

Package ‘vntrs’

October 25, 2025

Title Variable Neighborhood Trust Region Search

Version 0.2.0

Description An algorithm for nonlinear global optimization based on the variable neighbourhood trust region search (VNTRS) algorithm proposed by Bierlaire et al. (2009) ``A Heuristic for Nonlinear Global Optimization" <[doi:10.1287/ijoc.1090.0343](https://doi.org/10.1287/ijoc.1090.0343)>. The algorithm combines variable neighbourhood exploration with a trust-region framework to efficiently search the solution space. It can terminate a local search early if the iterates are converging toward a previously visited local optimum or if further improvement within the current region is unlikely. In addition to global optimization, the algorithm can also be applied to identify multiple local optima.

URL <https://loelschlaeger.de/vntrs/>,
<https://github.com/loelschlaeger/vntrs/>

BugReports <https://github.com/loelschlaeger/vntrs/issues>

License GPL-3

Encoding UTF-8

RoxygenNote 7.3.3

Imports checkmate, oeli (>= 0.7.5), Rcpp

LinkingTo Rcpp, RcppArmadillo

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

Author Lennart Oelschläger [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-5421-9313>>)

Maintainer Lennart Oelschläger <oelschlaeger.lennart@gmail.com>

Repository CRAN

Date/Publication 2025-10-25 07:30:02 UTC

Contents

vntrs 2

Index**4**

vntrs

*Variable neighborhood trust region search***Description**

This function performs variable neighborhood trust region search.

Usage

```
vntrs(
  f,
  npar,
  minimize = TRUE,
  init_runs = 5L,
  init_min = -1,
  init_max = 1,
  init_iterlim = 20L,
  neighborhoods = 5L,
  neighbors = 5L,
  beta = 0.05,
  iterlim = 100L,
  tolerance = 1e-06,
  inferior_tolerance = 1e-06,
  time_limit = NULL,
  cores = 1L,
  lower = NULL,
  upper = NULL,
  collect_all = FALSE,
  quiet = TRUE
)
```

Arguments

f	[function] A function, returning either • a numeric objective value or • a list with element value and optional gradient and hessian components for a numeric parameter vector. Missing derivatives are approximated by finite differences.
npar	[integer(1)] The number of parameters of f.
minimize	[logical(1)] Minimize f?

init_runs	[integer(1)] Number of random starting points for the initialization stage.
init_min, init_max	[numeric(1)] Lower and upper bound for the uniform sampling range during initialization.
init_iterlim	[integer(1)] Maximum iterations of the trust-region method during initialization.
neighborhoods	[integer(1)] Number of neighborhood expansions to perform.
neighbors	[integer(1)] Number of neighboring points drawn in each neighborhood.
beta	[numeric(1)] Non-negative scaling factor that controls the neighborhood expansion.
iterlim	[integer(1)] Maximum iterations of the trust-region method during the main search.
tolerance	[numeric(1)] Minimum distance between optima candidates to consider them distinct.
inferior_tolerance	[numeric(1)] Maximum allowed difference from the best known objective value when deciding if a local optimum should be discarded early.
time_limit	[integer(1) NULL] Optional time limit (in seconds) for the search. If reached, the search stops early with a warning.
cores	[integer(1)] Number of CPU cores used for parallel evaluation.
lower, upper	[numeric(npar) NULL] Optional lower and upper parameter bounds.
collect_all	[logical(1)] Keep every converged local optimum even if it is inferior to the best known solution and disable early stopping?
quiet	[logical(1)] Suppress messages?

Value

A data.frame summarizing the identified optima or NULL if none could be determined.

References

Bierlaire et al. (2009) "A Heuristic for Nonlinear Global Optimization" [doi:10.1287/ijoc.1090.0343](https://doi.org/10.1287/ijoc.1090.0343).

Examples

```
rosenbrock <- function(x) 100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2
vntrs(f = rosenbrock, npar = 2)
```

Index

vntrs, [2](#)