Package ‘Inventorymodel’

October 12, 2022

Encoding latin1

Title Inventory Models

Version 1.1.0

Date 2017-12-06

Author Alejandro Saavedra Nieves

Maintainer Alejandro Saavedra Nieves <alejandro.saavedra.nieves@gmail.com>

Depends R(>= 2.15.0),e1071, GameTheoryAllocation

Description

Determination of the optimal policy in inventory problems from a game-theoretic perspective.

License GPL-2

LazyLoad yes

Repository CRAN

Date/Publication 2017-12-06 16:10:55 UTC

NeedsCompilation no

R topics documented:

`coalitions` *Coalitions*

Description

Possible coalitions with n players.

Usage

```
coalitions(n)
```

Arguments

`n`	Number of players

Value

This function gives the coalitions in a binary mode and usual way.

Examples

```
coalitions(3)
#$Binary
#     [,1] [,2] [,3]
#[1,]    0    0    0
#[2,]    1    0    0
#[3,]    0    1    0
#[4,]    0    0    1
#[5,]    1    1    0
#[6,]    1    0    1
#[7,]    0    1    1
#[8,]    1    1    1
#
#$Classic
#[1] "0"     "'{ 1 }'" "'{ 2 }'" "'{ 3 }'" "'{ 1,2 }'" "'{ 1,3 }'" "'{ 2,3 }'"
#[8] "'{ 1,2,3 }'"
```

EOQ *EOQ*

Description

This function obtains the optimal orders and the associated cost in the EOQ model.

Usage

```
EOQ(n = NA, a = NA, d = NA, h = NA, m = NA)
```

Arguments

n	Number of agents in the inventory model.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding costs to each agent.
m	Vector. Number of orders to each agent (optional).

Value

This function calculates two vectors. The first shows the optimal order for each agent. The second vector indicates the associated cost to these orders.

Examples

```
EOQ(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10))

#EOQ model
#$"Optimal order"
#[1] 250.0000 180.9068 219.0890
#
#$"Order costs"
#[1] 2400.000 1989.975 2190.890
```

EOQcoo *EOQcoo*

Description

This function obtains the optimal orders and the associated cost when agents are cooperating in the EOQ model.

Usage

```
EOQcoo(n = NA, a = NA, d = NA, h = NA, m = NA)
```

Arguments

n	Number of agents in the inventory model.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding costs to each agent.
m	Vector. Number of orders to each agent (optional).

Value

A list with the following components:

- `Optimal order` If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The next *n* columns contain the associated cost to each agent in the coalition. Last column indicates the global cost of the optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.

Examples

```
EOQcoo(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10))

#EOQ model
#Cooperative case
#Optimal order
#    Coalition        1        2        3 Coalitional costs
#1           0   0.0000   0.0000   0.0000             0.000
#2     '{ 1 }' 250.0000   0.0000   0.0000          2400.000
#3     '{ 2 }'   0.0000 180.9068   0.0000          1989.975
#4     '{ 3 }'   0.0000   0.0000 219.0890          2190.890
#5   '{ 1,2 }' 192.4501 115.4701   0.0000          3117.691
#6   '{ 1,3 }' 184.6372   0.0000 147.7098          3249.615
#7   '{ 2,3 }'   0.0000 121.6327 162.1770          2959.730
#8 '{ 1,2,3 }' 157.4592  94.4755 125.9673          3810.512
```

`EPQ` *EPQ*

Description

This function obtains the optimal orders and the associated cost in the EPQ model.

Usage

```
EPQ(n = NA, a = NA, d = NA, h = NA, m = NA, r = NA, b = NA)
```

Arguments

n	Number of agents in the inventory model.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding costs to each agent.
m	Vector. Number of orders to each agent (optional).
r	Vector. Replacement rate to each agent. In general, r>d.
b	Vector. Cost of a shortage to each agent.

Value

This function calculates two vectors. The first one shows the optimal order for each agent. The second vector indicates the associated cost to these orders.

Examples

```
EPQ(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10),m=NA,r=rep(600,3),b=c(100,150,200))
#EPQ model
#$"Optimal order"
#[1] 641.0928 265.0557 388.8444
#
#$"Optimal shortages"
#[1] 9.359019 9.054699 6.172134
#
#$"Order costs"
#[1]  935.9019 1358.2049 1234.4268
```

EPQcoo *EPQcoo*

Description

This function obtains the optimal orders and the associated cost when agents are cooperating in the EPQ model.

Usage

```
EPQcoo(n = NA, a = NA, d = NA, h = NA, m = NA, r = NA, b = NA)
```

Arguments

n	Number of agents in the inventory model.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding costs to each agent.

m	Vector. Number of orders to each agent (optional).
r	Vector. Replacement rate to each agent. In general, r>d.
b	Vector. Cost of a shortage to each agent.

Value

A list with the following components:

- `Optimal order` If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The next *n* columns contain the associated cost to each agent in the coalition. Last column indicates the global cost of the optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.
- `Optimal shortages` A matrix, for each coalition (row), contains in the column *i* the allowed optimal shortages.

Examples

```
EPQcoo(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10),r=rep(600,3),b=c(100,150,200))

#EPQ model
#Cooperative case
#$`Optimal order
#        1        2        3     Costs
#   0.0000   0.0000   0.0000    0.0000
# 641.0928   0.0000   0.0000  935.9019
#   0.0000 265.0557   0.0000 1358.2049
#   0.0000   0.0000 388.8444 1234.4268
# 363.7611 218.2567   0.0000 1649.4341
# 387.3208   0.0000 309.8566 1549.1036
#   0.0000 196.1473 261.5297 1835.3556
# 291.2332 174.7399 232.9866 2060.2045
#
#$`Optimal shortages`
#        1        2        3
# 0.000000 0.000000 0.000000
# 9.359019 0.000000 0.000000
# 0.000000 9.054699 0.000000
# 0.000000 0.000000 6.172134
# 5.310381 7.455973 0.000000
# 5.654318 0.000000 4.918359
# 0.000000 6.700683 4.151265
# 4.251580 5.969377 3.698200
```

`Inventory Models` *Inventory Models*

Description

This package allows the determination of the optimal policy in terms of the number of orders to apply in the most common inventory problems. Moreover, game-theoretic procedures to share the costs of these situations have been considered by proposing allocations for the involved agents.

Details

Package:	Inventorymodel
Type:	Package
Version:	1.0
Date:	2017-04-05
License:	–

This package incorporates the functions `EOQ` and `EOQcoo`, which compute the optimal policy in an EOQ model. For studying the optimal orders and costs in an EPQ model, functions `EPQ` and `EPQcoo` can be used. The package includes the function `SOC` for the SOC allocation rule. For the inventory transportation system (STI), the functions `STI`, `STIcoo` and `reglalineacoalitional` implement the associated games to these situations and their allocation rule (line rule). The function `mfoc` calculates the optimal order and its associated cost to model with fixed order cost (MFOC). Shapley value can be obtained for this class of games with the function `shapley\_mfoc`. The basic EOQ system without holding costs and with transportation cost (MCT) can be studied with the functions `mct` and `twolines` (allocation rule). This package includes the function `mwhc` for models without holding costs (MWHC), the function `mwhc2c` when two suppliers are considered with differents costs of the product and the function `mwhcct` when the transportation costs are considered (MWHCCT).

Author(s)

Saavedra-Nieves, Alejandro

Maintainer: Alejandro Saavedra-Nieves <alejandro.saavedra.nieves@gmail.com>

References

M.G. Fiestras-Janeiro, I. García–Jurado, A. Meca, M. A. Mosquera (2011). Cooperative game theory and inventory management. *European Journal of Operational Research*, **210**(3), 459–466.

M.G. Fiestras-Janeiro, I. García-Jurado, A. Meca, M. A. Mosquera (2012). Cost allocation in inventory transportation systems. *Top*, **20**(2), 397–410.

M.G.~ Fiestras-Janeiro, I.~ García-Jurado, A.~Meca, M. A. ~Mosquera (2014). Centralized inventory in a farming community. *Journal of Business Economics*, **84**(7), 983–997.

M.G. Fiestras-Janeiro, I. García-Jurado, A. Meca, M.A. Mosquera (2015). Cooperation on capacitated inventory situations with fixed holding costs. emphEuropean Journal of Operational Research, **241**(3), 719–726.

A. Meca (2007). A core-allocation family for generalized holding cost games. *Mathematical Methods of Operation Research*, **65**, 499–517.

A. Meca, I. Garc\'ia-Jurado, P. Borm (2003). Cooperation and competition in inventory games. *Mathematical Methods of Operations Research*, **57**(3), 481–493.

A. Meca, J. Timmer, I. García-Jurado, P. Borm (1999). Inventory games. Discussion paper, 9953, Tilburg University.

A. Meca, J. Timmer, I. García-Jurado, P. Borm (2004). Inventory games. *European Journal of Operational Research*, **156**(1), 127–139.

M.A. Mosquera, I. García-Jurado, M.G. Fiestras-Janeiro (2008). A note on coalitional manipulation and centralized inventory management. *Annals of Operations Research*, **158**(1). 183–188.

A. Saavedra-Nieves, I. García-Jurado, M.G. Fiestras-Janeiro (2017a). Placing joint orders when holding costs are negligible and shortages are not allowed. *Game Theory in Management Accounting: Implementing Incentives and Fairness (to appear)*.

A. Saavedra-Nieves, I. García-Jurado, M.G. Fiestras-Janeiro (2017b). On coalition formation in a non-convex multi-agent inventory problem. *Submitted in Annals of Operations Research*.

`inventorygames` *Inventorygames*

Description

Generic function to show the associated cost game to a EOQ or EPQ model.

Usage

```
inventorygames(n = NA, a = NA, d = NA, h = NA, m = NA, r = NA, b = NA,
model = c("EOQ", "EPQ"))
```

Arguments

`n`	Number of agents in the inventory model.
`a`	The fixed cost per order.
`d`	Vector. Deterministic demands per time unit to each agent.
`h`	Vector. Holding costs to each agent.
`m`	Vector. Number of orders to each agent (optional).
`r`	Vector. Replacement rate to each agent. In general, r>d.
`b`	Vector. Cost of a shortage to each agent.
`model`	Model to be selected. `EOQ` and `EPQ` models can be considered.

Value

The characteristic function of the associated cost game is calculated to model `EOQ` or `EPQ`.

Examples

```
inventorygames(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10),model="EOQ")
#EOQ model
#Cooperative case
#Optimal order
#    Coalition        1        2        3 Coalitional costs
#1           0   0.0000   0.0000   0.0000             0.000
#2     '{ 1 }' 250.0000   0.0000   0.0000          2400.000
#3     '{ 2 }'   0.0000 180.9068   0.0000          1989.975
#4     '{ 3 }'   0.0000   0.0000 219.0890          2190.890
#5   '{ 1,2 }' 192.4501 115.4701   0.0000          3117.691
#6   '{ 1,3 }' 184.6372   0.0000 147.7098          3249.615
#7   '{ 2,3 }'   0.0000 121.6327 162.1770          2959.730
#8 '{ 1,2,3 }' 157.4592  94.4755 125.9673          3810.512
```

`inventorymodelfunction` *inventorymodelfunction*

Description

General function in Inventorymodel package for obtaining the optimal policy by selecting any model and its corresponding parameters.

Usage

```
inventorymodelfunction(model=c("EOQ","EPQ","STI","FOC","MCT","MWHC","MWHC2","MWHCCT"),
n=NA,a=NA,av=NA,d=NA,h=NA,m=NA,r=NA,K=NA,b=NA,c1=NA,c2=NA,
cooperation=c(0,1),allocation=c(0,1))
```

Arguments

`model`	Model to be selected. `EOQ`, `EPQ`, `STI`, `mfoc`, `mct`, `mwhc`, `mwhc2c` or `mwhcct` models can be considered.
`n`	Number of agents in the inventory model.
`a`	The fixed cost per order.
`av`	Vector. Transportation costs to each agent.
`d`	Vector. Deterministic demands per time unit to each agent.
`h`	Vector. Holding costs to each agent.
`m`	Vector. Number of orders to each agent(optional).
`r`	Vector. Replacement rate to each agent. In general, r>d.
`K`	Vector. Warehouse capacity to each agent.
`b`	Vector. Shortage cost to each agent.
`c1`	Value. Cost of a product from the first supplier.
`c2`	Value. Cost of a product from the second supplier.
`cooperation`	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.
`allocation`	Option to indicate the allocation. If it is required `allocation=1` else `allocation=0`.

See Also

EOQ, EOQcoo, EPQ, EPQcoo, STI, STIcoo, mct, mfoc, mwhc, mwhc2c and mwhcct.

Examples

```
inventorymodelfunction(model="MCT",n=3,a=400,av=c(300,500,200),d=c(2,2,5),K=c(9,8,7),
cooperation=1,allocation=1)
#MCT model
#Cooperative case
#Two-lines rule
#$`Optimal solution`
#   1 2 3 Coalition     Cost
#   0 0 0         0   0.0000
#   1 0 0         1 155.5556
#   0 1 0         2 225.0000
#   0 0 1         3 428.5714
#   1 1 0        12 225.0000
#   1 0 1        13 500.0000
#   0 1 1        23 642.8571
#   1 1 1       123 642.8571
#
#$`Allocation two-lines rule`
#[1]   0.0000 219.6429 423.2143
```

---

linerule *Line rule*

Description

Line rule for an inventory transportation system when coalition *N* is formed.

Usage

```
linerule(n=NA,a=NA,av=NA,d=NA,h=NA,m=NA)
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
av	Vector. The transportations cost per order to each agent.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding cost per time unit to each agent.
m	Vector. Number of orders to each agent (optional).

Value

Allocation for each agent if *N* was formed.

Examples

```
linerule(n=3,a=200,av=c(300,300,900),d=c(90,80,20),h=c(0.06,0.06,0.1),m=NA)

#Line rule
#[1] 51.38935 46.10733 66.33250
```

`linerulecoalitional` *Line rule (coalitional)*

Description

Line rule for an inventory transportation system for each possible coalition.

Usage

```
linerulecoalitional(n = NA, a = NA, av = NA, d = NA, h = NA, m = NA)
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
av	Vector. The transportations cost per order to each agent.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding cost per time unit to each agent.
m	Vector. Number of orders to each agent (optional).

Value

Matrix with allocations proposed by Line rule to agents (columns) in each possible coalition *S* (row).

Examples

```
linerulecoalitional(n=3,a=200,av=c(300,300,900),d=c(90,80,20),
h=c(0.06,0.06,0.1),m=NA)

#    Coalition        1        2      3 Coalitional cost
#1           0  0.00000  0.00000  0.0000          0.00000
#2     '{ 1 }' 73.48469  0.00000  0.0000         73.48469
#3     '{ 2 }'  0.00000 69.28203  0.0000         69.28203
#4     '{ 3 }'  0.00000  0.00000 66.3325         66.33250
#5   '{ 1,2 }' 52.59885 48.39619  0.0000        100.99505
#6   '{ 1,3 }' 61.26061  0.00000 66.3325        127.59310
#7   '{ 2,3 }'  0.00000 55.97858 66.3325        122.31108
#8 '{ 1,2,3 }' 51.38935 46.10733 66.3325        163.82918
```

marginal_contribution_mean
fauxiliar

Description

Intermediate auxiliar function to calculate the mean of the marginal contributions for a set of permutations.

Usage

```
marginal_contribution_mean(permute, costs)
```

Arguments

permute	Matrix with n columns. By rows, it contains a order.
costs	Vector with the associated costs to each posible coalition in a set of agents N.

Value

A vector with n elements with component i equal to the mean of the marginal contribution indicated by each order in *permute* for agent i.

mct *MCT*

Description

This function obtains the associated costs in a basic EOQ system without holding costs and with transportation cost.

Usage

```
mct(n = NA, a = NA, av = NA, d = NA, K = NA, cooperation = c(0, 1))
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
av	Vector. The transportations cost per order to each agent.
d	Vector. Deterministic demands per time unit to each agent.
K	Vector. Warehouse's capacity to each agent.
cooperation	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.

Value

If `cooperation=0`, a vector with the individual cost to each agent in a `MCT`. If `cooperation=1` and *n* is lesser than *0*, a matrix which contains the associated costs for each possible group. Otherwise, this matrix only contains the individual costs and the associated values for *N*.

Examples

```
mct(n=3,a=400,av=c(300,500,200),d=c(2,2,5),K=c(9,8,7),cooperation=0)

#MCT model
#Individual cost
#[1] 155.5556 225.0000 428.5714
```

`mfoc` *MFOC*

Description

This function obtains the associated costs in a fixed order cost model.

Usage

```
mfoc(n = NA, a = NA, d = NA, K = NA, cooperation = c(0, 1))
```

Arguments

`n`	Agents in the inventory situation.
`a`	The fixed cost per order.
`d`	Vector. Deterministic demands per time unit to each agent.
`K`	Vector. Warehouse's capacity to each agent.
`cooperation`	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.

Value

If `cooperation=0`, a vector with the individual cost to each agent in a `MFOC`. If `cooperation=1` and *n* is lesser than *0*, a matrix which contains the associated costs for each possible group. Otherwise, this matrix only contains the individual costs and the associated values for *N*.

Examples

```
mfoc(n=3,a=200,d=c(1.2,1.4,1.2),K=c(6,10,8),cooperation=0)

#MFOC model
#[1] 40 28 30
```

mwhc *MWHC*

Description

This function obtains the associated costs in a model without holding costs. Demands and capacities must be introduced in the order indicated by the ratios d/K. In other case, agents change their position.

Usage

```
mwhc(n = NA, a = NA, b = NA, d = NA, K = NA, cooperation = c(0, 1),
allocation = c(0, 1))
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
b	Vector. Shortage cost per unit to each agent.
d	Vector. Deterministic demands per time unit to each agent.
K	Vector. Warehouse's capacity to each agent.
cooperation	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.
allocation	Option to indicate the allocation. If it is required `allocation=1` else `allocation=0`.

Value

A list with the following components:

- `"Optimal policies"` If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The second column contains the optimal order to each coalition. Last column indicates the global cost of this optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.
- `"R-rule"` A matrix, for each coalition (row), contains the coalition *i(S)* and allocations proposed by R-rule.

Examples

```
mwhc(n=4,a=180,b=c(15,15,10,12),d=c(0.45,0.95,1.05,1.2),K=c(5,7.5,8,9),
cooperation=1,allocation=1)

#MWHC model
#Cooperative case
#$`Optimal policies`
#      Coalitions Optimal orders    Costs
#1              0     0.00000000  0.00000
#2        '{ 1 }'     0.07520921 14.74965
```

```
#3        '{ 2 }'     0.10684954 20.86510
#4        '{ 3 }'     0.10406757 20.89599
#5        '{ 4 }'     0.11094004 21.79985
#6      '{ 1,2 }'     0.10684954 20.86510
#           ..
#
#$`R-rule`
#  Coalition_SxT        1         2         3        4
#              0  0.00000  0.000000  0.000000  0.00000
#              1 14.74965  0.000000  0.000000  0.00000
#              2  0.00000 20.865100  0.000000  0.00000
#              3  0.00000  0.000000 20.895986  0.00000
#              4  0.00000  0.000000  0.000000 21.79985
#              2  0.00000 20.865100  0.000000  0.00000
#              3  0.00000  0.000000 20.895986  0.00000
#             ..
```

---

mwhc2c *MWHC2C*

Description

This function obtains the associated costs in a model without holding costs and with two differents cost of product. Demands and capacities must be introduced in the order indicated by the ratios `d/K`. In other case, agents change their position.

Usage

```
mwhc2c(n=NA,a=NA,b=NA,d=NA,K=NA,c1=NA,c2=NA,cooperation=c(0,1),allocation=c(0,1))
```

Arguments

`n`	Agents in the inventory situation.
`a`	The fixed cost per order.
`b`	Vector. Shortage cost per unit to each agent.
`d`	Vector. Deterministic demands per time unit to each agent.
`K`	Vector. Warehouse's capacity to each agent.
`c1`	Value. Cost per unit of product from the first vendor.
`c2`	Value. Cost per unit of product from the second vendor.
`cooperation`	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.
`allocation`	Option to indicate the allocation. If it is required `allocation=1` else `allocation=0`.

Value

A list with the following components:

- "Optimal policies" If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The second column contains the optimal order to each coalition. Last column indicates the global cost of this optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.
- "GR-rule" A matrix, for each coalition (row), contains the coalition *i(S)* and allocations proposed by GR-rule.

Examples

```
mwhc2c(n=3,a=1,b=c(10,10,10),d=c(30,45,46),K=c(30,45,46),c1=3.5,c2=3,cooperation=1,allocation=1)
#MWHC model
#$`Optimal policies`
#   Coalitions Optimal orders    Costs
#1           0      0.0000000   0.0000
#2     '{ 1 }'      0.9505864 105.5947
#3     '{ 2 }'      0.9515422 157.9165
#4     '{ 3 }'      0.9515838 161.4046
#5   '{ 1,2 }'      0.9523090 262.5595
#6   '{ 1,3 }'      0.9523241 266.0476
#7   '{ 2,3 }'      0.9525115 318.3690
#8 '{ 1,2,3 }'      0.9527470 423.0118
#
#$`GR-rule`
#  Coalition_SxT        1        2        3
#              0   0.0000   0.0000   0.0000
#              1 105.5947   0.0000   0.0000
#              2   0.0000 157.9165   0.0000
#              3   0.0000   0.0000 161.4046
#             12 105.0238 157.5357   0.0000
#             13 105.0188   0.0000 161.0288
#             23   0.0000 157.4352 160.9338
#            123 104.8790 157.3184 160.8144
```

---

mwhcct *MWHCCT*

Description

This function obtains the associated costs in a basic EOQ system without holding costs and with a general transportation cost.

Usage

```
mwhcct(n = NA, a = NA, av = NA, d = NA, K = NA, cooperation = c(0, 1),allocation = c(0,1))
```

Arguments

`n`	Agents in the inventory situation.
`a`	The fixed cost per order.
`av`	Vector. The transportations cost per order to each possible group of agents.
`d`	Vector. Deterministic demands per time unit to each agent.
`K`	Vector. Warehouse's capacity to each agent.
`cooperation`	Option to indicate cooperation. If it exists `cooperation=1` else `cooperation=0`.
`allocation`	Option to indicate the allocation. If it is required `allocation=1` else `allocation=0`.

Value

If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The next *n* columns contain the associated cost to each agent in the coalition. Last column indicates the global cost of the optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.

Examples

```
mwhcct(n=3,a=10,av=c(0,10,10,10,20,20,20,30),d=c(1,0.95,0.9),K=c(1,1,1),
cooperation=1,allocation = 1)
# MWHC with Transportation Costs model
#Cooperative case
#[1] "Shapley Value"
#$`Optimal solution`
#  1 2 3   Coalition Cost
#1 0 0 0           0  0.0
#2 1 0 0     '{ 1 }' 20.0
#3 0 1 0     '{ 2 }' 19.0
#4 0 0 1     '{ 3 }' 18.0
#5 1 1 0   '{ 1,2 }' 30.0
#6 1 0 1   '{ 1,3 }' 30.0
#7 0 1 1   '{ 2,3 }' 28.5
#8 1 1 1 '{ 1,2,3 }' 40.0
#
#$`Allocation R rule`
#      1     2  3
#  13.75 13.25 13
```

`shapley_mfoc` *SMFOC*

Description

Function to calculate the Shapley value for the associated game to a fixed order cost model.

Usage

```
shapley_mfoc(n = NA, a = NA, d = NA, K = NA)
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
K	Vector. Warehouse's capacity to each agent.

Value

Shapley value for the associated game to the fixed order cost model.

Examples

```
shapley_mfoc(n=5,a=200,d=c(1.2,1.4,1.2,1.3,0.4),K=c(6,10,8,8,4))

#Shapley-Value
#MFOC model
#[1]  4.000000  6.000000  6.666667  7.916667 15.416667
```

SOC *SOC*

Description

Generic function for showing the allocations proposed by SOC rule under an EOQ or EPQ model.

Usage

```
SOC(n = NA, a = NA, d = NA, h = NA, m = NA, r = NA, b = NA,
model = c("EOQ", "EPQ"))
```

Arguments

n	Number of agents in the inventory model.
a	The fixed cost per order.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding costs to each agent.
m	Vector. Number of orders to each agent(optional).
r	Vector. Replacement rate to each agent. In general, r>d.
b	Vector. Cost of a shortage to each agent.
model	Model to select. `EOQ` and `EPQ` models can be considered.

Value

Matrix with number of rows equal to the number of coalitions and *n* columns. For each coalition or row, the output shows the cost that SOC rule allocates to each player or column.

Examples

```
SOC(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10),m=NA,r=NA,b=NA,model="EOQ")

#EOQ model
#$`Share the ordering costs rule (individually)`
#[1] 2400.000 1989.975 2190.890

SOC(n=3,a=600,d=c(500,300,400),h=c(9.6,11,10),m=NA,r=NA,b=NA,model="EOQ")

#EOQ model
#$`Share the ordering costs rule (individually)`
#         1        2        3
#     0.000    0.000    0.000
#  2400.000    0.000    0.000
#     0.000 1989.975    0.000
#     0.000    0.000 2190.890
#  1847.521 1270.171    0.000
#  1772.517    0.000 1477.098
#     0.000 1337.960 1621.770
#  1511.608 1039.230 1259.673
```

STI *STI*

Description

This function obtains the optimal orders and the associated cost when agents are cooperating in the inventory transportation system.

Usage

```
STI(n = NA, a = NA, av = NA, d = NA, h = NA, m = NA)
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
av	Vector. The transportations cost per order to each agent.
d	Vector. Deterministic demands per time unit to each agent.
h	Vector. Holding cost per time unit to each agent.
m	Vector. Number of orders to each agent (optional).

Value

This function calculates two vectors. The first one shows the optimal order for each agent. The second vector indicates the associated cost to these orders.

Examples

```
STI(n=3,a=200,av=c(300,300,900),d=c(90,80,20),h=c(0.06,0.06,0.1),m=NA)
#STI model
#$"Optimal order"
#[1] 1224.745 1154.701  663.325
#
#$"Order cost"
#[1] 73.48469 69.28203 66.33250
```

`STIcoo` *STIcoo*

Description

This function obtains the optimal orders and the associated cost when agents are cooperating in the inventory transportation system when agents are cooperating.

Usage

```
STIcoo(n = NA, a = NA, av = NA, d = NA, h = NA, m = NA)
```

Arguments

`n`	Agents in the inventory situation.
`a`	The fixed cost per order.
`av`	Vector. The transportations cost per order to each agent.
`d`	Vector. Deterministic demands per time unit to each agent.
`h`	Vector. Holding cost per time unit to each agent.
`m`	Vector. Number of orders to each agent (optional).

Value

A list with the following components:

- `Optimal order` If *n* is lesser than *0*, a matrix with all possible coalitions in the first column. The next *n* columns contain the associated cost to each agent in the coalition. Last column indicates the global cost of the optimal order. Otherwise, this matriz contains the individual costs and the associated values for *N*.

Examples

```
STIcoo(n=3,a=200,av=c(300,300,900),d=c(90,80,20),h=c(0.06,0.06,0.1),m=NA)

#$`Optimal order`
#          1        2        3   Coalition Order cost
#1    0.0000    0.000   0.0000           0    0.00000
#2 1224.7449    0.000   0.0000     '{ 1 }'   73.48469
#3    0.0000 1154.701   0.0000     '{ 2 }'   69.28203
#4    0.0000    0.000 663.3250     '{ 3 }'   66.33250
#5  891.1328  792.118   0.0000   '{ 1,2 }'  100.99505
#6 1551.8080    0.000 344.8462   '{ 1,3 }'  127.59310
#7    0.0000 1438.954 359.7385   '{ 2,3 }'  122.31108
#8 1208.5759 1074.290 268.5724 '{ 1,2,3 }'  163.82918
```

`twolines` *Two-lines rule*

Description

Two-lines rule for a basic EOQ system without holding costs and with transportation cost.

Usage

```
twolines(n = NA, a = NA, av = NA, d = NA, K = NA)
```

Arguments

n	Agents in the inventory situation.
a	The fixed cost per order.
av	Vector. The transportations cost per order to each agent.
d	Vector. Deterministic demands per time unit to each agent.
K	Vector. Capacities of agents' warehouse to each agent.

Value

The output is a vector who contains the allocation to each player.

Examples

```
twolines(n=3,a=400,av=c(300,500,200),d=c(2,2,5),K=c(9,8,7))
#MCT model
#Cooperative case
#Two-lines rule
#[1]   0.0000 219.6429 423.2143
```

Index