Special Session on Real-Parameter Optimization at CEC'05, Edinburgh, 2 - 5 Sept. 2005

In this document, I provide a summary of comments given to us by potential participants.

1. Properties of benchmark functions:

1. shifted functions (i.e. not all variables should have the same numerical value at the optimum) rotated functions, noisy functions with noise in variables and in fitness values, arbitrary global optimal function values,
2. no clear structure in the fitness landscape
3. functions with high condition numbers.
4. local optima with wide basins of attractions and the global optimum with narrow basin of attraction
5. continuous and non-continuous functions.
6. the number of local optima from 0 to 100s / 1000s
7. additively non-separable or non-decomposable.
8. continuous / discrete / mixed variables
9. global optimum at the search boundary for some variables
10. scalability (i.e. 10 dimensions to 100 dimensions)
11. upper/lower bound constraints (at this stage no other (in)equality constraints)

Keeping these properties in mind, we plan to develop about 20 test functions. Each function is likely to have a few of the properties listed above. The noisy functions may have a restriction in that the noise values will be fixed once. If we wish to have different noise values included at every generation into fitness/variables, then the actual numbers can be different for different algorithms, as they have to be generated on different machines/environment. We propose the following general form for the benchmark functions.

$\boldsymbol{f_i(x)}$: basic function used to construct the composition function

$\boldsymbol{F(x)}$: composition function

$\boldsymbol{n}$: total number of basic functions $\boldsymbol{f_i(x)}$ (such as Sphere, Rastrigin, Rosenbrock, etc.)

$\boldsymbol{D}$: dimension

$\boldsymbol{Mg}$: global rotation matrix (If M_g=Ml_i=**I**, then there is no rotation. If $M_g \neq \mathbf{I}$, then Ml_i=**I**)

$\boldsymbol{Ml_i}$: local rotation matrix for each $\boldsymbol{f_i(x)}$

$\boldsymbol{o_i}$: new shifted optimal position for each $\boldsymbol{f_i(x)}$

$$F(x) = \sum_{i=1}^{n} \{w_i * [\, f_i(Ml_i * (Mg * (x - o_i) / \lambda_i)) + bias_i \,]\}$$

$\boldsymbol{w_i}$: weight value for each $\boldsymbol{f_i(x)}$, calculated as below:

$$\boldsymbol{w_i = \exp\left(-\frac{\sum_{k=1}^{D} (x_k - o_{ik})^2}{2\sigma_i^2}\right)} \text{ (}i.e.\text{ Gaussian shape)} \qquad \text{and} \quad \boldsymbol{w_i = w_i / \sum_{i=1}^{n} w_i}$$

λ_i and σ_i used to control each $f_i(x)$'s range of coverage, a small λ_i and σ_i gives a narrow range for that $f_i(x)$. o_i defines the global or local optima's position and $bias_i$ determines which optima is global optima. Global optima at the origin will be avoided. If n=1, we'll have a single function. We'll have about 7 single functions and about 13 composition functions??

2. Performance Evaluation Criteria:

1. We report several performance indicators observed at different points during the evolution. The performance indicators are: median, success rate, mean, maximum, minimum, standard deviation. These indicators should be reported in tabular form at 1e3, 5e3, 1e4, 5e4, 1e5, 2e5, 4e5, 6e5, 8e5 and 1e6. Convergence graphs can also be included in addition to the tables. As different papers may not have all the algorithms being compared, we recommend the usage of tables. We recommend 50 runs??
2. Counting just FEs hides the complexity of different algorithms. Hence, we propose to use the following:
   a) run a test program like (for example)
      do 10000 times
         x=x+x; x=x/2; x=x*x; x=sqrt(x); x=ln(x); x=exp(x); y=x/x;
      It gives a computing time T0
   b) evaluate the computing time just for the benchmark function. For N evaluations, it gives T1.
   c) the complete computing time for the algorithm with N evaluations is T2.

   Then the criterion to reflect the complexity of the algorithm: (T2-T1)/T0
   At this stage, we prefer this ratio computation over running all programs on a single machine.

3. How to incorporate the pre-runs needed to adjust various parameters. With respect to this problem, we can request the participants to list the parameters, their dynamic ranges, and how to search/estimate parameter values, given a new optimization problem. One approach would be to propose parameter values as functions of dimensions, dynamic ranges of variables, etc. We discourage participants searching for a distinct set of parameters for each problem/dimension/etc.

3. Software Concerns:

If you are only willing to let us share the executables of your real-world problems, then they can also be included. However, their usage may be limited.

We do not require that the complete codes of your search algorithms would be made available in public domain, although we encourage this.

We will provide Matlab codes, mathematical expressions and well formatted data files for the benchmark functions. At this stage, we are fortunate to have two participants happy to help to translate the codes for circulation to all. Dr Vitaliy Feoktistov (Vitaliy.Feoktistov@ema.fr) has kindly agreed to translate the codes to Visual Studio.NET. If you wish to make use of these translated codes, please communicate with Dr Vitaliy Feoktistov who may have some additional

information for you. Dr Ying-Ping Chen (ypchen@csie.nctu.edu.tw) has kindly agreed to translate to Java and C++ approximately within 2 weeks.

We believe that we've now major languages covered and that the C++/Java codes may run on Windows, Unix and various Liunx versions.

4. Attempt Just Two Times:

We may not be able to get everything agreeable to everyone in our first attempt. Therefore, we propose that we work with a set of benchmark functions for the CEC 2005 and based on our CEC 2005 experience/feedbacks, we shall go on to develop an improved benchmark suite for the edited volume. In other words, I propose that we revise the functions after CEC 2005 workshop, instead of revising them frequently. This will interfere in our efforts to translate codes into different languages.

5. Some functions/weblinks to consider:

1. Weiserstrass fractal functions (Marc Schoenauer)

2. Fletcher-Powell function (Hans-Paul Schwefel)

3. If you go to a chapter of my homepage: http://ls11-www.cs.uni-dortmund.de/lehre/wiley/index.html and click on "Appendix A" you may get the full test functions catalogue. (Hans-Paul Schwefel)

4. To begin, you may have a look at http://clerc.maurice.free.fr/pso/Semi-continuous_challenge/Semi-continuous_challenge.htm (Maurice Clerc)

5. I am interested in real-parameter optimization on hydrologic models for river flood routing and rainfall intensity using Harmony Search (www.hydroteq.com). (Zong Woo Geem)

6. You will find other elliptic function (Cigar and Tablet) with high condition number (build using the same procedure) in the following paper: p7 of the paper "Hansen, N., S.D. Müller and P. Koumoutsakos (2003). Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). Evolutionary Computation, 11(1), pp. 1-18"

   available at: http://www.bionik.tu-berlin.de/user/niko/publications.html
   Note that rotating the elliptic functions:
   $f_{\rm \it rotated elli}(x)= $f_{\rm \it elli}(P*x)$ with P an orthogonal matrix won't change the condition number (Anne Auger).

6. Methods and interested participants:

1. Ant Colony: SIARRY Patrick, Walid TFAILI, Johann Dréo,

2. DE: Bogdan Filipic, Dr. Hussein Abbass, M. Fatih Tasgetiren, Vitaliy FEOKTISTOV, Darrell Whitley, David Mayer

3. EP: Thomas Philip Runarsson, Dr. Hussein Abbass, Dapeng Li

4. EDAs: Marcus Gallagher

5. ES: Marc Schoenauer, William E. Hart, Thomas Philip Runarsson, Mike Preuss, Dr. Hussein Abbass, Nikolaus Hansen, Anne Auger, Silvia Alonso, Blas Galván, Gabriel Winter, J. Ignacio Franquiz, Begoña González,

6. Evol. Pattern Search: William E. Hart,

7. GA: SIARRY Patrick, Bogdan Filipic, SALHI Lounis,David Coley, V. Sundararajan, Dapeng Li, Asanga Ratnaweera, Francisco Herrera, Manuel Lozano, Fabio Boschetti, Darrell Whitley, Pedro J. Ballester, Jonathan Carter, Kalyanmoy Deb, Domingo Ortiz Boyer

8. Harmony search: Zong Woo Geem

9. Hybrid EAs: William E. Hart, Ong Yew Soon, Darrell Whitley,

10. Nelder Mead: Thomas Bartz-Beielstein

11. PSO: Marc Schoenauer, SIARRY Patrick, SALHI Lounis, Maurice Clerc, E. Parsopoulos, Asanga Ratnaweera, M. Fatih Tasgetiren, Ying-ping Chen, Suganthan, Johann Dréo,

12. QN (Quasi-Newton): Marc Schoenauer, Thomas Bartz-Beielstein, Darrell Whitley,

13. SA: SIARRY Patrick, V. Sundararajan, Darrell Whitley, Nélio Henderson, Gustavo M. Platt, Anna I Esparcia-Alcazar, Johann Dréo,

14. TS (Tabu): SIARRY Patrick, Darrell Whitley,

7. Relevance to CEC (Congress on Evolutionary Computation)

If you are more likely to use only non-evolutionary algorithm(s), then the relevance of your contribution to CEC'05 may be questioned. We will have to communicate with the organizers on this. You may please inform us if this is the case.