

Package ‘mco’

April 17, 2009

Version 1.0.4

Date 2008-10-14

Title Multi criteria optimization algorithms and related functions

Description Functions for multi criteria optimization using genetic algorithms and related test problems

Author Heike Trautmann <trautmann@statistik.uni-dortmund.de> and Detlef Steuer <detlef.steuer@hsu-hamburg.de> and Olaf Mersmann <olafm@statistik.uni-dortmund.de>

Maintainer Olaf Mersmann <olafm@statistik.uni-dortmund.de>

Depends R (>= 2.7.0)

Suggests scatterplot3d

License GPL-2

URL <http://r-forge.r-project.org/projects/desire>

LazyData yes

Repository CRAN

Date/Publication 2008-10-15 06:51:43

R topics documented:

generationalDistance	2
normalizeFront	3
nsga2	4
paretoFront	5

Index	7
--------------	----------

`generationalDistance`*Quality measures for MCO solutions*

Description

Functions to evaluate the quality of the estimated pareto front.

Usage

```
generationalDistance(x, o)
generalizedSpread(x, o)
dominatedHypervolume(x, ref)
```

Arguments

<code>x</code>	Estimated pareto front or an object which has a <code>paretoFront</code> method
<code>o</code>	True pareto front or an object which has a <code>paretoFront</code> method
<code>ref</code>	Reference point (may be omitted).

Details

Instead of the pareto front, one can also pass an object for which a `paretoFront` method exists to both methods.

For `dominatedHypervolume`, if no reference point is given, the maximum in each dimension is used as the reference point.

Value

The respective quality measure.

Author(s)

Heike Trautmann [⟨trautmann@statistik.uni-dortmund.de⟩](mailto:trautmann@statistik.uni-dortmund.de), Detlef Steuer [⟨steuer@hsu-hamburg.de⟩](mailto:steuer@hsu-hamburg.de)
and Olaf Mersmann [⟨olafm@statistik.uni-dortmund.de⟩](mailto:olafm@statistik.uni-dortmund.de)

References

Carlos M. Fonseca, Luis Paquete, and Manuel Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In IEEE Congress on Evolutionary Computation, pages 1157-1163, Vancouver, Canada, July 2006.

Examples

```
sch1 <- function(x) c(x^2, (x-2)^2)

## Estimate true front:
tf <- nsga2(sch1, 1, 2,
           lower.bounds=0, upper.bounds=1,
           popsize=1000, generations=20)
res <- nsga2(sch1, 1, 2,
           lower.bounds=0, upper.bounds=1,
           popsize=16, generations=c(2, 4, 6, 8, 10, 20, 50))
n <- length(res)
sapply(1:n, function(i) generationalDistance(res[[i]], tf))
sapply(1:n, function(i) generalizedSpread(res[[i]], tf))
```

normalizeFront	<i>Normalize a pareto front</i>
----------------	---------------------------------

Description

Rescales a pareto front to be in the unit hypercube

Usage

```
normalizeFront(front, minval, maxval)
```

Arguments

front	Matrix containing the pareto front
minval	Vector containing the minimum value of each objective. May be omitted.
maxval	Vector containing the maximum value of each objective. May be omitted.

Value

Matrix containing the rescaled pareto front.

Author(s)

Heike Trautmann <trautmann@statistik.uni-dortmund.de>, Detlef Steuer <steuer@hsu-hamburg.de>
and Olaf Mersmann <olafm@statistik.uni-dortmund.de>

nsga2

NSGA II MOEA

Description

Multicriterion optimization algorithm

Usage

```
nsga2(fn, idim, odim, ..., constraints = NULL, cdim = 0, lower.bounds = rep(-Inf, i
```

Arguments

<code>fn</code>	Function
<code>idim</code>	Input dimension
<code>odim</code>	Output dimension
<code>...</code>	Arguments passed through to 'fn'
<code>constraints</code>	Constraint function
<code>cdim</code>	Constraint dimension
<code>lower.bounds</code>	Lower bound of input
<code>upper.bounds</code>	Upper bound of input
<code>popsize</code>	Size of population
<code>generations</code>	Number of generations to breed. If a vector, then the result will contain the population at each given generation.
<code>cprob</code>	Crossing probability
<code>cdist</code>	Crossing distribution index
<code>mprob</code>	Mutation probability
<code>mdist</code>	Mutation distribution index

Value

If `generation` is an integer, a list describing the final population with components `par`, `value` and `pareto.optimal`. If `generations` is a vector, a list is returned. The *i*-th element of the list contains the population after `generations[i]` generations.

Author(s)

Heike Trautmann <trautmann@statistik.uni-dortmund.de>, Detlef Steuer <steuer@hsu-hamburg.de> and Olaf Mersmann <olafm@statistik.uni-dortmund.de>

References

Deb, K., Pratap, A., and Agarwal, S.. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6** (8) (2002), 182-197.

Examples

```

## Binh 1 problem:
binh1 <- function(x) {
  y <- numeric(2)
  y[1] <- crossprod(x, x)
  y[2] <- crossprod(x - 5, x - 5)
  return (y)
}
r1 <- nsga2(binhl, 2, 2,
            generations=150, popsize=100,
            cprob=0.7, cdist=20,
            mprob=0.2, mdist=20,
            lower.bounds=rep(-5, 2),
            upper.bounds=rep(10, 2))
plot(r1)

## VNT problem:
vnt <- function(x) {
  y <- numeric(3)
  xn <- crossprod(x, x)
  y[1] <- xn/2 + sin(xn);
  y[2] <- (crossprod(c(3, -2), x) + 4)^2/8 + (crossprod(c(1, -1), x) + 1)^2/27 + 15
  y[3] <- 1/(xn + 1) - 1.1*exp(-xn)
  return (y)
}

r2 <- nsga2(vnt, 2, 3,
            generations=150, popsize=100,
            lower.bounds=rep(-3, 2),
            upper.bounds=rep(3, 2))
plot(r2)

```

paretoFront

Pareto Front and pareto set accessor

Description

Extract the pareto front or pareto set from an mco result object.

Filter an mco result and extract the pareto-optimal solutions.

Usage

```

paretoFront(x, ...)
paretoSet(x, ...)
paretoFilter(x, ...)

```

Arguments

x	matrix or mco result object
...	Ignored

Value

A matrix containing the pareto front or pareto set.

`paretoFilter` returns those values in x which are not dominated by any other solution.

Author(s)

Heike Trautmann [⟨trautmann@statistik.uni-dortmund.de⟩](mailto:trautmann@statistik.uni-dortmund.de), Detlef Steuer [⟨steuer@hsu-hamburg.de⟩](mailto:steuer@hsu-hamburg.de)
and Olaf Mersmann [⟨olafm@statistik.uni-dortmund.de⟩](mailto:olafm@statistik.uni-dortmund.de)

Index

*Topic **optimize**

nsga2, [4](#)

dominatedHypervolume
(*generationalDistance*), [2](#)

generalizedSpread
(*generationalDistance*), [2](#)

generationalDistance, [2](#)

normalizeFront, [3](#)

nsga2, [4](#)

paretoFilter (*paretoFront*), [5](#)

paretoFront, [5](#)

paretoSet (*paretoFront*), [5](#)