

Министерство науки и высшего образования Российской Федерации
Московский государственный технический университет имени
Н. Э. Баумана

Факультет: Информатика и системы управления
Кафедра: Информационная безопасность (ИУ8)

Рубежный контроль №3
ПО ДИСЦИПЛИНЕ «ТЕОРИЯ ИГР И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ»
«Кооперативные игры. Вектор Шепли»

Вариант 1

Студент: Анаян М. С., ИУ8-104
Преподаватель: Коннова Н. С.

Москва, 2020

Цель и задачи выполнения. Исходные данные

Цель работы – Изучить постановку кооперативной игры и найти оптимальное распределение выигрыша (дележ) между игроками путем вычисления компонент вектора Шепли.

Постановка задачи

1. Проверить кооперативную игру на супераддитивность и выпуклость.
2. Составить программу вычисления компонент вектора Шепли и, в зависимости от варианта, рассчитать его.
3. Проверить условия индивидуальной и групповой рационализации.

Исходные данные

ХФ	1
$v(\emptyset)$	0
$v(\{1\})$	4
$v(\{2\})$	1
$v(\{3\})$	3
$v(\{4\})$	1
$v(\{1, 2\})$	6
$v(\{1, 3\})$	8
$v(\{1, 4\})$	6
$v(\{2, 3\})$	5
$v(\{2, 4\})$	3
$v(\{3, 4\})$	5
$v(\{1, 2, 3\})$	9
$v(\{1, 2, 4\})$	8
$v(\{1, 3, 4\})$	10
$v(\{2, 3, 4\})$	7
$v(I)$	11

Выполнение лабораторной работы

Заданная по условию варианта кооперативная игра является супераддитивной, но не является выпуклой (поскольку для наборов $\{1, 2\}$ и $\{1, 3\}$: $v(\{1,2,3\}) + v(\{1\}) < v(\{1,2\}) + v(\{1,3\})$, $9 + 4 < 6 + 8$).

Вычисленный вектор Шепли:

$$X(v) = (4.42, 1.42, 3.42, 1.75).$$

Для него выполняется условие группового рационализма:

$$\sum_{i \in I} x_i(v) = v(I) = 11,$$

а также выполняется условие индивидуального рационализма:

$$x_i(v) \geq v(\{i\}), i \in I.$$

Исходный код на Python 3.* для решения задачи представлен в Приложении.

Выводы

В ходе лабораторной работы получены следующие результаты:

- Заданная условием игра – супераддитивная, но не выпуклая;
- Составлена программа вычисления компонент вектора Шепли, полученный результат: $X(v) = (4.42, 1.42, 3.42, 1.75)$;
- Выполнена проверка полученного вектора на условия индивидуальной и групповой рационализации.

Приложение

Исходный код на языке Python 3.* также представлен в репозитории по ссылке: <https://github.com/hms2010/GameTheory/tree/master/src/rk3>

```
from game_params import *
from math import factorial
from fractions import Fraction

def is_supperadditive(chars):
    for i in chars:
        for j in chars:
            if (i & j) == 0:
                if chars[i | j] < (chars[i] + chars[j]):
                    return False
    return True

def is_convex(chars):
    for i in chars:
        for j in chars:
            if (chars[i | j] + chars[i & j]) < (chars[i] + chars[j]):
                return False
    return True

def sets_power(s):
    res = 0
    while s:
        res += s & 0x01
        s >>= 1
    return res

def shapley_value(chars):
    vec = []
    for i in range(N):
        res = Fraction(0)
        for coal in chars:
            if (coal & 2**i) == 0:
                continue
            coal_power = sets_power(coal)
            res += (factorial(coal_power - 1) *
                    factorial(N - coal_power) *
                    (chars[coal] - chars[coal & ~(2**i)]))
        res /= factorial(N)
        vec.append(res)
    return vec

def check_group_rationalization(vec):
    return sum(vec) == chars[2**N - 1]

def check_individual_rationalization(vec):
    for i in range(N):
        if vec[i] < chars[2**i]:
            return False
    return True
```

```

def main():
    if is_supperadditive(chars):
        print("Game is supperadditive")
    else:
        print("Game isn't supperadditive")

    if is_convex(chars):
        print("Game is convex")
    else:
        print("Game isn't convex")

    shapley_vector = shapley_value(chars)
    print("Shapley vector: [{:s}]" .format(", ".join(["{: .2f}]" .format(float(i)) for i in shapley_vector])))

    if check_group_rationalization(shapley_vector):
        print("Group rationalization: OK")
    else:
        print("Group rationalization: FAILED")

    if check_individual_rationalization(shapley_vector):
        print("Individual rationalization: OK")
    else:
        print("Individual rationalization: FAILED")

if __name__ == "__main__":
    main()

```