Package ‘PoA’

January 20, 2025

Type Package

Title Finds the Price of Anarchy for Routing Games

Version 1.2.1

Author Hector Haffenden

Maintainer Hector Haffenden <haffendenh@cardiff.ac.uk>

Description Computes the optimal flow, Nash flow and the Price of Anarchy for any routing game defined within the game theoretical framework. The input is a routing game in the form of it's cost and flow functions. Then transforms this into an optimisation problem, allowing both Nash and Optimal flows to be solved by nonlinear optimisation. See <https://en.wikipedia.org/wiki/Congestion_game> and Knight and Harper (2013) <doi:10.1016/j.ejor.2013.04.003> for more information.

Imports stats, dplyr, tibble, pracma, nloptr,

License GPL-2

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2019-03-18 17:33:45 UTC

Contents

`PriceOfAnarchy` *Compoute the Price of Anarchy for a Routing Game.*

Description

'PriceOfAnarchy()' returns Price of Anarchy (PoA) for a routing game. This is a routing game solver for a generic network. For this to work properly, set the inputs as shown in the examples. This is as a tibble, with one column containing the cost functions, and another containing the flow.

Usage

```
PriceOfAnarchy(cost.and.flow.tibble, variable.limits)
```

Arguments

`cost.and.flow.tibble`	A tibble with 2 columns, cost which contains the cost functions, and flow containing the respective flows. When defining the flow functions, use all variables as inputs for each function.
`variable.limits`	A vector of limits for each variable, described by the proportions at each source node.

Value

Returns the Price of Anarchy, Optimal cost and the respective parameters, Nash cost and the respective parameters.

Requires tidyverse, pracma and nloptr packacges

Examples

```
library(stats, "na.omit")
library(dplyr)
library(tibble)
library(pracma)
library(nloptr)

### 2 player game ###
player2 <- tibble(cost = c(function(x){0}, function(x){0}, function(x){NA},
                  function(x){x^2}, function(x){(3/2)*x},  function(x){x}),
              flow = c(function(alpha,beta){(1/2)-alpha},function(alpha,beta){(1/2)-beta},
                     function(alpha,beta){NA},function(alpha,beta){alpha},
                     function(alpha,beta){beta},function(alpha, beta){1-alpha-beta}))
PriceOfAnarchy(player2, c((1/2), (1/2)))

### 3 player - 2x2 middle ###
player3middle2x2 <- tibble(cost = c(function(x){0}, function(x){x},function(x){x^2},
function(x){x},function(x){x},function(x){0},function(x){x},function(x){x^2},
function(x){x^2},function(x){x},function(x){(1/2)*x},function(x){x}),
                            flow = c(function(d1,d2,d3,d4,d5){d1},
                            function(d1,d2,d3,d4,d5){1-d1},function(d1,d2,d3,d4,d5){d2},
                            function(d1,d2,d3,d4,d5){1-d2},function(d1,d2,d3,d4,d5){d3},
                            function(d1,d2,d3,d4,d5){1-d3},function(d1,d2,d3,d4,d5){d4},
                      function(d1,d2,d3,d4,d5){d1+d2+d3-d4},function(d1,d2,d3,d4,d5){d5},
                            function(d1,d2,d3,d4,d5){1-d1-d2-d3-d5},
                      function(d1,d2,d3,d4,d5){d4+d5},function(d1,d2,d3,d4,d5){1-d4-d5}))
PriceOfAnarchy(player3middle2x2, c((6/10),(1/10),(3/10),1,1))
```

Index