Package ‘intpoint’
February 20, 2015

Type Package
Title linear programming solver by the interior point method and graphically (two dimensions)
Version 1.0
Date 2012-05-30
Author Alejandro Quintela del Rio
Maintainer Alejandro Quintela del Rio <aquintela@udc.es>
Description Solves linear programming problems by the interior point method, and plots the graphical solution of a linear programming problem of two dimensions.
License GPL-2
LazyLoad yes
Repository CRAN
Date/Publication 2012-05-30 15:31:53
NeedsCompilation no

R topics documented:

  interior_point .................................................. 1
  random_problem ................................................... 3
  solve2dlp ......................................................... 4

Index

interior_point Solves a linear programming problem using the interior point method

Description

This function solves a linear programming problem using a modification of Karmarkar’s linear programming algorithm, developed by Vanderbei, Meketon and Freedman (1986). This algorithm uses a recentered projected gradient approach without a priori knowledge of the optimal objective function value.
Usage

interior_point(t, c, bA = NULL, A = NULL, bm = NULL, m = NULL, bM = NULL, M = NULL, e = 1e-04, a1 = 1, a2 = 0.97)

Arguments

t Type of problem. Put 1 if it is a maximization problem, and -1 if it is a minimization one.
c The vector corresponding to the coefficients of the objective function
bA The vector corresponding to the right hand side constants (with =)
A The coefficients of the problem constraints matrix A (with =)
bm The vector corresponding to the right hand side constants (with <=)
m The coefficients of the problem constraints matrix A (with <=)
bM The vector corresponding to the right hand side constants (with >=)
M The coefficients of the problem constraints matrix A (with >=)
e The value of $\epsilon$. Default is 1e-04
a1 The value of $\alpha_1$. Default is 1
a2 The value of $\alpha_2$. Default is 0.97

Details

The function is defined in the form that we only need to put the values used in the problem. For example, for a linear programming problem in the form:

Maximize $Z = c_1x_1 + c_2x_2 + ... + c_nx_n$
subject to
$a_{11}x_1 + a_{12}x_2 + ... a_{1n}x_n \leq b_1$
$a_{21}x_1 + a_{22}x_2 + ... a_{2n}x_n \leq b_2$
... 
a_{m1}x_1 + a_{m2}x_2 + ... a_{mn}x_n \leq b_m$

that is, all the restrictions are inequality in the form “$\leq$”, we need only to specify the vector bm and the vector(s) corresponding to the row(s) of A.

Value

a list with the values

$Z$ The optimum value of the objective function
$xf$ The solution vector
$n$ The number of iterations to solve the problem

Author(s)

Alejandro Quintela del Rio <aquintela@udc.es>
References


Examples

```r
##
c<-c(1,3,-2)
bA<-c(25,30)
A<-array(4,c(2,3))
interior_point(-1,c,bA,A)

##
c<-c(1,2)
bm<-c(4)
BM<-c(1)
m<-array(1,c(1,2))
m[1,1]<-0
M<-array(1,c(1,2))
interior_point(1,c,bm=m,m=bM,BM=M)

##
# This example is taken from Exercise 7.5 of Gill, Murray, # and Wright (1991).

enj <- c(200, 6000, 3000, -200)
fat <- c(800, 6000, 1000, 400)
vitx <- c(50, 3, 150, 100)
vity <- c(10, 10, 75, 100)
vitz <- c(150, 35, 75, 5)
interior_point(1,c=enj, m=fat, bm=13800, M=rbind(vitx, vity, vitz),
BM = c(600, 300, 550))
```

**random_problem**

Random squared (n x n) linear programming problems are generated and solved by the interior point method

**Description**

A random linear programming problem, with the form $\text{Max} \ Z = CX$ subject to $AX \leq b, X \geq 0$ is generated, using $U(0,1)$ values for both the matrix $A$ and the vectors $C$ and $b$. Next, the interior point is used to solve the problem. If the number of equations (variables) is less than or equal to 5, the input problem is shown.
solve2dlp

Usage

random_problem(n)

Arguments

n The size of the problem (number of equations and variables)

Value

A The coefficient matrix A

b The right hand side constants

z Optimum value for the objective function

xf Solution vector

n Number of iterations

Author(s)

Alejandro Quintela del Rio <aquintela@udc.es>

References


Examples

```r
## generating and solving a linear programming problem with uniform (0,1) 
## random values
random_problem(10)
```

solve2dlp Graphical solution of a two-dimensional linear programming problem. The sequence of points calculated by the interior point method is also shown, by request.
**Description**

The graphical method for solving linear programming problems in two variables is implemented. The lines corresponding to the constraints are drawn; next, the coordinates of the corner points of the feasible region are computed, and the objective function is evaluated to obtain the optimal value. Finally, the objective function is drawn over the optimal point. If the interior point algorithm is loaded, then the sequence of points obtained by this method are shown. Details: 1. Graph the feasible region. 2. Compute the coordinates of the corner points. 3. Substitute the coordinates of the corner points into the objective function to see which gives the optimal value. This point is the solution to the linear programming problem. 4. If the feasible region is not bounded, this method can be misleading: optimal solutions always exist when the feasible region is bounded, but may or may not exist when the feasible region is unbounded. If the feasible region is unbounded, we are minimizing the objective function, and its coefficients are non-negative, then a solution exists, so this method yields the solution.

**Usage**

```r
solve2dlp(t = 1, c = NULL, bA = NULL, A = NULL, bm = NULL, m = NULL, 
bM = NULL, M = NULL, z = 1, ip = 1, e = 1e-04, a1 = 1, a2 = 0.97)
```

**Arguments**

- `t` Type of problem. Put 1 in the case of a maximization problem, and -1 if it is a minimization one.
- `c` The vector corresponding to the coefficients of the objective function
- `bA` The vector corresponding to the right hand side constants (with =)
- `A` The coefficients of the problem constraints matrix A (with =)
- `bm` The vector corresponding to the right hand side constants (with <=)
- `m` The coefficients of the problem constraints matrix A (with <=)
- `bM` The vector corresponding to the right hand side constants (with >=)
- `M` The coefficients of the problem constraints matrix A (with >=)
- `z` Put 0 if you only want to show the feasible region, not the solution. Default is 1
- `ip` Put 1 if you want to solve the problem by the interior point method, and 0 if not. The sequence of calculated points is displayed. Default is 1
- `e` The value of $\epsilon$. Default is 1e-04
- `a1` The value of $\alpha_1$. Default is 1
- `a2` The value of $\alpha_2$. Default is 0.97

**Value**

The graphical solution to the linear programming problem. Lines corresponding to the constrain are coloured in green, the corner points in blue and the optimal point (if there exists) in magenta (with the objective function in blue). The sequence of interior points is drawn in red.

**Author(s)**

Alejandro Quintela del Rio <aquintela@udc.es>
References


Examples

```r
# Max Z = x2
# Subject to 2x1 - x2 = -2
# x1 + 2x2 = 8
# x1, x2 >= 0

c=c(3,2)
M1=c(2,-1)
bM1=-2
m1=c(1,2)
bm1=8
solve2dlp(t=1, m=m1, bm=bm1, M=M1, bM=bM1, c=c, z=0, ip=0)

# A unbounded problem
# Z = x1 + 2x2
# Subject to x1 + x2 = 1
# x2 = 4
# x1, x2 = 0

m1<-c(0,1)
bm1<-4
M1<-c(1,1)
bM1<-1
c<-c(1,2)
solve2dlp(t=1, m=m1, bm=bm1, M=M1, bM=bM1, c=c, z=1, ip=1)

# a problem with several constraints
m1<-rbind(c(1,0), c(1,2),c(0,1))
bm1<-c(5,10,4)
c=c(1,3)
solve2dlp(t=1, m=m1, bm=bm1, c=c, z=1, ip=1)
```
Index

*Topic **math**
  interior_point, 1
  random_problem, 3
  solve2dlp, 4

*Topic **optimize**
  interior_point, 1
  random_problem, 3
  solve2dlp, 4