, NY, pp 829–836

Molina D, Lozano M, García-Martínez C, Herrera F (2010) Memetic algorithms for continuous optimisation based on local search chains. Evolutionary Comput 18(1):27–63

Müller CL, Baumgartner B, Sbalzarini IF (2009) Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes. In: Proceeding of IEEE Congress on Evolutionary Computation, CEC 09. IEEE Press, Piscataway, pp 2685–2692

Nannen V, Eiben AE (2007) Relevance estimation and value calibration of evolutionary algorithm parameters. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. Morgan Kaufmann, San Francisco, pp 975–980

Ros R (2009) Benchmarking sep-CMA-ES on the BBOB- 2009 noisy testbed. In: Proceedings of the Conference Companion on Genetic and Evolutionary Computation Conference: late breaking papers. GECCO 09. ACM, NY , pp 2441–2446

Ros R, Hansen N (2008) A simple modification in CMA-ES achieving linear time and space complexity. In: Rudolph G et al. (eds) Parallel problem solving from nature PPSN X, vol 5199. Springer, Berlin, LNCS, pp 296–305

Smit SK, Eiben AE (2010) Beating the world champion evolutionary algorithm via REVAC tuning. In: Proceeding of IEEE Congress on Evolutionary Computation, CEC 10. IEEE Press, Piscataway, pp 1–8

Suganthan PN, Hansen N, Liang JJ, Deb K, Chen Y, Auger A, Tiwari S (2005) Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Tech. Rep. 2005005, Nanyang Technological University