Отчёт к лабораторной работе

по дисциплине  
«Теория принятия решений»

выполнил   
студент гр. ИС/б-18-1-з Демиденко А. А.  
принял Кротов К. В.

Лабораторной работа № 5  
«Исследование применения теории важности критериев для решения задачи выбора альтернатив»

## Цель работы

Исследовать применение теории важности критериев для решения задачи выбора альтернатив.

## Программа выполнения работы

Для первого варианта задания предусматривается следующий порядок действий по выполнению лабораторной работы:

1. на основе информации Ω о качественной важности критериев сформировать матрицы А1 и А2 отношений предпочтения и эквивалентности для критериев ;
2. разработать процедуру определения доминируемых решений, выполняющую для каждого решения xl сравнение его значений скалярных оценок kli вектора K(xl) с такими же скалярными оценками khi решений xh; тем самым должны быть определены решения xh, доминируемые текущим рассматриваемым решением xl (при и h ≠ l); результатом выполнения процедуры является множество XΩ не сравнимых между собой с использованием отношения предпочтения ≻ решений;
3. разработать процедуру, использующую информацию Ω о важности критериев, входными данными для которой будет являться матрица А1 отношения предпочтения для критериев; разрабатываемая процедура должна выполнять следующие операции:
   1. для решений xl (при ) формировать новые векторные оценки Kji (xl) путем перестановки скалярных компонент kli и klj в исходной векторной оценке K(xl) (индексы i и j соответствуют критериям Ki и Kj , связанным отношением предпочтения в следующем виде: Ki ≻ Kj);
   2. для каждого решения xl для его модифицированных векторных оценок Kji (xl) проконтролировать выполнение условия доминирования им других решений xh для их векторных оценок Kji (xh) (при и h ≠ l ) (т.е. выполняется поэлементное сравнение оценок kli и khi из соответствующих векторов Kji (xl) и Kji (xh)); при выполнении условия xl ≻ xh, процедура реализует исключение решения xh из множества XΩ : XΩ = XΩ \ xh;
   3. результатом выполнения разрабатываемой программы является определение множества не сравнимых решений XΩ, сформированного на основе информации Ω о предпочтениях критериев вида Ki ≻ Kj;
4. разработать процедуру, использующую информацию Ω о важности критериев, входными данными для которой будет являться матрица А2 отношения эквивалентности для критериев; разрабатываемая процедура должна выполнять следующие операции:
   1. для решений xl (при ) формировать новые векторные оценки Kji (xl) путем перестановки скалярных компонент kli и klj в исходной векторной оценке K(xl) (индексы i и j соответствуют критериям Ki и Kj , связанным отношением эквивалентности в следующем виде: Ki ~ Kj);
   2. для модифицированных векторных оценок Kji (xl) каждого решения xl (при ) проконтролировать выполнение условия доминирования им других решений xh для их векторных оценок Kji (xh) (при и h ≠ l) (т.е. выполняется поэлементное сравнение оценок kli и khi из соответствующих векторов Kji (xl) и Kji (xh)); при выполнении условия xl ≻ xh, процедура реализует исключение решения xh из множества XΩ : XΩ = XΩ \ xh;
   3. результатом выполнения разрабатываемой программы является определение множества не сравнимых решений XΩ, сформированного на основе информации Ω об эквивалентности критериев вида Ki ~ Kj;
5. выполнить вывод множества XΩ, полученного в результате исключения из него доминируемых решений xh при учете дополнительной информации Ω о предпочтениях и эквивалентности критериев.

В качестве исходных данных для выполнения задания по лабораторной работе (для всех вариантов) заданы: множество решений вида X = {x1, x2, x3, x4, x5, x6, x7, x8}, оценки пяти критериев сведены в таблицу 4.



**Вариант 1.** Определить множество несравнимых решений XΩ, используя качественную информацию о важности критериев Ω в следующем виде: Ω = {K1≻ K2, K2 ~ K3, K3 ~ K4, K4 ≻ K5}.

## Ход работы

Реализуем объявление и инициализацию матрицы со значениями критериев:

type Row a = [a]

type Matrix a = [Row a]

input :: Matrix Int

input =

[ [ 3, 5, 5, 4, 4 ]

, [ 4, 4, 4, 5, 4 ]

, [ 5, 4, 3, 3, 5 ]

, [ 3, 5, 3, 5, 3 ]

, [ 4, 2, 4, 5, 5 ]

, [ 3, 5, 3, 5, 3 ]

, [ 5, 3, 4, 3, 4 ]

, [ 4, 5, 3, 4, 3 ]

]

Определим матрицы А1 и А2:

inputA1 :: Matrix Bool

inputA1 =

[ [ False, True, False, False, False ]

, [ False, False, False, False, False ]

, [ False, False, False, False, False ]

, [ False, False, False, False, True ]

, [ False, False, False, False, False ]

]

inputA2 :: Matrix Bool

inputA2 =

[ [ False, False, False, False, False ]

, [ False, False, True, False, False ]

, [ False, False, False, True, False ]

, [ False, False, False, False, False ]

, [ False, False, False, False, False ]

]

Реализуем функцию, удаляющую доминируемые решения:

import Data.List (delete)

checkDominated :: Ord a => Row a -> Row a -> Bool

checkDominated a b = all (== True) (zipWith (>=) a b) && or (zipWith (>) a b)

cleanDominated :: Matrix Int -> Matrix Bool -> Matrix Bool -> Row Int

cleanDominated matrix a1 a2 = calculate (zip [1..] matrix) [1 .. length matrix]

where

calculate [] p = p

calculate ((i, m):ms) p = calculate ms $

foldr (\(l, row) result ->

if checkDominated m row

then delete l result

else result

) p ms

Реализуем функцию, учитывающую относительную важность критериев:

get :: Row a -> Int -> a

get s i = s !! (i - 1)

replace :: Int -> a -> [a] -> [a]

replace \_ \_ [] = []

replace 1 val (\_:xs) = val : xs

replace n val (x:xs) = x : replace (n - 1) val xs

swap :: (Int, Int) -> Row a -> Row a

swap (i, j) x = replace j (get x i) $ replace i (get x j) x

getOmega :: Matrix Bool -> Row (Int, Int)

getOmega = foldr (\(i, row) result ->

if or row

then filter snd (zip [1..] row) >>= \j -> (i, fst j) : result

else result

) [] . zip [1..]

cleanPreferable :: Matrix Int -> Matrix Bool -> Row Int

cleanPreferable matrix a1 = calculate (getOmega a1) $ cleanDominated matrix

where

matrixI = zip [1..] matrix

calculate [] p = p

calculate (s@(j, \_):ss) p = calculate ss $

foldr (\i result ->

let x = get matrix i; x' = swap s x

in if get x j > get x' j

then foldr (delete . fst) result $

filter (\(i', y) -> i /= i' && checkDominated x' y) matrixI

else result

) p p

Реализуем учёт матрицы эквивалентности и окончательный поиск решения:

import Data.List (nub)

findSolution :: Matrix Int -> Matrix Bool -> Matrix Bool -> Row Int

findSolution matrix a1 a2 = calculate (getOmega a2) $ cleanPreferable matrix a1

where

matrixI = zip [1..] matrix

calculate [] p = p

calculate (s@(j, \_):ss) p = calculate ss $

foldr (\i result ->

let x = swap s $ get matrix i

in foldr (delete . fst) result $

filter (\(i', y) -> i /= i' && checkDominated x y) matrixI

) p p

λ> findSolution input inputA1 inputA2

[1,2,3,5,8]

## Выводы

В ходе лабораторной работы была рассмотрена задача принятия решения при помощи теории важности критериев. Для решения задачи был написан ряд функций на языке Haskell.