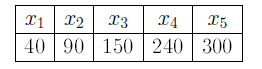
1. Теоретическая часть

Вектор Шепли — это один из возможных подходов к решению проблемы справедливого дележа. Мы рассмотрим его на примере модели «Малое Гадюкино». Село Малое Гадюкино примыкает к автомобильной дороге. Все n домов в этом селе стоят на одной-единственной улице и находятся на расстояниях x1, x2, . . . , xn метров от автодороги. Жители села решили заасфальтировать эту улицу. Каждый метр асфальта стоит 1. Как жителям распределить взносы? Мы можем с самого начала считать, что все расстояния упорядочены, то есть x1 < x2 < . . . < xn. Каждый из Гадюкинцев хотел бы заплатить поменьше. Вообще, каждый из Гадюкинцев хотел бы участвовать в финансировании дороги только до своего дома, потому что остальной частью улицы он не пользуется. Тем не менее общие интересы требуют асфальтирования всей улицы. Концепция вектора Шепли предлагает такой рецепт. Допустим, что в число плательщиков все жители деревни добавляются в некотором определенном порядке. Если дорога до вновь добавившегося жителя не оплачена, он оплачивает строительство участка от уже построенного ранее до его дома. Если дорога до его дома уже оплачена, то он не платит ничего. Всех n жителей можно расположить в некотором порядке n! способами. Набор усредненных по всем n! перестановкам платежей для всех жителей деревни и есть вектор Шепли. Рассмотрим пример. Пусть n = 5, а расстояния до дороги приведены в следующей таблице



Рассмотрим перестановку (42153). Первым в число плательщиков добавляется четвертый житель, и он оплачивает 240 метров дороги. Следующими добавляются второй и первый жители, и они не доплачивают ничего, потому что дорога до них уже про финансирована. Следующим добавляется пятый житель, и он оплачивает оставшиеся до него 60 метров дороги. Третий житель также ничего не платит.

Рассмотрим теперь перестановку (21534). В этом случае второй оплачивает 90 метров, пятый оплачивает 210 метров, а остальные не платят ничего. Для других перестановок платежи, разумеется, будут другие. Из 5 объектов можно составить 5! = 120 перестановок. Усредненные по всем этим перестановкам платежи и есть вектор Шепли.

1. Программный код

"""

Программа, вычисляющая вектор Шепли

Автор: Афанасьев И.Е.

Дата написания: 20.09.2020

"""

# импортируем перестановки

from itertools import permutations

# функция, вычисляющая факториал числа

def fact(i):

if i <= 1:

return 1

return i \* fact(i - 1)

# Задаём расстояния до домов ("Малое Гадюкино")

X = [40, 90, 150, 240, 300]

# Порядковые номера домов

X\_i = [i for i in range(len(X))]

# Факториал от числа домов

factX = fact(len(X))

# Перестановки всех домов (И дублирование для номеров домов)

perm = permutations(X)

perm\_i = list(permutations(X\_i))

# Характеристическая функция

hf = []

for i in perm:

hf1 = []

for j in range(len(i)):

if j == 0:

hf1 += [i[j]]

else:

# Отражает вклад каждого жителя в постройку дороги

hf1 += [i[j] - max(i[:j]) if i[j] - max(i[:j]) > 0 else 0]

hf += [hf1]

# Инициализируем вектор Шепли

shapley = [0]\*len(X)

# Пробегаемся по всем перестановкам

for i in range(factX):

for j in range(len(X)):

for k in range(len(shapley)):

if perm\_i[i][j] == k:

shapley[k] += hf[i][j]

# Усредняем

for i in range(len(shapley)):

shapley[i] /= factX

print(shapley)

1. Результаты работы программы

При X = [40, 90, 150, 240, 300]

Шепли:

[8.0, 20.5, 40.5, 85.5, 145.5]

При X = [40, 90, 150, 240, 300, 500]

Шепли:

[6.666666666666667, 16.666666666666668, 31.666666666666668, 61.666666666666664, 91.66666666666667, 291.6666666666667]