
A Note on Bound Constraints Handling for the IEEE CEC'05 Benchmark Function Suite*

Tianjun Liao

liaotianjun@gmail.com

State Key Laboratory of Complex System Simulation, Beijing Institute of System Engineering, Beijing, China

Daniel Molina

daniel.molina@uca.es

Department of Computer Engineering, University of Cadiz, Cadiz, Spain

Marco A. Montes de Oca

mmontes@math.udel.edu

Department of Mathematical Sciences, University of Delaware, Newark, Delaware, USA

Thomas Stützle

stuetzle@ulb.ac.be

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

doi:10.1162/EVCO_a_00120

Abstract

The benchmark functions and some of the algorithms proposed for the special session on real parameter optimization of the 2005 IEEE Congress on Evolutionary Computation (CEC'05) have played and still play an important role in the assessment of the state of the art in continuous optimization. In this article, we show that if bound constraints are not enforced for the final reported solutions, state-of-the-art algorithms produce infeasible best candidate solutions for the majority of functions of the IEEE CEC'05 benchmark function suite. This occurs even though the optima of the CEC'05 functions are within the specified bounds. This phenomenon has important implications on algorithm comparisons, and therefore on algorithm designs. This article's goal is to draw the attention of the community to the fact that some authors might have drawn wrong conclusions from experiments using the CEC'05 problems.

Keywords

Continuous optimization, bound constraints, feasible solutions, IPOP-CMA-ES, MA-LSch-CMA.

1 Introduction

The special session on real parameter optimization of the 2005 IEEE Congress on Evolutionary Computation (CEC'05) has played an important role in evolutionary computation and other affine fields for two reasons. First, it provided a set of 25 scalable benchmark functions that anyone can use to evaluate the performance of new algorithms. Those 25 functions have become a standard benchmark set that researchers use to compare algorithms. The central role that this benchmark function suite plays is illustrated by the more than 800 citations (according to Google Scholar as of January 2014) to the original technical report that introduced it (Suganthan et al., 2005). Second, it served to assess the state of the art in continuous optimization. In particular, the

*Supplementary material can be found at <http://iridia.ulb.ac.be/supp/IridiaSupp2011-013>.

best performing algorithm of the special session, IPOP-CMA-ES (Auger and Hansen, 2005), is since then considered to be a representative of the state of the art in continuous optimization. Consequently, it is nowadays standard practice to compare the results of a new algorithm to the published results of IPOP-CMA-ES.

When evaluating algorithms, all should be run under the same conditions. Of particular interest in this article is whether or not to consider bound constraints. If we consider the definition of benchmark problems in continuous optimization, we may distinguish the following three situations.

- S1** Bound constraints are defined and are to be enforced at any stage of the search process—solutions outside the bounds are invalid.
- S2** Bound constraints are defined and are enforced for the final reported solutions; however, solutions outside the bounds may be evaluated and used to drive the search process.
- S3** No bound constraints are defined but bounds may be indicated to provide an initialization range.

The definition of each CEC'05 benchmark function states that each component of the solution vector \mathbf{x} must be a value in an interval $[x_{\min}, x_{\max}]$, $x_{\min} < x_{\max}$. There are two exceptions, which are functions f_7 and f_{25} , where the given interval specifies only an initialization range, and not a bound constraint. For the other 23 functions, their global optima are guaranteed to be within the specified bounds; on functions f_8 and f_{20} , the global optima are known to be on the bounds. However, later in the report it is mentioned that “All problems, except 7 and 25, have the global optimum within the given bounds and there is no need to perform search outside of the given bounds for these problems” (Suganthan et al., 2005, p. 40). This remark can be interpreted as allowing the algorithms to search outside the given bounds. While together with the definition of the CEC'05 benchmark functions this would indicate a type **S2** situation, this remark may have led to misinterpretation. In this paper, we give evidence that some claims of statistically significantly better performance than IPOP-CMA-ES (e.g., Müller et al., 2009; Molina et al., 2010) may not be valid because the authors may have interpreted the remark as a suggestion and reported results as when facing situation **S3**.

We became aware of possible confusion between situations **S2** and **S3** while reporting results for the CEC'05 benchmark functions when running experiments with the C implementation of CMA-ES available from Hansen's website, http://www.lri.fr/~hansen/cmaes_inmatlab.html, to implement IPOP-CMA-ES. This C version of CMA-ES does not use an explicit bound constraint handling mechanism. When running this code (without bound constraint handling) on the CEC'05 benchmark functions, we noticed that for a majority of the benchmark functions, the best solutions found do violate the bound constraints even though their global optima are known to be inside the bounds for 23 of the 25 functions. While it is known that this can happen on other functions,¹ we were surprised by the high frequency with which this phenomenon occurs on the CEC'05 benchmark function set.

¹One example is Schwefel's sine root function (Schwefel, 1981), which has its global optimum outside the usual feasible search space defined by bound constraints.

This observation raises the more general and critical issue of validity of published results that rely on the CEC'05 benchmark set. In fact, the vast majority of published articles do not explicitly report whether a bound constraint handling mechanism was used and if they do, many do not describe it. Perhaps more importantly, claims that an algorithm outperforms IPOP-CMA-ES in a statistically significant way (e.g., Müller et al., 2009; Molina et al., 2010) may not be valid because the comparison that supports those claims may include algorithms that enforce bound constraints in some way (as in **S1** or **S2**) and algorithms that do not (as in **S3**).

To show how misleading such a comparison can be, we report experimental results on the impact of handling bound constraints with IPOP-CMA-ES. We evaluate three variants of the C version of IPOP-CMA-ES. In the first variant, bound constraints are never enforced (we refer to this variant as IPOP-CMA-ES-*ncb*, where *ncb* stands for “never clamp bounds”); it simulates situation **S3**. The second is a variant in which we introduce a mechanism to enforce bound constraints (*acb* for “always clamp bounds”; this variant is referred to as IPOP-CMA-ES-*acb*). In particular, we clamp a variable's value that is outside the variable's feasible domain dimension by dimension to the closest boundary value; that is, if $x_i < x_{\min}$ we set $x_i = x_{\min}$ and if $x_i > x_{\max}$ we set $x_i = x_{\max}$ before evaluating these solutions and continuing with the algorithm execution. Note that this variant can tackle both situations, **S1** and **S2**: in the **S2** case it can be seen as a simple way to handle bound constraints and to ensure that final solutions are feasible. Additionally, we have run experiments with a variant that directly addresses situation **S2**; in this variant, we let IPOP-CMA-ES search outside the bounds without restrictions but take care that the final solution reported is the best feasible solution that has been identified during the search process. The results with this latter version were very poor and we report them only in the supplementary material to this article.² The same three variants are tested using a memetic algorithm, MA-LSch-CMA (Molina et al., 2010), which is a recent memetic algorithm that uses CMA-ES as a local search and which was reported to perform better than IPOP-CMA-ES at a statistically significant level.

2 Experiments on Enforcing Bound Constraints

In the first experiment, we followed the protocol described by Suganthan et al. (2005), that is, we ran IPOP-CMA-ES using its default parameter settings 25 times on each function and recorded the evolution of the objective function value with respect to the number of function evaluations used. The maximum number of function evaluations was $10,000 \cdot D$, where $D \in \{10, 30, 50\}$ is the dimensionality of a function. The algorithm stops when the maximum number of evaluations is reached or the error is lower than 10^{-8} . Error values lower than this optimum threshold are considered equal to 10^{-8} .

We compare IPOP-CMA-ES-*ncb* and IPOP-CMA-ES-*acb* in Table 1.³ The two-sided Wilcoxon matched-pairs signed-rank test at the .05 level of the error of first type was used to check for statistical differences on each function. Depending on the dimensionality, in 14 to 17 functions IPOP-CMA-ES-*ncb* obtains final solutions outside the bounds. In most of the functions for which infeasible solutions are found, all the 25 runs return final solutions that are outside the bounds. We observed statistically

²This version is referred to as *kbf* for “known best feasible” in the supplementary material.

³The results in Tables 1, 2, and 3 are based on average errors. Additional tables are given in the supplementary pages to this article; they show the median results and more detailed information such as the best, 25th percentile, median, 75th percentile, and worst error values for each function.

Table 1: The comparison between IPOPOP-CMA-ES-*ncb* and IPOPOP-CMA-ES-*acb* over 25 independent runs for each of the CEC'05 functions except f_7 and f_{25} (these two functions are excluded as for these only an initialization range is specified). \odot denotes that all 25 final solutions are outside the bounds. \ominus denotes that some but not all of the 25 final solutions are outside the bounds. Symbols $<$, \approx , and $>$ denote whether the performance of IPOPOP-CMA-ES-*ncb* is statistically better, indifferent, or worse than that of IPOPOP-CMA-ES-*acb* according to a two-sided Wilcoxon matched-pairs signed-rank test at the .05 α -level. The average errors that correspond to a statistically better result are highlighted. The numbers in parentheses at the bottom of the table represent the frequency of $<$, \approx , and $>$, respectively.

f_{cec}	10 dimensions		30 dimensions		50 dimensions	
	<i>ncb</i>	<i>acb</i>	<i>ncb</i>	<i>acb</i>	<i>ncb</i>	<i>acb</i>
f_1	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08
f_2	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08
f_3	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08
f_4	1.00E-08 \approx	1.00E-08	2.44E+03 $\odot \approx$	6.58E+02	1.32E+05 $\odot >$	1.43E+04
f_5	1.00E-08 $\odot \approx$	1.00E-08	2.30E+01 $\odot >$	1.00E-08	7.91E+02 $\odot >$	7.41E-02
f_6	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08	1.00E-08 \approx	1.00E-08
f_8	2.01E+01 $\odot \approx$	2.00E+01	2.07E+01 $\odot >$	2.04E+01	2.11E+01 $\odot >$	2.09E+01
f_9	1.59E-01 \approx	1.59E-01	1.01E+00 $<$	1.87E+00	1.12E+00 $\odot <$	4.36E+00
f_{10}	1.19E-01 \approx	3.18E-01	1.37E+00 \approx	1.44E+00	2.36E+00 \approx	2.89E+00
f_{11}	6.44E-01 $\odot >$	1.00E-08	6.36E+00 $\odot >$	7.17E-02	1.49E+01 $\odot >$	9.94E-02
f_{12}	6.77E+01 $\odot <$	4.07E+03	1.38E+03 $\odot <$	1.19E+04	7.38E+03 $\odot <$	4.25E+04
f_{13}	6.78E-01 \approx	6.49E-01	2.47E+00 \approx	2.63E+00	4.31E+00 \approx	4.44E+00
f_{14}	2.61E+00 $\odot >$	1.96E+00	1.28E+01 $\odot \approx$	1.26E+01	2.34E+01 $\odot >$	2.28E+01
f_{15}	2.00E+02 $\odot \approx$	2.15E+02	2.01E+02 $\odot >$	2.00E+02	2.01E+02 $\odot >$	2.00E+02
f_{16}	9.02E+01 \approx	9.04E+01	7.95E+01 $\odot >$	1.48E+01	1.36E+02 $\odot >$	1.10E+01
f_{17}	1.33E+02 $\odot \approx$	1.17E+02	4.31E+02 $\odot >$	2.52E+02	7.69E+02 $\odot >$	1.91E+02
f_{18}	7.48E+02 $\odot >$	3.16E+02	8.16E+02 $\odot <$	9.04E+02	8.36E+02 $\odot <$	9.13E+02
f_{19}	7.75E+02 $\odot >$	3.20E+02	8.16E+02 $\odot <$	9.04E+02	8.36E+02 $\odot <$	9.13E+02
f_{20}	7.62E+02 $\odot >$	3.20E+02	8.16E+02 $\odot <$	9.04E+02	8.36E+02 $\odot <$	9.15E+02
f_{21}	1.06E+03 $\odot >$	5.00E+02	8.57E+02 $\odot >$	5.00E+02	7.15E+02 $\odot \approx$	6.64E+02
f_{22}	6.38E+02 $\odot <$	7.28E+02	5.98E+02 $\odot <$	8.10E+02	5.00E+02 $\odot <$	8.19E+02
f_{23}	1.09E+03 $\odot >$	5.86E+02	8.69E+02 $\odot >$	5.34E+02	7.27E+02 $\odot \approx$	6.97E+02
f_{24}	4.05E+02 $\odot >$	2.33E+02	2.10E+02 $\odot >$	2.00E+02	2.14E+02 $\odot >$	2.00E+02
	$(<, \approx, >): (2, 13, 8)$ $< \text{ or } > : 10/23 (43\%)$ functions \odot or \ominus : 14/23 (61%)		$(<, \approx, >): (6, 8, 9)$ $< \text{ or } > : 15/23 (65\%)$ functions \odot or \ominus : 16/23 (70%)		$(<, \approx, >): (6, 8, 9)$ $< \text{ or } > : 15/23 (65\%)$ functions \odot or \ominus : 17/23 (74%)	

significant differences between IPOPOP-CMA-ES-*ncb* and IPOPOP-CMA-ES-*acb* when the final solutions of IPOPOP-CMA-ES-*ncb* are outside the bounds. While a priori we expected that IPOPOP-CMA-ES-*ncb* would give worse results than IPOPOP-CMA-ES-*acb* because, for these functions, the optima are known to be inside the bounds, IPOPOP-CMA-ES-*ncb* outperforms IPOPOP-CMA-ES-*acb* in six functions ($f_9, f_{12}, f_{18}, f_{19}, f_{20}$, and f_{22} in dimensions 30 and 50). In all these functions except for f_9 , all solutions obtained by IPOPOP-CMA-ES-*ncb* are outside the bounds.

Table 2: The comparison between MA-LSch-CMA-*ncb* and MA-LSch-CMA-*acb* over 25 independent runs for each of the CEC'05 functions except f_7 and f_{25} (these two functions are excluded as for these only an initialization range is specified). For an explanation of the symbols and their interpretation we refer to the caption of Table 1.

f_{cec}	10 dimensions		30 dimensions		50 dimensions	
	<i>ncb</i>	<i>acb</i>	<i>ncb</i>	<i>acb</i>	<i>ncb</i>	<i>acb</i>
f_1	1.00E-08	≈	1.00E-08	1.00E-08	≈	1.00E-08
f_2	1.00E-08	≈	1.00E-08	2.51E-08	≈	1.00E-08
f_3	3.68E+02	>	1.00E-08	4.41E+03	⊙ ≈	2.75E+04
f_4	1.00E-08	≈	5.54E-03	1.28E+02	<	3.02E+02
f_5	7.78E+01	⊙ >	6.75E-07	6.12E+02	⊙ <	1.26E+03
f_6	1.00E-08	≈	3.19E-01	2.31E+02	⊙ >	1.12E+00
f_8	2.00E+01	⊙ ≈	2.00E+01	2.00E+01	≈	2.00E+01
f_9	1.00E-08	≈	1.00E-08	1.00E-08	≈	1.00E-08
f_{10}	3.14E+00	≈	2.67E+00	2.00E+01	≈	2.25E+01
f_{11}	4.53E+00	⊙ >	2.43E+00	2.20E+01	≈	2.15E+01
f_{12}	2.95E+02	⊙ ≈	1.14E+02	7.52E+02	⊙ <	1.67E+03
f_{13}	5.03E-01	≈	5.45E-01	2.04E+00	≈	2.03E+00
f_{14}	2.87E+00	⊙ >	2.25E+00	1.32E+01	⊙ >	1.25E+01
f_{15}	2.27E+02	⊙ ≈	2.24E+02	2.59E+02	⊙ <	3.00E+02
f_{16}	9.45E+01	⊙ ≈	9.18E+01	1.06E+02	≈	1.26E+02
f_{17}	1.04E+02	≈	1.01E+02	1.66E+02	⊙ ≈	1.83E+02
f_{18}	8.20E+02	⊙ <	8.84E+02	8.22E+02	⊙ <	8.98E+02
f_{19}	8.17E+02	⊙ ≈	8.78E+02	8.22E+02	⊙ <	9.01E+02
f_{20}	7.69E+02	⊙ ≈	8.63E+02	8.23E+02	⊙ <	8.96E+02
f_{21}	8.57E+02	⊙ ≈	7.94E+02	8.47E+02	⊙ >	5.12E+02
f_{22}	7.63E+02	⊙ >	7.53E+02	5.34E+02	⊙ <	8.80E+02
f_{23}	8.74E+02	≈	8.88E+02	8.40E+02	⊙ >	5.34E+02
f_{24}	3.94E+02	⊙ >	2.28E+02	2.14E+02	⊙ >	2.00E+02

($<$, \approx , $>$): (1, 16, 6)	($<$, \approx , $>$): (8, 10, 5)	($<$, \approx , $>$): (8, 6, 9)
$<$ or $>$: 7/23 (30%)	$<$ or $>$: 13/23 (57%)	$<$ or $>$: 17/23 (74%)
functions \emptyset or \odot : 13/23 (57%)	functions \emptyset or \odot : 18/23 (79%)	functions \emptyset or \odot : 19/23 (83%)

Table 2 shows the performance of *ncb* and *acb* versions for MA-LSch-CMA (MA-LSch-CMA is run using default parameter settings). Again, version *ncb* obtains many final solutions outside the bounds: this is the case on 18 and 19 functions for 30 and 50 dimensions, respectively. Taking the 50-dimensional benchmark functions as an example, all functions for which MA-LSch-CMA-*ncb* outperforms MA-LSch-CMA-*acb* are cases in which all solutions obtained by MA-LSch-CMA-*ncb* are outside the bounds ($f_5, f_{11}, f_{12}, f_{15}, f_{18}, f_{19}, f_{20}$, and f_{22}).

3 The Impact of Bound Handling on Algorithm Comparisons

We now focus on the comparison of the average errors between PS-CMA-ES (Müller et al., 2009), MA-LSch-CMA (Molina et al., 2010), IPOP-CMA-ES-*ncb*, and IPOP-CMA-ES-05. IPOP-CMA-ES-05 uses the MATLAB version of CMA-ES and was used

to generate the results for the CEC'05 benchmark functions presented in Auger and Hansen (2005); it handles bound constraints by an approach based on penalty functions, which is described in Hansen et al. (2009b). PS-CMA-ES and MA-LSch-CMA are examples of algorithms that have been reported to outperform IPOP-CMA-ES-05; they use CMA-ES as a local search operator inside a particle swarm optimization algorithm and a real-coded steady-state genetic algorithm, respectively.

Table 3 shows that PS-CMA-ES, MA-LSch-CMA, but also IPOP-CMA-ES-*ncb*, are superior to IPOP-CMA-ES-05 in 30 and 50 dimensions in the sense that they find more often better average errors than IPOP-CMA-ES-05. However, there is an interesting pattern, whether IPOP-CMA-ES-*ncb* has the final solutions outside the bounds or not. Let us focus on the cases where IPOP-CMA-ES-*ncb* obtains all solutions outside the bounds and statistically significantly improves over IPOP-CMA-ES-05 (as indicated by the “<” symbol in Table 3). In many such cases, PS-CMA-ES does obtain the same average errors (see, e.g., functions f_{18} – f_{20} and f_{24} for both 30 and 50 dimensions, and function f_{22} for 50 dimensions), or very similar values (see, e.g., functions f_{21} and f_{23} for 50 dimensions); such cases are underlined in Table 3. A similar pattern arises for the published results of the MA-LSch-CMA algorithm. Interestingly, MA-LSch-CMA checks bound constraints only for the solutions generated by the steady-state GA part, but not for solutions returned by the CMA-ES local search. After rerunning the publicly available version of MA-LSch-CMA, we found that it returns, for several functions, infeasible final solutions (as indicated by the symbols \odot and \circ in Table 3). This knowledge, together with the similar pattern of the average errors, puts at least serious doubts on the fact of whether the average errors reported in Müller et al. (2009) correspond all to solutions that are inside the bounds. This analysis shows that claims of superiority of one algorithm over another may in fact not be valid if the algorithms confound situations **S2** and **S3**.

4 Conclusions

In this note, we first show that IPOP-CMA-ES and MA-LSch-CMA surprisingly often return infeasible solutions for the CEC'05 benchmark functions if the situations **S2** and **S3** are confounded. In many cases, these infeasible solutions are better than the best feasible solutions found even though it is known that the optimal solutions are within the bounds. This issue points toward a significant impact on CEC'05 benchmark functions for what concerns algorithm comparisons. In particular, claims about superior performance of one algorithm over another might be erroneous as infeasible solutions with respect to bound constraints may have been reported.

It is interesting to examine whether misunderstandings may potentially arise in other benchmark sets such as those proposed by Tang et al. (2007); Hansen et al. (2009a); and Herrera et al. (2010). For the CEC'08 benchmark set (Tang et al., 2007), formulations analogous to the description in the CEC'05 benchmark set are used, thus giving a chance of misinterpretations analogous to those indicated in the introduction. In the BBOB benchmark definition, it is stated that all functions are defined and can be evaluated at any point, but that the search domain is $[-5, 5]^D$, where D is the dimension of the search space. Since the notion of search domain leaves room for interpretation, it may remain unclear whether situation **S2** or **S3** is intended. We were confirmed that for BBOB the setting **S3** is intended (Hansen, 2013). In the SOCO benchmark set (Herrera et al., 2010), each function definition restricts the feasible interval. This clearly excludes situation **S3** but leaves some doubt as to whether situation **S1** or **S2** is intended (the latter is actually the case).

To avoid possible doubts about the feasibility of the solutions, we strongly recommend that in the future every paper that reports results using the IEEE CEC'05 benchmark function suite, or any other benchmark suite, should (1) explicitly describe the bound handling mechanism used (if any), (2) explicitly check the feasibility of the final solutions,⁴ and (3) present the final solutions at least in supplementary material for the article to avoid misinterpretation. Regarding benchmarking, we recommend that the designers of benchmark sets clearly state for which of the situations **S1**, **S2**, or **S3** the benchmark set is designed to be used. In addition, if code is provided, it should support proper evaluation by returning null or infinity as values if generated solutions violate bound constraints in situation **S1** or by providing tools for checking solution feasibility, and computing statistics in the case of situation **S2**.

All the solutions generated by the algorithms discussed in this article are available at <http://iridia.ulb.ac.be/supp/IridiaSupp2011-013>.

Acknowledgments

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 senior research associate. Tianjun Liao acknowledges a fellowship from the China Scholarship Council and support from the National Natural Science Foundation of China. We also want to acknowledge the detailed comments of the anonymous referees and the associate editor, which helped to improve the presentation of the article. In particular, we thank the anonymous referee who suggested explicitly spelling out situations **S1** to **S3**.

References

- Auger, A., and Hansen, N. (2005). A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC'05*, pp. 1769–1776.
- Hansen, N. (2013). Personal communication.
- Hansen, N., Finck, S., Ros, R., and Auger, A. (2009a). Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA.
- Hansen, N., Niederberger, A. S. P., Guzzella, L., and Koumoutsakos, P. (2009b). A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197.
- Herrera, F., Lozano, M., and 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. URL: <http://sci2s.ugr.es/eamhco/>.
- Molina, D., Lozano, M., García-Martínez, C., and Herrera, F. (2010). Memetic algorithms for continuous optimization based on local search chains. *Evolutionary Computation*, 18(1):27–63.
- Müller, C., Baumgartner, B., and Sbalzarini, I. (2009). Particle swarm CMA evolution strategy for the optimization of multi-funnel landscapes. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC'09*, pp. 2685–2692.
- Schwefel, H. P. (1981). *Numerical optimization of computer models*. New York: Wiley.

⁴We note that many bound handling mechanisms use penalty approaches such as the one used by Hansen et al. (2009b). However, it still is possible to obtain infeasible solutions if bound constraints are not explicitly enforced for the final reported solutions, no matter what penalty approach is used as the bound handling mechanism.

Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A., and Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report 2005005, NTU.

Tang, K., Yao, X., Suganthan, P., MacNish, C., Chen, Y., Chen, C., and Yang, Z. (2007). Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Technical report, NICAL. Available at <http://nical.ustc.edu.cn/cec08ss.php>

