Differential Evolution (DE)

for Continuous Function Optimization (an algorithm by Kenneth Price and Rainer Storn)

Table of contents

- History
- Basics
- Practical Advice
- Java Code
- C Code
- MATLAB ® Code
- Scilab Code
- PSather Code
- C++ Code
- Fortran90 Code
- Python Code
- Labview Code
- R Code
- Pascal Code
- C# Code
- Maple Code
- Demo-Programs
- Applications of DE
- Commercial SW using DE
- Academic SW-packages using DE
- Other links, books, ...
- Conferences
- Special Issue on DE in IEEE TEVC
- Contributors to this page

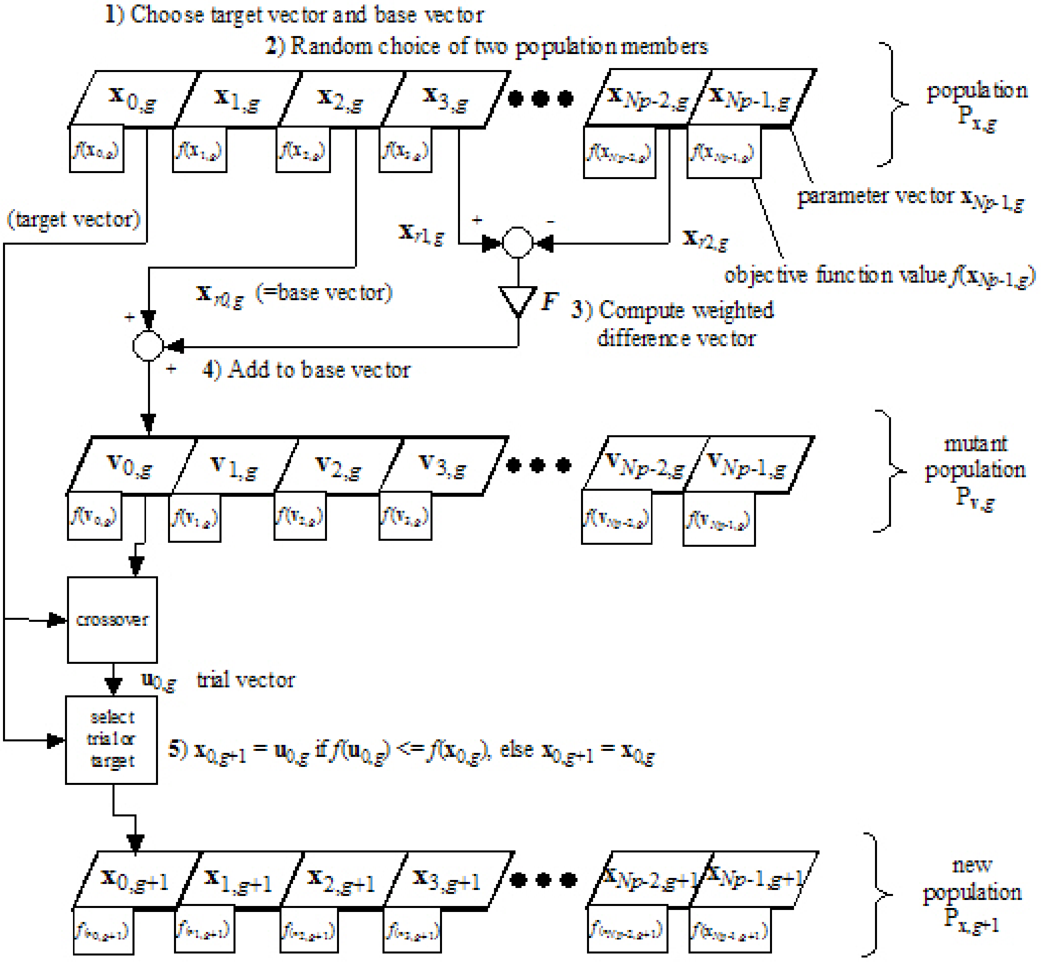
History

Differential Evolution grew out of Ken Price's attempts to solve the Chebychev Polynomial fitting Problem that had been posed to him by Rainer Storn. A breakthrough happened, when Ken came up with the idea of using vector differences for perturbing the vector population. Since this seminal idea a lively discussion between Ken and Rainer and endless ruminations and computer simulations on both parts yielded many substantial improvements which make DE the versatile and robust tool it is today. The "DE community" has been growing since the early DE years of 1994 - 1996 and ever more researchers are working on and with DE. Those scientists who contributed actively to this homepage are listed at the bottom in alphabetical order. It is the strong wish of Ken and Rainer that DE will be developed further by scientists around the world and that DE may improve to help more users in their daily work. This wish is the reason why DE has not been patented in any way.

Basics

DE is a very simple population based, stochastic function minimizer which is very powerful at the same time. DE managed to finish 3rd at the First International Contest on Evolutionary Computation

(1stICEO) which was held in Nagoya, may 1996. DE turned out to be the best genetic type of algorithm for solving the real-valued test function suite of the 1st ICEO (the first two places were given to non-GA type algorithms which are not universally applicable but solved the test-problems faster than DE). The crucial idea behind DE is a scheme for generating trial parameter vectors.



Basically, DE adds the weighted difference between two population vectors to a third vector. This way no separate probability distribution has to be used which makes the scheme completely self-organizing. For further details see the literature page.

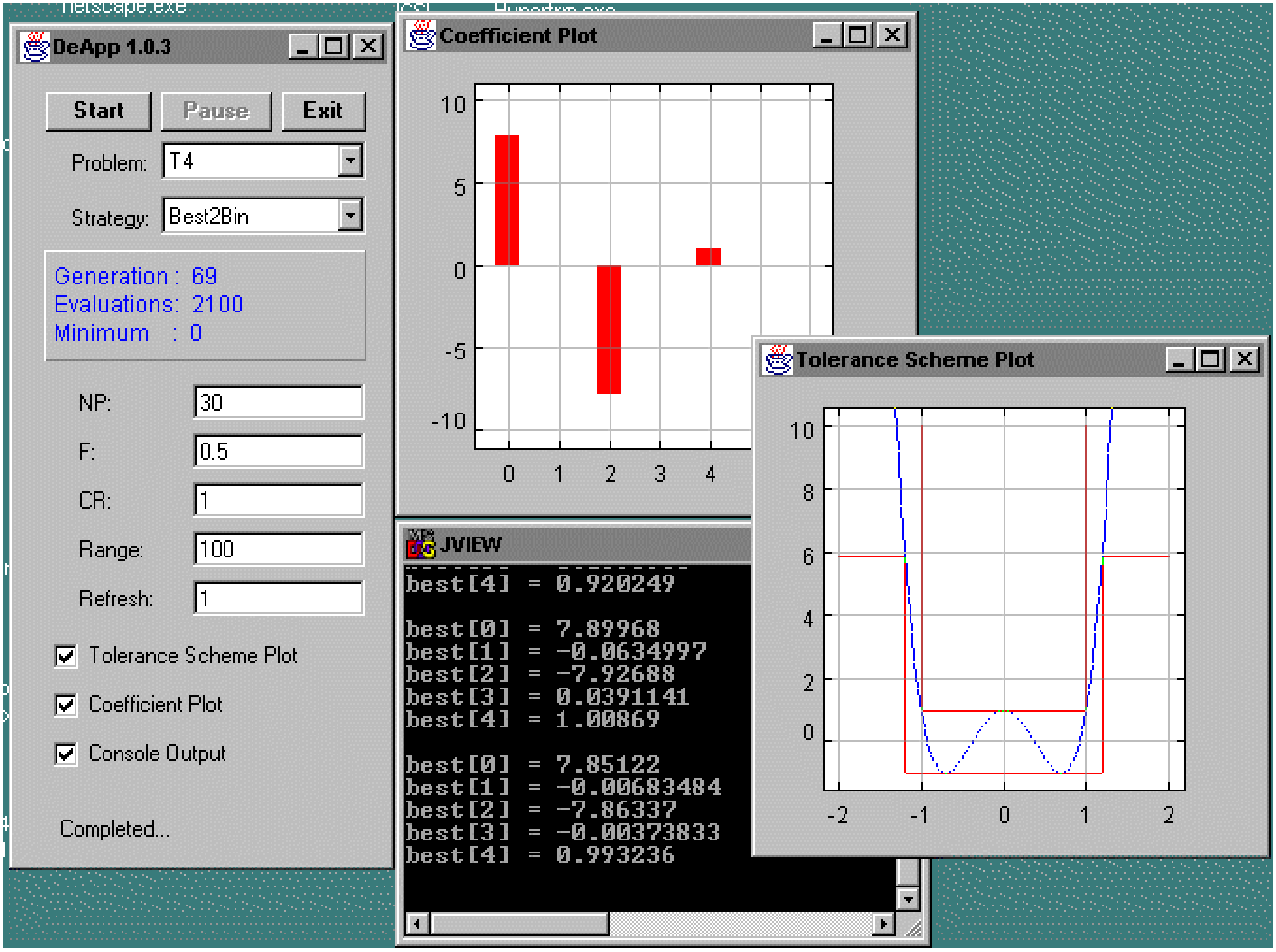
Practical Advice

If you are going to optimize your own objective function with DE, you may try the following classical settings for the input file first: Choose method e.g. DE/rand/1/bin, set the number of parents NP to 10 times the number of parameters, select weighting factor F=0.8, and crossover constant CR=0.9. It has been found recently that selecting F from the interval [0.5, 1.0] randomly for each generation or for each difference vector, a technique called dither, improves convergence behaviour significantly, especially for noisy objective functions. It has also been found that setting CR to a low value, e.g. CR=0.2 helps optimizing separable functions since it fosters the search along the coordinate axes. On the contrary this choice is not effective if parameter dependence is encountered, something which is frequently occuring in real-world optimization problems rather than artificial test functions. So for parameter dependence the choice of CR=0.9 is more appropriate. Another interesting empirical finding is that rasing NP above, say, 40 does not substantially improve the convergence, independent of the number of parameters. It is worthwhile to experiment with these suggestions. Make sure that you initialize your parameter vectors by exploiting their full numerical range, i.e. if a parameter is allowed to exhibit values in the range [-100, 100] it's a good idea to pick the initial values from this range instead of unnecessarily restricting diversity.

Keep in mind that different problems often require different settings for NP, F and CR (have a look into the different papers to get a feeling for the settings). If you still get misconvergence you might want to try a different method. We mostly use DE/rand/1/... or DE/best/1/... . The crossover method is not so important although Ken Price claims that binomial is never worse than exponential. In case of misconvergence also check your choice of objective function. There might be a better one to describe your problem. Any knowledge that you have about the problem should be worked into the objective function. A good objective function can make all the difference.

Java Code

In order to make it easy to do DE optimization on every platform with the support of graphics a Java application of a DE optimizer has been written. The current version (1.0.3) runs now both in JDK 1.0.2 as well as JDK 1.1.7. The code has been enriched with the magnificent plotting capabilities of ptplot, written by Edward A. Lee and Christopher Hylands from the University of California, Berkeley. Now it is possible to zoom in and out of the plots as the optimization is ongoing. A corrected version of the code documentation is also available.



Problem: T4

Strategy: Best2Bin

Generation : 69
Evaluations: 2100
Minimum : 0

NP: 30

F: 0.5

CR: 1

Range: 100

Refresh: 1

☑ Tolerance Scheme Plot

☑ Coefficient Plot

☑ Console Output

Completed...

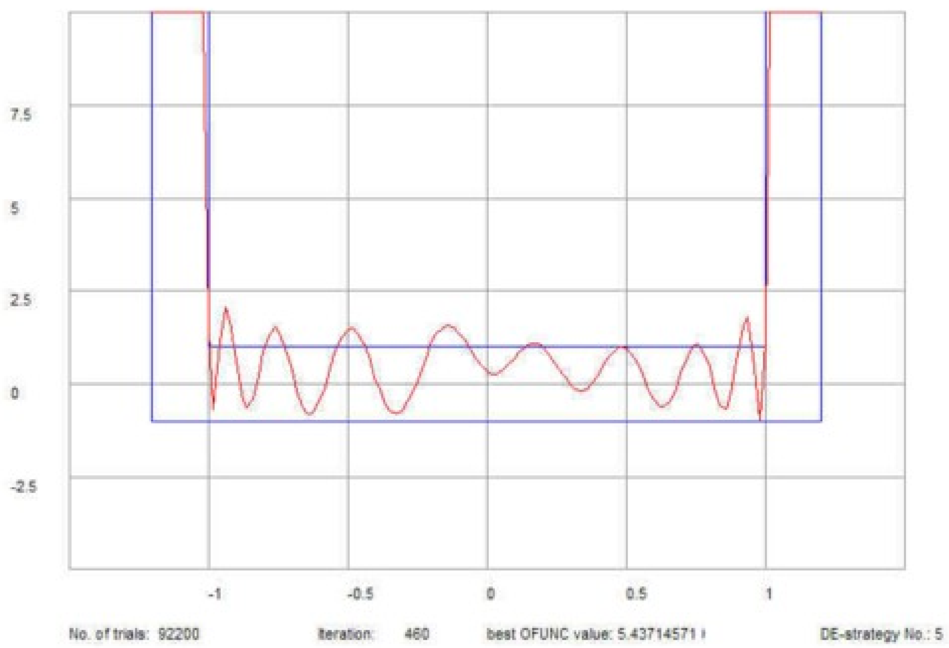
```
best[4] = 0.920249

best[0] = 7.89968
best[1] = -0.0634997
best[2] = -7.92688
best[3] = 0.0391141
best[4] = 1.00869

best[0] = 7.85122
best[1] = -0.00683484
best[2] = -7.86337
best[3] = -0.00373833
best[4] = 0.993236
```

C Code

The latest C code from the book Differential Evolution - A Practical Approach to Global Optimization is available here by courtesy of Springer publisher. The code has been written with MS Visual C++ v5.0 and also contains some MS Windows based graphics routines (see example plot below). In order to make the code simple to port to other operating systems it contains the compiler switch DO_PLOTTING which has to be turned off in order to turn the code into a console application. The code uses hungarian prefix notation to make the data types used more explicit and hence the code hopefully more clear.



```
No. of trials:  92200          Iteration:    460     best OFUNC value: 5.43714571 ı          DE-strategy No.: 5
```

There are two ways to work with DeWin:

1) As a Windows application under Microsoft Windows ®. For example, in Visual C++ you have to generate a workspace of type "Win32Application". The compiler switch "DO_PLOTTING" should be defined. Then insert all .cpp files into the project and go to Build | Rebuild All to obtain an executable code. Note that the plotting features are working only for the Win32 environment.

2) As a Windows console application under Microsoft Windows ®. Generate a workspace of type "Win32ConsoleApplication". Make sure the compiler switch "DO_PLOTTING" is undefined. Then insert all .cpp files into the project and go to Build | Rebuild All to obtain an executable code. With this source code configuration the code should be portable to other operation systems with only minor modifications.

In contrast to older versions of DE-C-Code (see below) this one has been enhanced to support incorporation of constraints (bounds, equality, inequality). More details and more sample functions for the code can be found in the above book.

Another C-Version provided by Peter Turney updated an older version of the C-code above to make it more compatible to Visual C++ 6.0 and also to Unix-based C-compilers.

MATLAB ® Code

The latest MATLAB ® code from the book Differential Evolution - A Practical Approach to Global Optimization is available here by courtesy of Springer publisher. See an example plot below.

The code is designed to incorporate bounds, inequality, and equality constraints. The above book contains a detailed explanation of the code and some more examples in the CD companion. However, the code for download here contains the main engine in its full functionality for experimentation. Special attention has been given to coding conventions in hungarian prefix notation to make the programs easier to understand and use.

The script file Rundeopt.m (Run DE optimization) is the main control file in the MATLAB ® environment. Plotting can be turned off by setting the variable I_plotting=0 in rundeopt.m. Per default this variable is set to 1.

This is some older DE-Code in MATLAB ® which may still be interesting to some users. The m-files necessary to run the DE-demo in MATLAB ® are the demo shell dedemov.m, the actual DE-minimizer devec.m and the plot-supporting function xplt.m. If you want to use the DE-strategy in MATLAB ® for your own purposes it is better to use devec3.m which minimizes an externally defined objective function, e.g. rosen.m. Use the help function in MATLAB ® to obtain the details on how to use devec3.m. Two helper files, run1.m and run2.m show you how to work with devec3.m more conveniently.

Scilab Code

There is also a Scilab version of DE written by Walter Di Carlo, Helmut Jarausch, and recently updated and improved by Rainer Storn for Scilab 5.4.1. Scilab is a public domain clone of MATLAB ®.

PSather Code

DE is perfectly suited for parallel computation which already has found an implementation in PSather, a parallel object oriented language being developed at ICSI. The code was developed by Claudio Fleiner.

C++ Code

You can also get a completely object oriented C++ version written by Lester Godwin in Visual C++ 5.0. It is available as devcpp.zip and implements a DE-version which uses a single array rather than two for the population vectors. The single-array version does not lend itself for parallel computation but is a little more greedy than the two-array version. Another C++ version (VC++ 2010) has been contributed by Charles Brauer. There is also a great and fairly recent C++ contribution by Adrian Michel called the Differential Evolution C++ library.

Fortran90 Code

Another contribution is a Fortran90 version of DE developed by Prof. Feng-Sheng Wang and his students.

Python Code

A recent addition is a Python version of DE.

Labview Code

The latest contribution is a Labview version of DE developed by Franz Josef Ahlers.

R Code

Also available is now the R version of DE developed by David Ardia. Some further references to the usage of R can be found in the Journal of Statistical Software and on The R Journal.

Pascal Code

The Pascal Code of DE has been contributed by Hubert Geldon and Piotr A. Gauden.

C# Code

The C# Code of DE has been contributed by K.R. Shivaram .

Maple Code

The Maple Code of DE has been contributed by Dimitry Novikov .

Demo-Programs

If you happen to have access to a DOS-PC and have the graphics driver egavga.bgi from Borland's C++ library (version 3.1) you have all the requirements to get two little demo programs running which show DE at work. These programs are dedemo.exe and dedemo2.exe (fetch them via SHIFT+"Click" in Netscape) along with their data files demo.dat and demo2.dat. After placing these files in the pertinent directory, all you have to do is type

dedemo demo.dat

which shows you the polynomial fitting problem, or

dedemo2 demo2.dat

which visualizes DE working on Rosenbrock's saddle.

You can even manipulate the .dat files to experiment with different parameter values. Note however, that the demo programs are not crash proof, i.e. if you exceed certain limits for the parameters the programs will crash. So it's best to change e.g. only the number of parents NP < 200, weighting constant F ex [0,1] and crossing over factor CR ex [0,1] but NOT to change the number of parameters D. All the other parameters might be safely changed if the parameters are > 0. But remember: "everything that is free comes with no guarantee".

Applications of DE

A selection of scientific or commercial applications of DE which are accessible by a URL are listed below. It has to be noted that more than ten years from the inception of DE this list is impossible to keep complete.A simple google search for the keyword "Differential Evolution" shows that the selection below provides just a few applications of DE.

1) Open Source Optimizer "PyGMO" of the European Space Agency.

2) General Optimization Framework "Mystic" by Mike McKerns, Caltech.

3) Multiprocessor synthesis.

4) Neural network learning.

5) Chrystallographic characterization.

6) Synthesis of modulators.

7) Heat transfer parameter estimation in a trickle bed reactor.

8) Scenario-Integrated Optimization of Dynamic Systems.

9) Optimal Design of Shell-and-Tube Heat Exchangers.

10) Optimization of an Alkylation Reaction.

11) Optimization of Thermal Cracker Operation.

12) Optimization of Non-Linear Chemical Processes.

13) Optimum planning of cropping patterns.

14) Optimization of Water Pumping System .

15) Optimal Design of Gas Transmission Network .

16) Differential Evolution for Multi-Objective Optimization .

17) Physiochemistry of Carbon Materials.

18) Radio Network Design.

19) Reflectivity Curve Simulation.

Commercial SW using DE

1) Built in optimizer in MATHEMATICA's function Nminimize (since version 4.2).

2) MATLAB's GA toolbox contains a variant of DE.

3) Digital Filter Design.

4) Diffraction grating design.

5) Electricity market simulation.

6) Auto2Fit.

7) LMS Virtual Lab Optimization.

8) Optimization of optical systems.

9) Finite Element Design.

10) LabView.

11) Microwave Office 10.0 by AWR Corp..

12) OPTIMUS.

Academic SW-packages using DE

1) PyGMO.

2) Mystic.

3) RLaB.

4) NGSPICE.

5) Evolutionary Algorithms Workbench EVA2.

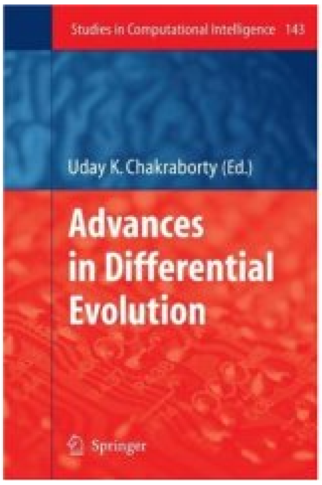
6) OpenOpt.

Other links, books, ...

Louni Lampinen's DE-bibliography.

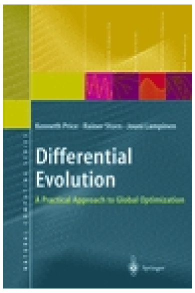
The Differential Evolution Society.

Available Now !


Studies in Computational Intelligence 143
Uday K. Chakraborty (Ed.)
Advances in Differential Evolution
Springer

Order the book via IN ASSOCIATION WITH amazon.com

The latest book about DE, "Advances in Differential Evolution (Studies in Computational Intelligence)", by editor Uday Chakraborty seeks to present a comprehensive study of the state of the art in this technology and also directions for future research. The fourteen chapters of this book have been written by leading experts in the area. The first seven chapters focus on algorithm design, while the last seven describe real-world applications. Chapter 1 introduces the basic differential evolution (DE) algorithm and presents a broad overview of the field. Chapter 2 presents a new, rotationally invariant DE algorithm. The role of self-adaptive control parameters in DE is investigated in Chapter 3. Chapters 4 and 5 address constrained optimization; the former develops suitable stopping conditions for the DE run, and the latter presents an improved DE algorithm for problems with very small feasible regions. A novel DE algorithm, based on the concept of "opposite" points, is the topic of Chapter 6. Chapter 7 provides a survey of multi-objective differential evolution algorithms. A review of the major application areas of differential evolution is presented in Chapter 8. Chapter 9 discusses the application of differential evolution in two important areas of applied electromagnetics. Chapters 10 and 11 focus on applications of hybrid DE algorithms to problems in power system optimization. Chapter 12 applies the DE algorithm to computer chess. The use of DE to solve a problem in bioprocess engineering is discussed in Chapter 13. Chapter 14 describes the application of hybrid differential evolution to a problem in control engineering.



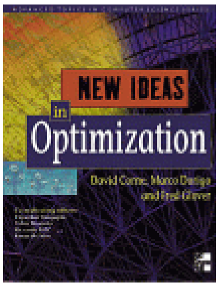
Order the book via IN ASSOCIATION WITH amazon.com or via amazon.co.uk or via

The book "Differential Evolution - A Practical Approach to Global Optimization" by Ken Price, Rainer Storn, and Jouni Lampinen (Springer, ISBN: 3-540-20950-6) provides the latest findings concerning DE. DE is a practical approach to global numerical optimization that is easy to understand, simple to implement, reliable, and fast. Packed with illustrations, computer code, new insights, and practical advice, this volume explores Differential Evolution in both principle and practice. It is a valuable resource for professionals needing a proven optimizer. A companion CD includes the latest DE-based optimization software in several programming languages (C, C++, MATLAB ®, Mathematica, Java, Fortran90, Scilab, Labview). The C and MATLAB ® versions are enhanced with code for constrained and multiobjective optimization. The C, MATLAB ®, Mathematica, and Java versions come with animated graphics support. See the table of contents for more details.

Studies in Fuzziness and Soft Computing
New Optimization Techniques in Engineering
Springer

Order via IN ASSOCIATION WITH amazon.com

Another book - "New Optimization Techniques in Engineering" - has been published by Springer in 2004 and contains information about the recent developments of Differential Evolution, especially concerning applications in the combinatorial problem domain.


NEW IDEAS in Optimization
David Corne, Marco Dorigo and Fred Glover

Order via amazon.co.uk.

The book "New Ideas in Optimization" by McGraw-Hill contains a large section about Differential Evolution. It is demonstrated that DE is not only a powerful function optimizer but can also be used for mixed integer programming and game strategy optimization. Other topics are Ant Colony Optimization, Immune System Methods, Memetic Algorithms, Particle Swarms (which is similiar to Differential Evolution), and others.

Contributors to this page

Franz Josef Ahlers, Walter Di Carlo, Claudio Fleiner, Lester Godwin, Mick (Mikal Keenan), Rituraj Deb Nath, Arnold Neumaier, James R. Phillips, Kenneth Price, Rainer Storn , Peter Turney, Feng-Sheng Wang, Jim Van Zandt, Hubert Geldon, Piotr A. Gauden, Charles Brauer, K.R. Shivaram, Dimitry Novikov

.

Rainer Storn's Home Page