# Package: GBOP2 (via r-universe)

May 22, 2026

**Type** Package

**Title** Generalized Bayesian Optimal Phase II Design (G-BOP2)

**Version** 0.1.4

**Depends** R (>= 4.1.0)

**Maintainer** Wanni Lei <wanni.lei17@gmail.com>

**Description** Provides functions for implementing the Generalized Bayesian Optimal Phase II (G-BOP2) design using various Particle Swarm Optimization (PSO) algorithms, including: - PSO-Default, based on Kennedy and Eberhart (1995) <doi:10.1109/ICNN.1995.488968>, ``Particle Swarm Optimization''; - PSO-Quantum, based on Sun, Xu, and Feng (2004) <doi:10.1109/ICCIS.2004.1460396>, ``A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization''; - PSO-Dexp, based on Stehlík et al. (2024) <doi:10.1016/j.asoc.2024.111913>, ``A Double Exponential Particle Swarm Optimization with Non-Uniform Variates as Stochastic Tuning and Guaranteed Convergence to a Global Optimum with Sample Applications to Finding Optimal Exact Designs in Biostatistics''; - and PSO-GO.

**Imports** tidyr, R6, Rcpp, doParallel, foreach, dplyr, stats, globpso, parallel, utils

**Suggests** knitr, rmarkdown, roxygen2, testthat (>= 3.0.0), R.rsp

**LinkingTo** Rcpp, RcppArmadillo

**Config/parallel** false

**Config/testthat/edition** 3

**VignetteBuilder** R.rsp

**License** GPL-2

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** yes

**Author** Xinying Fang [aut], Wanni Lei [aut, cre], Shouhao Zhou [aut], Ping-Yang Chen [aut], Ray-Bing Chen [aut], Weng Kee Wong [aut], J. Jack Lee [aut]

**Repository** https://cran.r-universe.dev
**Date/Publication** 2025-12-22 13:00:09 UTC
**RemoteUrl** https://github.com/cran/GBOP2
**RemoteRef** HEAD
**RemoteSha** 3da9e291ae61b541d95b5d4ec4d01ed8ca734c00

# Contents



---

`GBOP2_maxP_dualE` *PSOGO: Power maximizing design with dual boundaries*

---

## Description

This function implements PSOGO to find a power maximizing design with dual boundaries.

## Arguments

| | |
|---|---|
| `design` | fixed as "optimal", which can not be modified by user |
| `nlooks` | number of interim looks |
| `p0` | Null hypothesis response rate |
| `p1` | Alternative hypothesis response rate |
| `err1` | Type I error rate |
| `nParallel` | number of pso ensemble |
| `minPower` | power |
| `totalPatients` | total patients |
| `Nmin_cohort1` | minimum number of first cohort |
| `Nmin_increase` | minimum number of increase in each cohort |
| `pso_method` | "all" for using three distinct pso, otherwise indicate single pso method |
| `seed` | seed for pso |
| `nSwarm` | nSwarm for pso |
| `maxIter` | maxIter for pso |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using 'init_cluster(nCore)' and 'stop_cluster()' manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
# GBOP2_maxP_dualE(
#   nlooks = 1,
#   p0 = 0.2,
#   p1 = 0.4,
#   err1 = 0.05,
#   minPower = 0.8,
#   totalPatients = 26,
#   Nmin_cohort1 = 10,
#   Nmin_increase = 5,
#   pso_method = "default",
#   nParallel = 3,
#   seed = 1024,
#   nSwarm = 64,
#   maxIter = 200
# )
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_singleE() manually for real optimization.")
```

---

# GBOP2_maxP_singleE *PSOGO: Power maximizing design with single boundary for futility*

---

## Description

This function implements PSOGO to find a power maximizing design with single boundary for futility.

## Usage

```
GBOP2_maxP_singleE(
  nlooks = 1,
  p0 = 0.2,
  p1 = 0.4,
```

```
  err1 = 0.05,
  minPower = 0.8,
  totalPatients = 5,
  Nmin_cohort1 = 1,
  Nmin_increase = 1,
  pso_method = "default",
  nParallel = 3,
  seed = 1024,
  nSwarm = 64,
  maxIter = 200
)
```

## Arguments

| | |
|---|---|
| nlooks | number of interim looks |
| p0 | Null hypothesis response rate |
| p1 | Alternative hypothesis response rate |
| err1 | Type I error rate |
| minPower | power |
| totalPatients | total number of patients |
| Nmin_cohort1 | minimum number of first cohort |
| Nmin_increase | minimum number of increase in each cohort |
| pso_method | "all" for using three distinct pso, otherwise indicate single pso method |
| nParallel | number of pso ensemble |
| seed | Random seed for reproducibility |
| nSwarm | nSwarm for pso |
| maxIter | maxIter for pso |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using ‘init_cluster(nCore)‘ and ‘stop_cluster()‘ manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
#   GBOP2_maxP_singleE(
#   nlooks = 1,
#   p0 = 0.2,
#   p1 = 0.4,
```

```
#   err1 = 0.05,
#   minPower = 0.8,
#   totalPatients = 26,
#   Nmin_cohort1 = 10,
#   Nmin_increase = 5,
#   pso_method = "default",
#   nParallel = 3,
#   seed = 1024,
#   nSwarm = 64,
#   maxIter = 200
# )
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_singleE() manually for real optimization.")
```

---

GBOP2_maxP_TE *PSOGO: Power maximizing design with efficacy and toxicity boundaries*

---

## Description

This function implements PSOGO to find a power maximizing design with efficacy and toxicity boundaries.

## Arguments

| | |
|---|---|
| design | fixed as "optimal", cannot be modified by user |
| pso_method | method for single PSO, choose from "default", "quantum" or "dexp" |
| nlooks | number of interim looks |
| skip_efficacy | default is NULL, indicate skip efficacy as 1 and not skip as 0 in a vector |
| skip_toxicity | default is NULL, indicate skip toxicity as 1 and not skip as 0 in a vector |
| totalPatients | number of total patients |
| Nmin_cohort1 | maximum number of patients |
| Nmin_increase | minimum number of first cohort |
| p01 | H0 for efficacy |
| p02 | H0 for toxicity |
| p03 | H0 for Eff and Tox |
| p11 | H1 for efficacy |
| p12 | H1 for toxicity |
| p13 | H1 for Eff and Tox |
| err_eff | Type I error rate: Efficacious but toxic |

| | |
|---|---|
| err_tox | Type I error rate: Safe but futile |
| err_all | Type I error rate: Futile and toxic |
| power_all | power: Futile and toxic |
| nSwarm | nSwarm in PSO |
| maxIter | maxIter in PSO |
| nParallel | number of PSO ensemble |
| seed | Random seed for reproducibility |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using 'init_cluster(nCore)' and 'stop_cluster()' manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
# GBOP2_maxP_TE(
# design = "optimal",
# nlooks = 1,
# skip_efficacy = NULL,
# skip_toxicity = NULL,
# totalPatients = 50,
# Nmin_cohort1 = 10,
# Nmin_increase = 5,
# p01 = 0.15,
# p02 = 0.16,
# p03 = 0.024,
# p11 = 0.4,
# p12 = 0.08,
# p13 = 0.032,
# err_eff = 1,
# err_tox = 1 ,
# err_all = 0.1,
# power_all = 0.8,
# nParallel = 3,
# seed = 5321,
# pso_method = "default",
# nSwarm = 32,
# maxIter = 100)
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_singleE() manually for real optimization.")
```

GBOP2_minSS_dualE *PSOGO: Optimal/Minimax design with dual boundaries*

## Description

This function implements PSOGO to find an optimal or minimax design with dual boundaries.

## Usage

```
GBOP2_minSS_dualE(
  design = "optimal",
  unified.u = unified.u,
  weight = 1,
  nlooks = 1,
  p0 = 0.2,
  p1 = 0.4,
  err1 = 0.05,
  minPower = 0.8,
  maxPatients = 5,
  Nmin_cohort1 = 1,
  Nmin_increase = 1,
  pso_method = "default",
  nParallel = 3,
  seed = 123,
  nSwarm = 64,
  maxIter = 200
)
```

## Arguments

| | |
|---|---|
| design | choose from "optimal", "minimax", or "unified" |
| unified.u | specify when design = "unified", u in zero to one |
| weight | weight of sample size under null |
| nlooks | number of interim looks |
| p0 | Null hypothesis response rate |
| p1 | Alternative hypothesis response rate |
| err1 | Type I error rate |
| minPower | power |
| maxPatients | maximum number of patients |
| Nmin_cohort1 | minimum number of first cohort |
| Nmin_increase | minimum number of increase in each cohort |
| pso_method | "all" for using three distinct pso, otherwise indicate single pso method |

| | |
|---|---|
| nParallel | number of pso ensemble |
| seed | seed for pso |
| nSwarm | nSwarm for pso |
| maxIter | maxIter for pso |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using 'init_cluster(nCore)' and 'stop_cluster()' manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
#  GBOP2_minSS_dualE(
#    design = "optimal",
#    unified.u = unified.u,
#    nlooks = 1,
#    p0 = 0.2,
#    p1 = 0.4,
#    err1 = 0.05,
#    minPower = 0.8,
#    weight = 1,
#    maxPatients = 25,
#    Nmin_cohort1 = 10,
#    Nmin_increase = 5,
#    pso_method = "default",
#    nParallel = 3,
#    seed = 123,
#    nSwarm = 64,
#    maxIter = 200
#  )
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_dualE() manually for real optimization.")
```

GBOP2_minSS_singleE *PSOGO: Optimal/Minimax design with single boundary for futility*

## Description

This function implements PSOGO to find an optimal or minimax design with single boundary for futility.

## Usage

```
GBOP2_minSS_singleE(
  design = "optimal",
  unified.u = 1,
  weight = 1,
  nlooks = 2,
  p0 = 0.2,
  p1 = 0.4,
  err1 = 0.05,
  minPower = 0.8,
  maxPatients = 5,
  Nmin_cohort1 = 1,
  Nmin_increase = 1,
  pso_method = "default",
  nParallel = 3,
  seed = 456,
  nSwarm = 64,
  maxIter = 200
)
```

## Arguments

| | |
|---|---|
| design | choose from "optimal", "minimax", or "unified" |
| unified.u | specify when design = "unified", u in zero to one |
| weight | weight of sample size under null |
| nlooks | number of interim looks |
| p0 | Null hypothesis response rate |
| p1 | Alternative hypothesis response rate |
| err1 | Type I error rate |
| minPower | power |
| maxPatients | maximum number of patients |
| Nmin_cohort1 | minimum number of first cohort |
| Nmin_increase | minimum number of increase in each cohort |
| pso_method | "all" for using three distinct pso, otherwise indicate single pso method |

| | |
|---|---|
| nParallel | number of pso ensemble |
| seed | Random seed for reproducibility |
| nSwarm | nSwarm for pso |
| maxIter | maxIter for pso |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using 'init_cluster(nCore)' and 'stop_cluster()' manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
#  GBOP2_minSS_singleE(
#   design = "optimal",
#    unified.u = 1,
#    nlooks = 1,
#    p0 = 0.2,
#    p1 = 0.4,
#    err1 = 0.05,
#    minPower = 0.8,
#    weight = 1,
#    maxPatients = 25,
#    Nmin_cohort1 = 10,
#    Nmin_increase = 5,
#    pso_method = "default",
#    nParallel = 3,
#    seed = 1024,
#    nSwarm = 64,
#    maxIter = 200
#  )
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_singleE() manually for real optimization.")
```

---

GBOP2_minSS_TE *PSOGO: Optimal/Minimax design with efficacy and toxicity boundaries*

---

## Description

This function implements PSOGO to find an optimal or minimax design with efficacy and toxicity boundaries.

## Usage

```
GBOP2_minSS_TE(
  design = "optimal",
  unified.u = 1,
  nlooks = 1,
  skip_efficacy = NULL,
  skip_toxicity = NULL,
  maxPatients = 26,
  Nmin_cohort1 = 13,
  Nmin_increase = 13,
  p01 = 0.3,
  p02 = 0.4,
  p03 = 0.2,
  p11 = 0.6,
  p12 = 0.2,
  p13 = 0.15,
  err_eff = 0.1,
  err_tox = 0.1,
  err_all = 0.05,
  power_all = 0.8,
  pso_method = "all",
  nParallel = 3,
  seed = 1324,
  nSwarm = 32,
  maxIter = 100
)
```

## Arguments

| | |
|---|---|
| design | choose from "optimal", "minimax", or "unified" |
| unified.u | specify when design = "unified", u in zero to one |
| nlooks | number of interim looks |
| skip_efficacy | default is NULL, indicate skip efficacy as 1 and not skip as 0 in a vector |
| skip_toxicity | default is NULL, indicate skip toxicity as 1 and not skip as 0 in a vector |
| maxPatients | maximum number of patients |
| Nmin_cohort1 | minimum number of first cohort |
| Nmin_increase | minimum number of increase in each cohort |
| p01 | H0 for efficacy |
| p02 | H0 for toxicity |
| p03 | H0 for Eff and Tox |

| | |
|---|---|
| p11 | H1 for efficacy |
| p12 | H1 for toxicity |
| p13 | H1 for Eff and Tox |
| err_eff | Type I error rate: Efficacious but toxic |
| err_tox | Type I error rate: Safe but futile |
| err_all | Type I error rate: Futile and toxic |
| power_all | power: Futile and toxic |
| pso_method | "all" for using three distinct pso, otherwise indicate single pso method |
| nParallel | number of pso ensemble |
| seed | Random seed for reproducibility |
| nSwarm | nSwarm in PSO |
| maxIter | maxIter in PSO |

## Details

Parallel computing is only used when the user explicitly sets nCore > 1. No more than 2 cores should be used unless the user is aware and permits it. The function defaults to sequential execution. If multiple analyses are planned, consider using 'init_cluster(nCore)' and 'stop_cluster()' manually to control the backend.

## Value

A list on design parameters and operating characteristics

## Examples

```
# init_cluster(2)
#  GBOP2_minSS_TE(
#   design = "optimal",
#    unified.u = 1,
#    nlooks = 1,
#    skip_efficacy = NULL,
#    skip_toxicity = NULL,
#    maxPatients = 25,
#    Nmin_cohort1 = 10,
#    Nmin_increase = 5,
#    p01 = 0.3,
#    p02 = 0.4,
#    p03 = 0.2,
#    p11 = 0.6,
#    p12 = 0.2,
#    p13 = 0.15,
#    err_eff = 0.1,
#    err_tox = 0.1,
#    err_all = 0.05,
#    power_all = 0.8,
#    pso_method = "default",
#    nParallel = 3,
```

```
#    seed = 5321,
#    nSwarm = 64,
#    maxIter = 100
#  )
# stop_cluster()  # Only if init_cluster() was used
#
message("Run GBOP2_minSS_singleE() manually for real optimization.")
```

---

`get_cluster` *Get current cluster*

---

## Description

Returns the current parallel cluster object, if initialized.

## Usage

```
get_cluster()
```

## Value

A cluster object or NULL.

---

`init_cluster` *Initialize parallel cluster*

---

## Description

Creates and registers a parallel backend using the specified number of cores. Falls back to sequential execution if nCore <= 1.

## Usage

```
init_cluster(nCore = 2)
```

## Arguments

| | |
|---|---|
| `nCore` | Number of cores to use (default is 2). |

## stop_cluster *Stop and clean up the cluster*

### Description

Stops the currently running parallel cluster and reverts to sequential execution.

### Usage

```
stop_cluster()
```

## summary.gbop2 *Summary function Summary function for gbop2 objects*

### Description

Summary function Summary function for gbop2 objects

### Usage

```
## S3 method for class 'gbop2'
summary(object, ...)
```

### Arguments

| | |
|---|---|
| object | GBOP2_maxP_dualE GBOP2_maxP_singleE GBOP2_maxP_TE GBOP2_minSS_dualE GBOP2_minSS_singleE GBOP2_minSS_TE |
| ... | ignored arguments |

### Value

A summary table

### Examples

```
design <- GBOP2_minSS_singleE(
  design = "optimal",
  unified.u = 1,
  nlooks = 1,
  p0 = 0.2,
  p1 = 0.4,
  err1 = 0.05,
  minPower = 0.8,
  weight = 1,
  maxPatients = 25,
  Nmin_cohort1 = 10,
```

```
  Nmin_increase = 5,
  pso_method = "default",
  nParallel = 1,
  seed = 1024,
  nSwarm = 64,
  maxIter = 200
)

summary(design)
```

# Index