outDEoptim <- DEoptim(Rosenbrock, lower, upper, DEoptim.control(NP = 80,

itermax = 400, F = 1.2, CR = 0.7))

outDEoptim <- DEoptim(Wild, lower = -50, upper = 50,

[**control**](http://inside-r.org/r-doc/boot/control) = DEoptim.control([**trace**](http://inside-r.org/r-doc/base/trace) = **FALSE**))

oneCore <- [**system.time**](http://inside-r.org/r-doc/base/system.time)( DEoptim(fn=Genrose, lower=[**rep**](http://inside-r.org/r-doc/base/rep)(-25, n), upper=[**rep**](http://inside-r.org/r-doc/base/rep)(25, n),

[**control**](http://inside-r.org/r-doc/boot/control)=[**list**](http://inside-r.org/r-doc/base/list)(NP=10\*n, itermax=maxIt)))

withParallel <- [**system.time**](http://inside-r.org/r-doc/base/system.time)( DEoptim(fn=Genrose, lower=[**rep**](http://inside-r.org/r-doc/base/rep)(-25, n), upper=[**rep**](http://inside-r.org/r-doc/base/rep)(25, n),

[**control**](http://inside-r.org/r-doc/boot/control)=[**list**](http://inside-r.org/r-doc/base/list)(NP=10\*n, itermax=maxIt, parallelType=1)))

DEoptim(Wild, lower = -50, upper = 50,

[**control**](http://inside-r.org/r-doc/boot/control) = DEoptim.control(NP = 50))

**Usage**

DEoptim.control(VTR = -Inf, strategy = 2, bs = FALSE, NP = NA,

itermax = 200, CR = 0.5, F = 0.8, trace = TRUE, initialpop = NULL,

storepopfrom = itermax + 1, storepopfreq = 1, p = 0.2, c = 0, reltol,

steptol, parallelType = 0, packages = c(), parVar = c(),

foreachArgs = list())

**Arguments**

VTR

the value to be reached. The optimization process will stop if either the maximum number of iterationsitermax is reached or the best parameter vector bestmem has found a value fn(bestmem) <= VTR. Default to -**Inf**.

strategy

defines the Differential Evolution strategy used in the optimization procedure:  
1: DE / rand / 1 / bin (classical strategy)  
2: DE / local-to-best / 1 / bin (default)  
3: DE / best / 1 / bin with jitter  
4: DE / rand / 1 / bin with per-vector-dither  
5: DE / rand / 1 / bin with per-generation-dither  
6: DE / current-to-p-best / 1  
any value not above: variation to DE / rand / 1 / bin: either-or-algorithm. Default strategy is currently 2. See \*Details\*.

bs

if **FALSE** then every mutant will be tested against a member in the previous generation, and the best value will proceed into the next generation (this is standard trial vs. target selection). If **TRUE** then the old generation and NP mutants will be sorted by their associated objective function values, and the best NPvectors will proceed into the next generation (best of parent and child selection). Default is **FALSE**.

NP

number of population members. Defaults to **NA**; if the user does not change the value of NP from **NA** or specifies a value less than 4 it is reset when [DEoptim](http://inside-r.org/packages/cran/DEoptim) is called as 10\*[**length**](http://inside-r.org/r-doc/base/length)(lower). For many problems it is best to set NP to be at least 10 times the length of the parameter vector.

itermax

the maximum iteration (population generation) allowed. Default is 200.

CR

crossover probability from interval [0,1]. Default to 0.5.

F

differential weighting factor from interval [0,2]. Default to 0.8.

trace

Positive integer or logical value indicating whether printing of progress occurs at each iteration. The default value is **TRUE**. If a positive integer is specified, printing occurs every [**trace**](http://inside-r.org/r-doc/base/trace) iterations.

initialpop

an initial population used as a starting population in the optimization procedure. May be useful to speed up the convergence. Default to **NULL**. If given, each member of the initial population should be given as a row of a numeric matrix, so that initialpop is a matrix with NP rows and a number of columns equal to the length of the parameter vector to be optimized.

storepopfrom

from which generation should the following intermediate populations be stored in memory. Default toitermax + 1, i.e., no intermediate population is stored.

storepopfreq

the frequency with which populations are stored. Default to 1, i.e., every intermediate population is stored.

p

when strategy = 6, the top (100 \* p)% best solutions are used in the mutation. p must be defined in (0,1].

c

[**c**](http://inside-r.org/r-doc/base/c) controls the speed of the crossover adaptation. Higher values of [**c**](http://inside-r.org/r-doc/base/c) give more weight to the current successful mutations. [**c**](http://inside-r.org/r-doc/base/c) must be defined in (0,1].

reltol

relative convergence tolerance. The algorithm stops if it is unable to reduce the value by a factor ofreltol \* ([**abs**](http://inside-r.org/r-doc/base/abs)(val) +    reltol) after steptol steps. Defaults to[**sqrt**](http://inside-r.org/r-doc/base/sqrt)([**.Machine**](http://inside-r.org/r-doc/base/.Machine)$double.eps), typically about 1e-8.

steptol

see reltol. Defaults to itermax.

parallelType

Defines the type of parallelization to employ, if any.  : The default, this uses [DEoptim](http://inside-r.org/packages/cran/DEoptim) one only one core.1: This uses all available cores, via the parallel package, to run [DEoptim](http://inside-r.org/packages/cran/DEoptim). If parallelType=1, then thepackages argument and the parVar argument need to specify the packages required by the objective function and the variables required in the environment, respectively. 2: This uses the foreach package for parallelism; see the sandbox directory in the source code for examples. If parallelType=1, then theforeachArgs argument can pass the options to be called with foreach.

packages

Used if parallelType=1; a list of package names (as strings) that need to be loaded for use by the objective function.

parVar

Used if parallelType=1; a list of variable names (as strings) that need to exist in the environment for use by the objective function or are used as arguments by the objective function.

foreachArgs

A list of named arguments for the foreach function from the package foreach. The arguments i,.combine and .export are not possible to set here; they are set internally.

**Details**

This defines the Differential Evolution strategy used in the optimization procedure, described below in the terms used by Price et al. (2006); see also Mullen et al. (2009) for details.

* strategy = 1: DE / rand / 1 / bin.   
  This strategy is the classical approach for DE, and is described in [DEoptim](http://inside-r.org/packages/cran/DEoptim).
* strategy = 2: DE / local-to-best / 1 / bin.   
  In place of the classical DE mutation the expression is used, where old\_i,g and best\_g are the i-th member and best member, respectively, of the previous population. This strategy is currently used by default.
* strategy = 3: DE / best / 1 / bin with jitter.  
  In place of the classical DE mutation the expression is used, where jitter is defined as 0.0001 \* rand + F.
* strategy = 4: DE / rand / 1 / bin with per vector dither.  
  In place of the classical DE mutation the expression is used, where dither is calculated as F + \code{rand} \* (1 - F).
* strategy = 5: DE / rand / 1 / bin with per generation dither.  
  The strategy described for 4 is used, but dither is only determined once per-generation.
* strategy = 6: DE / current-to-p-best / 1.  
  The top (100\*p) percent best solutions are used in the mutation, where p is defined in (0,1].
* any value not above: variation to DE / rand / 1 / bin: either-or algorithm.  
  In the case that rand < 0.5, the classical strategy strategy = 1 is used. Otherwise, the expression is used.

Several conditions can cause the optimization process to stop:

* if the best parameter vector (bestmem) produces a value less than or equal to VTR (i.e. fn(bestmem)<= VTR), or
* if the maximum number of iterations is reached (itermax), or
* if a number (steptol) of consecutive iterations are unable to reduce the best function value by a certain amount (reltol \*       ([**abs**](http://inside-r.org/r-doc/base/abs)(val) + reltol)). 100\*reltol is approximately the percent change of the objective value required to consider the parameter set an improvement over the current best member.

Zhang and Sanderson (2009) define several extensions to the DE algorithm, including strategy 6, DE/current-to-p-best/1. They also define a self-adaptive mechanism for the other control parameters. This self-adaptation will speed convergence on many problems, and is defined by the control parameter [**c**](http://inside-r.org/r-doc/base/c). If [**c**](http://inside-r.org/r-doc/base/c) is non-zero, crossover and mutation will be adapted by the algorithm. Values in the range of [**c**](http://inside-r.org/r-doc/base/c)=.05 to [**c**](http://inside-r.org/r-doc/base/c)=.5appear to work best for most problems, though the adaptive algorithm is robust to a wide range of [**c**](http://inside-r.org/r-doc/base/c).