**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по практической работе №2**

**по дисциплине «Теория принятия решений»**

Тема: Бесконечные антагонистические игры

|  |  |  |
| --- | --- | --- |
| Студентка гр. 8383 |  | Сырцова Е.А. |
| Преподаватель |  | Попова Е.В. |

Санкт-Петербург

2022

**Цель работы**

Использование инструментальных средств для решения задач поддержки принятия решения, а также овладение навыками принятия решения на основе бесконечных антагонистических игр.

**Основные теоретические положения**

В данной работе рассматриваются антагонистические игры, которые отличаются от матричных тем, что в них один или оба игрока имеют бесконечное (счётное или континуум) множество стратегий. С теоретико-игровой точки зрения это отличие малосущественно, поскольку игра остаётся антагонистической и проблема состоит в использовании более сложного аналитического аппарата исследования.

Таким образом, исследуются общие антагонистические игры, т.е. системы вида

|  |  |
| --- | --- |
|  |  |

где и – произвольные бесконечные множества, элементы которых являются стратегиями игроков 1 и 2 соответственно, а – функция выигрыша игрока 1. Выигрыш игрока 2 в ситуации равен (игра антагонистическая). Далее рассматриваются такие игры, у которых функция ограничена.

*Одновременная игра преследования на плоскости*

Пусть *S1* и *S2* – множества на плоскости. Игра заключается в следующем. Игрок 1 выбирает некоторую точку , а игрок 2 выбирает точку . При совершении выбора игроки 1 и 2 не имеют информации о действиях противника, поэтому подобный выбор удобно интерпретировать как одновременный. В этом случае точки являются стратегиями игроков 1 и 2 соответственно. Таким образом, множества стратегий игроков совпадают с множествами *S1* и *S2* на плоскости.

Целью игрока 2 является минимизация расстояния между ним и игроком 1 (игрок 1 преследует противоположную цель). Поэтому под выигрышем игрока 1 в этой игре понимается евклидово расстояние между точками и , т.е.

|  |
| --- |
|  |

Выигрыш игрока 2 полагаем равным выигрышу игрока 1, взятому с обратным знаком, а именно (игра антагонистическая).

*Модель покера с одним кругом ставок и одним размером ставки*

В начале партии каждый из двух игроков *A* и *B* ставит по единице. После того, как каждый из игроков получит карту, ходит игрок *A*: он может или поставить ещё *c* единиц или спасовать и потерять свою начальную ставку. Если *A* ставит, то у *B* две альтернативы: он может или спасовать (теряя при этом свою начальную ставку), или уровнять, поставив *c* единиц. Если *B* уравнивает, то игроки открывают свои карты и игрок с лучшей картой выигрывает единицу (банк).

Обозначим карту игрока *A* через , а карту игрока *B* через , при этом предполагаем, что случайные величины и имеют равномерное распределение на единичном интервале.

Стратегии строятся следующим образом. Пусть

* – вероятность того, что если *A* получит , то *A* поставит *c*,
* – вероятность того, что если *A* получит , то *A* спасует,
* – вероятность того, что если *B* получит , то *B* уравняет ставку *c*,
* – вероятность того, что если *B* получит , то *B* спасует.

Средний выигрыш *A* *H* определяется соотношениями:

-1 с вероятностью

+1 с вероятностью

с вероятностью

Первый игрок максимизирует свой выигрыш, второй игрок минимизирует свой проигрыш.

**Постановка задачи**

Используя инструментальные средства компьютерной алгебры решить задачи преследования и покера.

Порядок выполнения работы:

1. Для задачи преследования отобразить фигуры на плоскости с помощью инструментального средства или вручную.
2. Рассмотреть два случая задачи: центр масс фигуры *S1* принадлежит фигуре *S2* и центр масс фигуры *S1* не принадлежит фигуре *S2*.
3. Решить задачу аналитически и с помощью программы.
4. Решить задачу игры в покер аналитически и с помощью программы.

Найти выигрыши и оптимальные стратегии для двух типов оптимальных стратегий.

Для задачи одновременной игры преследования на плоскости множества *S1* – окружность с радиусом *R*, *S2* – равнобедренный треугольник с основанием *a* и высотой *h*.

Для задачи игры в покер с одним кругом ставок и одним размером ставки, размер ставки *c* = 23.

**Выполнение работы**

Для решения задач одновременной игры преследования на плоскости была написана программа на языке программирования Python, исходный код которой приведен в приложении A.

|  |
| --- |
| Нижнее значение цены игры: |

Пусть и . Тогда минимум достигается в точке , являющейся точкой пересечения перпендикуляра, проведенного из точки , к границе треугольника . Величина достигается в точке лежащей на границе круга , через его центр.

Таким образом нижнее значение игры,

|  |  |
| --- | --- |
|  |  |

Графическое изображение рассуждений представлено на рисунке:

***S*2**

***O*2**

*M*1

*y*

*y*0

***O*1**

*x*0

***S*1**

*M*

*x*

Для вычисления верхнего значения игры рассмотрено два случая.

1. Центр масс фигуры принадлежит фигуре .

Для любой точки , принадлежащей максимальным расстоянием до некоторой точки , принадлежащей , будет являться прямая проходящая через центр круга до диаметрально противоположной стороны круга. Для того, чтобы данное расстояние было при этом минимально возможным, необходимо чтобы совпадала с центром круга .

Исходя из всего вышесказанного:

|  |  |
| --- | --- |
| . |  |

Таким образом, верхнее и нижнее значение игры совпадут, когда будет лежать на границе треугольника. Графическое изображение рассуждений представлено на рисунке:

***S*2**

***O*2**

*y*0

*y*

***O*1**

*x*0

***S*1**

*M*

*x*

1. Центр масс фигуры *S1* не принадлежит фигуре *S2*.

Для любой точки , принадлежащей *S2* максимальным расстоянием до некоторой точки , принадлежащей *S1*, будет являться прямая проходящая через центр круга до диаметрально противоположной стороны круга. Для того, чтобы данное расстояние было при этом минимально возможным, необходимо чтобы совпадала с концом перпендикуляра, проведенного через центр круга к стороне треугольника (точкой ).

Исходя из всего вышесказанного:

|  |
| --- |
| . |

Графическое изображение представлено на рисунке:

***S*2**

***O*2**

*y*0

*M*1

*y*

***O*1**

***S*1**

*M*

*x*0

*x*

Решим задачу аналитически и с помощью программы.

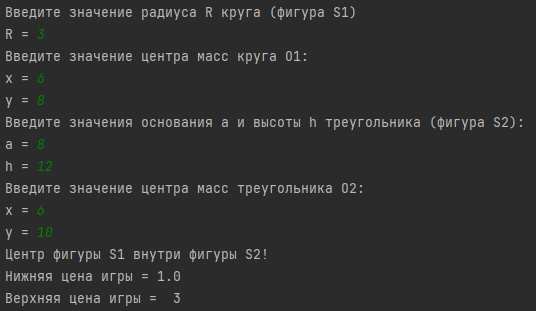
1. Центр масс фигуры принадлежит фигуре

Например, , .

Аналитически: ,

.

С помощью программы:



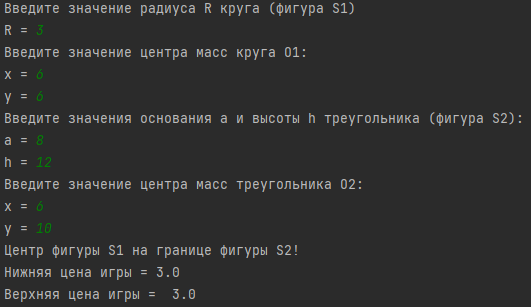
1. Центр масс фигуры принадлежит границе фигуры

Например, , .

Аналитически: ,

.

С помощью программы:



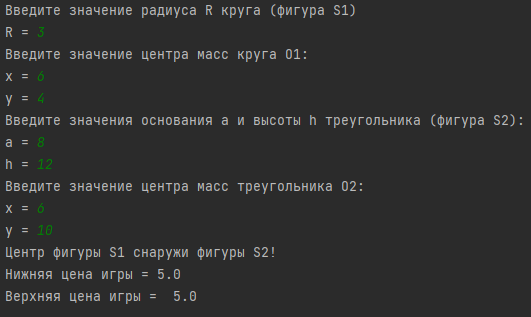
1. Центр масс фигуры не принадлежит фигуре .

Например,

Аналитически: ,

.

С помощью программы:



Рассмотрим задачу игры в покер с одним кругом ставок и одним размером ставки, размер ставки *c* = 23.

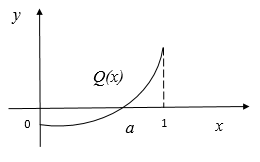
Для решения игры в покер с одним кругом ставок и одним размером ставки была написана программа на языке программирования Python, исходный код которой приведен в приложении Б.

Решим задачу аналитически:

Средний выигрыш *A:*

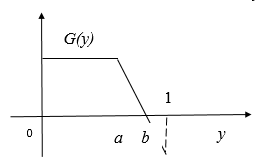
Первый игрок максимизирует свой выигрыш

– может принимать любые значения



Второй игрок минимизирует свой проигрыш

*A* использует стратегию с порогом *a*. Проигрыш *B*

**

Найдем *b*,

Минимальное значение проигрыша *B*

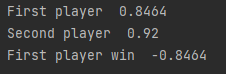
Подставим *b*,

– стратегия *A*. Постарается минимизировать проигрыш *B*. Нужно найти максимум параболы

Оптимальные стратегии игроков для модели покера с одним кругом ставок и одним размером ставки

Значение среднего выигрыша первого игрока

Вычисления с помощью программы:



При использовании стратегии с блефом.

Так как второй игрок не может блефовать, то

Стратегия игрока *B*

Для найденного *b* найдем *Q(x)*

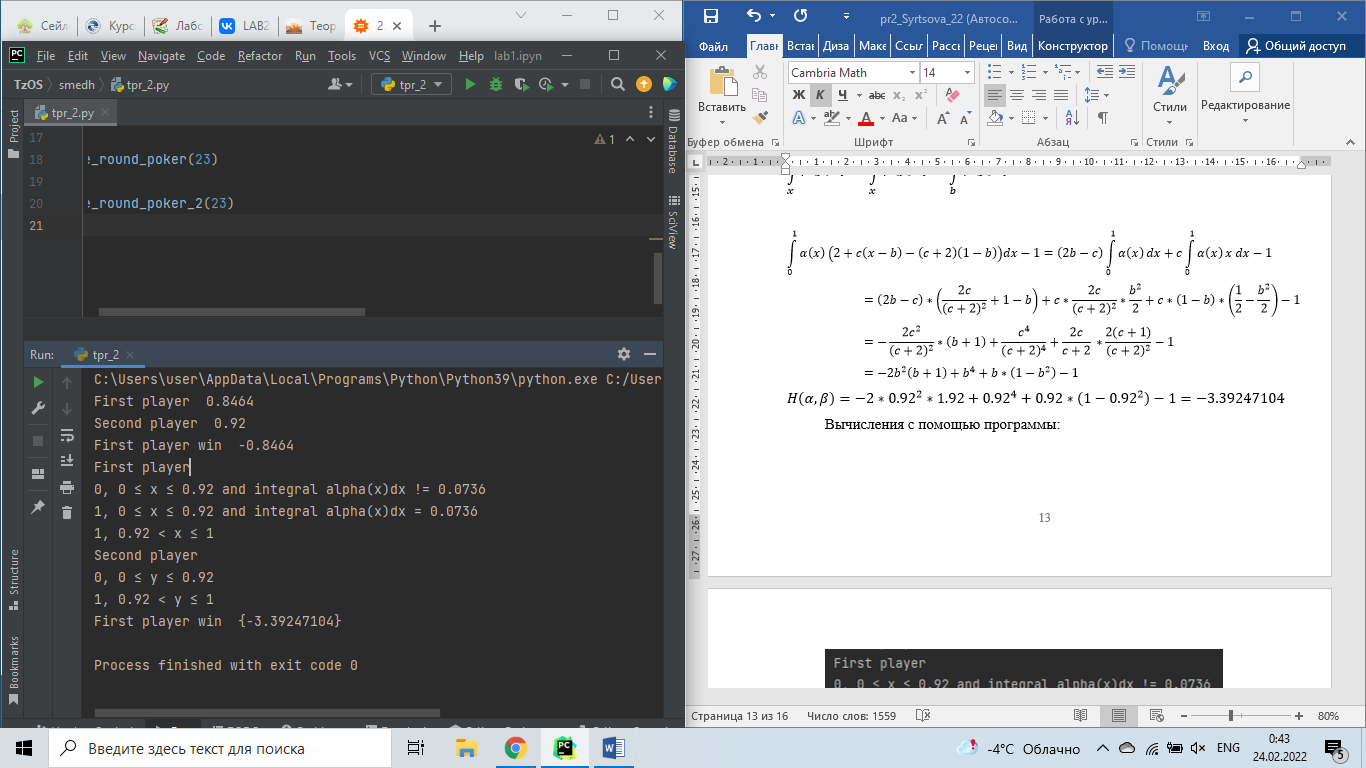
Получили:

Т.е. игроку *A* следует блефовать при условии что

Стратегия игрока *A*

Выигрыш первого игрока:

Вычисления с помощью программы:



**Выводы**

В процессе выполнения практической работы были разработаны программы на языке программирования Python для решения игры одновременного преследования на плоскости и игры в покер с одним кругом ставок и одним размером ставки. Для игры одновременного преследования с помощью программы были вычислены границы игры для случаев, когда центр масс фигуры *S1* принадлежит фигуре *S2*, центр масс фигуры *S1* не принадлежит фигуре *S2*. Для игры в покер аналитически были найдены оптимальные стратегии игроков, также с помощью программы были вычислены оптимальные стратегии игроков для указанного размера ставки.

Приложение А

Исходный код программы для решения игры одновременного преследования на плоскости

import math  
  
print("Введите значение радиуса R круга (фигура S1)")  
r = int(input("R = "))  
print("Введите значение центра масс круга O1:")  
x1 = int(input("x = "))  
y1 = int(input("y = "))  
print("Введите значения основания a и высоты h треугольника (фигура S2):")  
a = int(input("a = "))  
h = int(input("h = "))  
print("Введите значение центра масс треугольника O2:")  
x2 = int(input("x = "))  
y2 = int(input("y = "))  
while x2 != x1:  
 print("Введите значение центра масс круга O1:")  
 x1 = int(input("x = "))  
 y1 = int(input("y = "))  
 print("Введите значение центра масс треугольника O2:")  
 x2 = int(input("x = "))  
 y2 = int(input("y = "))  
while (r \* r \* math.pi >= 0.5 \* a \* h):  
 print("Площадь круга должна быть меньше площади треугольника!")  
 print("Введите значение радиуса R круга (фигура S1)")  
 r = int(input("R = "))  
 print("Введите значения основания a и высоты h треугольника (фигураS2) :")  
 a = int(input("a = "))  
 h = int(input("h = "))  
  
if (abs(y1 - y2) < h / 3):  
 print("Центр фигуры S1 внутри фигуры S2! ")  
 print("Нижняя цена игры =", y2-y1+r-h/3)  
 print("Верхняя цена игры = ", r)  
elif (abs(y1 - y2) == h / 3):  
 print("Центр фигуры S1 на границе фигуры S2! ")  
 print("Нижняя цена игры =", y2-y1+r-h/3)  
 print("Верхняя цена игры = ", y2-y1+r-h/3)  
else:  
 print("Центр фигуры S1 снаружи фигуры S2! ")  
 print("Нижняя цена игры =", y2-y1+r-h/3)  
 print("Верхняя цена игры = ", y2-y1+r-h/3)

Приложение Б

Исходный код программы для решения игры В покер с одним кругом ставок и одним размером ставки

### def one\_round\_poker(c): print("First player ", pow(c, 2)/pow(c+2, 2)) print("Second player ", c/(c+2)) print("First player win ", -pow(c, 2)/pow(c+2, 2)) def one\_round\_poker\_2(c): print("First player") print(f"0, 0 ≤ x ≤ {c / (c + 2)} and integral alpha(x)dx != {2 \* c / pow(c+2, 2)}\n" f"1, 0 ≤ x ≤ {c / (c + 2)} and integral alpha(x)dx = {2 \* c / pow(c+2, 2)}\n" f"1, {c / (c + 2)} < x ≤ 1") print("Second player") print(f"0, 0 ≤ y ≤ {c / (c + 2)}\n" f"1, {c / (c + 2)} < y ≤ 1") print("First player win ", {-2\*pow(c / (c + 2), 2)\*(1 + (c / (c + 2))) + pow(c / (c + 2), 4) + (c / (c + 2))\*(1 - pow(c / (c + 2), 2)) - 1}) one\_round\_poker(23) one\_round\_poker\_2(23)