Package ‘QPmin’

April 15, 2021

Version 0.5-1

Date 2021-03-29

Title Linearly Constrained Indefinite Quadratic Program Solver

Author Andrea Giusto

Maintainer Andrea Giusto <andrea.giusto@gmail.com>

Depends R (>= 3.1.0), Matrix, methods

Description Active set method solver for the solution of indefinite quadratic programs, subject to lower bounds on linear functions of the variables and simple bounds on the variables themselves. The function QPmin() implements an algorithm similar to the one described in Gould (1991) <doi:10.1093/imanum/11.3.299> with the exception that an efficient sparse internal representation of the basis matrix is maintained thus allowing the solution of somewhat large problems.

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-15 11:30:05 UTC

R topics documented:

`QPmin-package` *Linearly Constrained Indefinite Quadratic Program Solver*

Description

Active set method solver for the solution of indefinite quadratic programs, subject to lower bounds on linear functions of the variables and simple bounds on the variables themselves. The function QPmin() implements an algorithm similar to the one described in Gould (1991) <doi:10.1093/imanum/11.3.299> with the exception that an efficient sparse internal representation of the basis matrix is maintained thus allowing the solution of somewhat large problems.

Details

The DESCRIPTION file:

Package:	QPmin
Version:	0.5-1
Date:	2021-03-29
Title:	Linearly Constrained Indefinite Quadratic Program Solver
Author:	Andrea Giusto
Maintainer:	Andrea Giusto <andrea.giusto@gmail.com>
Depends:	R (>= 3.1.0), Matrix, methods
Description:	Active set method solver for the solution of indefinite quadratic programs, subject to lower bounds on linear fu
License:	GPL (>= 2)

Index of help topics:

`QPgen.internal.bilinear`	`QPgen.internal.bilinear`
`QPgen.internal.concave`	`QPgen.internal.concave`
`QPgen.internal.convex`	`QPgen.internal.convex`
`QPmin`	`QPmin`
`QPmin-package`	`Linearly Constrained Indefinite Quadratic Program Solver`
`randomQP`	`randomQP`

Active Set Method (ASM) for minimizinig indefinite quadratic programs subject to linear constraints: QPmin. The package also provide a set of additional functions for the random generation of specific quadratic programs.

Author(s)

Andrea Giusto

Maintainer: Andrea Giusto <andrea.giusto@gmail.com>

References

"A new technique for generating quadratic programming test problems," Calamai P.H., L.N. Vicente, and J.J. Judice, Mathematical Programming 61 (1993), pp. 215-231.

“A General Quadratic Programming Algorithm,” Fletcher, R., IMA Journal of Applied Mathematics 7 (1971), pp. 76-91.

“An Algorithm for Large-Scale Quadratic Programming,” Gould, N.I.M., IMA Journal of Numerical Analysis 11 (1991), pp. 299-324.

“A Stabilization of the Simplex Method,” Bartels, R.H., Numerische Mathematik 16 (1971), pp. 414-434.

Examples

```
set.seed(2)
RP <- randomQP(8, "indefinite")
with(RP, QPmin(G, g, t(A), b, neq = 0, tol = 1e-8))
```

`QPgen.internal.bilinear` *QPgen.internal.bilinear*

Description

Generates a separable bilinear problem of the form

$$\min_x \frac{1}{2} x^T G x + x^T g$$
$$Ax \geq b$$

Usage

```
QPgen.internal.bilinear(m, alphas)
```

Arguments

`m`	Integer parameter controlling the number of variables (2m) and constraints (3m) for the generated problem.
`alphas`	m positive parameters.

Details

The problem is an indefinite problem with 2^m local minima of which 2^n are global. Here, n is equal to the number of alphas exactly equal to 0.5. The constraints are guaranteed independent only at each solution, but not generally everywhere in the feasible region.

Value

`G`	The quadratic component of the objective function.
`g`	The linear component of the objective function
`A`	The constraints coefficient matrix. This matrix has 3m rows and 2m columns.
`b`	The vector with the lower bounds on the constraints.
`opt`	An approximate expected value at the optimum solutions.
`globals`	A list containing all of the global solutions to the problem.

Note

The function 'randomQP' uses 'QPgen.internal.bilinear' to construct non-separable indefinite problems. The technique used to conceal the separability of the problem also eliminates bi-linearity.

Author(s)

Andrea Giusto

References

"A new technique for generating quadratic programming test problems," Calamai P.H., L.N. Vicente, and J.J. Judice, Mathematical Programming 61 (1993), pp. 215-231.

Examples

```
m <- 3
nhalves <- ceiling(log(m))
nmiss <- m - nhalves
alphas <- c(runif(nmiss), rep(0.5, nhalves))
QPgen.internal.bilinear(m, alphas)
```

QPgen.internal.concave

QPgen.internal.concave

Description

Generates a separable strictly concave quadratic problem of the form

$$\min_x \frac{1}{2} x^T G x + x^T g$$

$$Ax \geq b$$

Usage

```
QPgen.internal.concave(m, thetas, alphas, betas, L)
```

Arguments

`m`	Integer parameter controlling the number of variables (2m) and constraints (3m) for the generated problem.
`thetas`	m binary $\{0, 1\}$ parameters.
`alphas`	m parameters taking on either of the two values $\{1.5, 2\}$
`betas`	m parameters taking on either of the two value $\{1.5, 2\}$. Each entry of betas must be different from each entry of alphas.
`L`	A positive parameter

Details

Denoting m0 the number of thetas == 0, the generated problem has a unique global minimum and 3^m0 local ones. All of the local minima have distinct function values. While the constraints are guaranteed to be indipendent at each solution, they are not guaranteed independent everywhere, for this reason too large problems tend to not be solvable with QPmin, which relies on a basic version of the simplex method for linear programming to find an initial feasible point. A large enough problem will generally halt the execution due to linear dependence in the basis constraints.

Value

`G`	The quadratic component of the objective function.
`g`	The linear component of the objective function
`A`	The constraints coefficient matrix. This matrix has 3m rows and 2m columns.
`b`	The vector with the lower bounds on the constraints.
`opt`	An approximate expected value at the optimum solutions.
`globals`	A list containing all of the global solutions to the problem.

Note

The function 'randomQP' uses 'QPgen.internal.concave' to construct non-separable concave problems.

Author(s)

Andrea Giusto

References

"A new technique for generating quadratic programming test problems," Calamai P.H., L.N. Vicente, and J.J. Judice, Mathematical Programming 61 (1993), pp. 215-231.

Examples

```
m <- 3
thetas <- round(runif(m))
draws <- runif(m)
belowhalf <- draws < 0.5
alphas <- betas <- c()
alphas[belowhalf] <- 1.5
alphas[!belowhalf] <- 2
betas[belowhalf] <- 2
betas[!belowhalf] <- 1.5
L <- ceiling(log(m))
QPgen.internal.concave(m, thetas, alphas, betas, L)
```

QPgen.internal.convex *QPgen.internal.convex*

Description

Generates a separable convex quadratic problem of the form

$$\min_x \frac{1}{2} x^T G x + x^T g$$

$$Ax \geq b$$

Usage

```
QPgen.internal.convex(m, alphas, rhos, omegas)
```

Arguments

`m`	Integer parameter controlling the number of variables (2m) and constraints (3m) for the generated problem.
`alphas`	m parameters taking values between 5 and 7.5.
`rhos`	m parameters taking values in {0, 1}.
`omegas`	m parameters taking values in {0, 1}.

Details

The problem has a unique global minimum and the constraints are linearly independent at all of the solutions.

Value

`G`	The quadratic component of the objective function.
`g`	The linear component of the objective function
`A`	The constraints coefficient matrix. This matrix has 3m rows and 2m columns.
`b`	The vector with the lower bounds on the constraints.
`opt`	An approximate expected value at the optimum solutions.
`globals`	A list containing all of the global solutions to the problem.

Note

The function 'randomQP' uses 'QPgen.internal.convex' to construct non-separable convex problems.

Author(s)

Andrea Giusto

References

"A new technique for generating quadratic programming test problems," Calamai P.H., L.N. Vicente, and J.J. Judice, Mathematical Programming 61 (1993), pp. 215-231.

Examples

```
m <- 3
alphas <-  runif(m, min = 5, max = 7.4999)
rhos <- round(runif(m))
omegas <- round(runif(m))
QPgen.internal.convex(m, alphas, rhos, omegas)
```

QPmin *QPmin*

Description

Active set method for the solution of the quadratic program

$$\min_x \frac{1}{2} x^T G x + x^T g$$

$$A^T x \geq b$$

$$Lb \leq x \leq Ub$$

where the first neq rows of the system $A^T x \geq b$ are strict equalities.

Usage

```
QPmin(G, g, A, b, neq, Lb = NULL, Ub = NULL, tol = 1e-06, initialPoint = NULL,
      existingLU = NULL, returnLU = FALSE)
```

Arguments

G	The quadratic term of the objective function. Must be symmetric.
g	The linear term of the objective function.
A	The constraints coefficient matrix. This matrix has as many rows as the dimensions of G and g and an arbitrary number of columns. If the problem has no linear constraints (but it still may have simple bound constraints), it is still necessary to pass a matrix with the correct number of rows and zero columns.
b	The lower bounds on the constraints.
neq	The number of rows of A to be interpreted as strict equalities.
Lb	An array with the lower bounds on the variables. If there are no lower bounds, this matrix may be omitted, but if otherwise specified, it must contain an entry for every variable of the problem in the corresponding position. The value -Inf can be used to specify that a variable is unconstrained from below.

Ub	An array with the upper bounds on the variables. The rules about Lb apply to this parameter, too.
tol	A tolerance parameter for the evaluation of the convergence criterion.
initialPoint	Intended as a "hot start" facility, when the user has a feasible starting point for which the second order conditions are satisfied. The code checks if initialPoint is feasible and it terminates if it isn't. However, there are no checks to insure thta the second order conditions are satisfied. See Fletcher (1971).
existingLU	Intended as a "warm start" facility, it allows the user to avoid the re-factorization of a valid basis matrix into its L and U components. See Fletcher (1971) and Bartels (1969).
returnLU	Logical value controlling if the updated LU decomposition of the basis matrix should be returned for later re-use as part of a warm-start strategy.

Value

x	The (possibly local) solution found by the algorithm.
active	A list with the active constraints. The first neq constraints are always binding. The numbers above m relate to the lower and upper bounds respectively as specified by the user. For example in a case with 8 variables 5 constraints (2 of which are equalities), with lower bounds on the third and fifth variables, and an upper bound on the first variable, an active set of {1, 2, 4, 6, 8 } indicates that the first, second and fourth constraints are binding, as well as both the lower bounds on variable 3 and upper bound on variable 1.
lambdas	The Lagrange multipliers associated with each one of the binding constraints
warmStart	The existing LU decoposition in its product form.

Author(s)

Andrea Giusto

References

"A General Quadratic Programming Algorithm," Fletcher, R., IMA Journal of Applied Mathematics 7 (1971), pp. 76-91.

"An Algorithm for Large-Scale Quadratic Programming," Gould, N.I.M., IMA Journal of Numerical Analysis 11 (1991), pp. 299-324.

"A Stabilization of the Simplex Method," Bartels, R.H., Numerische Mathematik 16 (1971), pp. 414-434.

Examples

```
G <- diag(5)
g <- c(0,-6,-6,-12,-9)
A <- matrix(c(2, rep(0,3), -1, 5, 0, -3, 0, -1, 0, -1, 0, -3, 0), 5, 3)
b <- c(0, 0, 0)
Lb <- c(-Inf, -Inf, 0, 0, 0)
Ub <- c(4, 8, Inf, Inf, Inf)
```

```
neq <- 0
QPmin(G, g, A, b, neq = neq, Lb, Ub, tol = 1e-6)

set.seed(2)
RP <- randomQP(8, "indefinite")
Lb <- rep(-Inf, 8)
Ub <- -Lb
with(RP, QPmin(G, g, t(A), b, neq = 0, Lb, Ub, tol = 1e-06))
```

---

randomQP *randomQP*

Description

Generates a non-separable random quadratic program of the specified type.

Usage

```
randomQP(n, type = c("convex", "concave", "indefinite"))
```

Arguments

n	The random problem generated will have n variables and m = 3n/2 constraints. Must be an even number.
type	Specifies the curvature of the objective function.

Details

The algorithm is based on Calamai, Vicente, and Judice (1993). It generates a random quadratic program with the following form

$$\min_x \frac{1}{2}x^T G x + x^T g$$

$$Ax \geq b$$

Value

G	The quadratic component of the objective function. Must be symmetric.
g	The linear component of the objective function
A	The constraints coefficient matrix. This matrix has n rows and m columns.
b	The vector with the lower bounds on the constraints.
opt	An approximate expected value at the optimum solutions.
solutions	A list containing all of the global solutions to the problem.

Author(s)

Andrea Giusto

References

"A new technique for generating quadratic programming test problems," Calamai P.H., L.N. Vicente, and J.J. Judice, Mathematical Programming 61 (1993), pp. 215-231.

Examples

```
n <- 8

RP <- randomQP(n, "concave")

RP2 <- randomQP(n, "indefinite")

RP3 <- randomQP(n, "convex")
```

Index