Санкт-Петербургский государственный университет
Кафедра математической теории игр и статистических решений

Пуртян Александра Сергеевна

Выпускная квалификационная работа бакалавра

Равновесие по Нэшу в одной игре преследования со многими участниками

Направление 010400
Прикладная математика, фундаментальная информатика
и основы программирования

Научный руководитель,
кандидат физ.-мат. наук,
ассистент
Панкратова Я. Б.

Санкт-Петербург
2017

Содержание

Введение

Теория дифференциальных игр позволяет формализовать задачи управления с бесконечным множеством альтернатив. Одним из наиболее часто рассматриваемых классов задач являются дифференциальные игры преследования (в частности на быстродействие), широко применяемые в описании задач военного характера. В случае с одним убегающим и одним преследователем речь идет о классических антагонистических дифференциальных играх преследования, подробно описанных в работе Р.Айзекса [1], если же в игре участвует большее количество игроков, то речь идет о неантогонистических дифференциальных играх преследования. Основопологающий вклад в изучение которых внесли Л.А.Петросян и В.Д. Ширяев [5], впервые сформулировав игру преследования с одним преследователем и двумя убегающими. Дальнейшее развитие данная тема получила в работах С.И Тарашниной [7].

Несмотря на то, что теория дифференциальных игр возникла из-за необходимости решения задач военного характера, применение теории дифференциальных игр отнюдь не ограничивается данной сферой, и сейчас все больше развивается интерес к подобной формализации задач экономики и других областей человеческой деятельности. Так, понимая под встречей возможность обмена некой информацией или товаром, можно описать в виде неантогонистической дифференциальной игры основные задачи логистики.

Таким образом, рассматриваемая в данной работе неантогонистическая игра, в которой нет убегающего, а все игроки преследуют цель наискорейшей встречи, описывает актуальную задачу и представляет практический интерес.

Постановка задачи

В данной работе были поставлены следующие задачи:

1. Изучение теории дифференциальных игр преследования;
2. Применение этой теории для решения задач о встречи;
3. Программная реализация нахождения ситуации равновесия по Нэшу в неантагонистической игре n лиц.

Глава 1. Неантагонистическая игра n лиц

1.1. Неантагонистическая игра преследования

Рассмотрим дифференциальную игру преследования с одним убегающим S и n преследователями $C_1, C_2, \ldots, C_n$. Игроки перемещаются на плоскости с постоянными по модулю скоростями β и $\alpha_1, \alpha_2, \ldots, \alpha_n$ соответственно. Причем $\beta < \min\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

Игра начинается в момент времени $t_0 = 0$ из начальных положений игроков:

$$S^0 = \{x^0, y^0\} = z^0;\ C_1^0 = \{x_1^0, y_1^0\} = z_1^0,\ C_2^0 = \{x_2^0, y_2^0\} = z_2^0, \ldots, C_n^0 = \{x_n^0, y_n^0\} = z_n^0.$$

В каждый момент времени $t \geq 0$ игроки имеют возможность изменить направление своего движения, выбрав вектор скорости u_S^t из своего множества допустимых управлений. Множество допустимых управлений определяется следующим образом

$$U_{C_j} = \{\mathrm{u}_{C_j} = (u_{C_j}^1, u_{C_j}^2) : (u_{C_j}^1)^2 + (u_{C_j}^2)^2 = \alpha_j^2\}, \quad j = 1, \ldots, n$$
$$U_S = \{\mathrm{u}_S = (u_S^1, u_S^2) : (u_S^1)^2 + (u_S^2)^2 = \beta^2\}.$$

Движение игроков можно описать системой дифференциальных уравнений

$$\begin{aligned} &\dot{z} = \mathrm{u}_S, \ \ \mathrm{u}_S \in U_S, \\ &\dot{z}_j = \mathrm{u}_{C_j}, \ \ \mathrm{u}_{C_j} \in U_{C_j}, \ \ j = 1, \ldots, n. \end{aligned} \tag{1}$$

с начальными условиями $z(0) = z^0, z_1(0) = z_1^0, \ldots, z_n(0) = z_n^0$.

Будем рассматривать случай полной информации, то есть все игроки в каждый момент времени t знают свое местоположение и местоположения остальных игроков. А игрокам C_j $(j = 1, \ldots, n)$, кроме того, известно направление движения игрока S в каждый момент времени (то есть значение параметра u_S^t).

Стратегия игрока должна определять его поведение в любой момент игры в любом его местоположении, таким образом стратегией S является функция $u_S(t, z_1(t), \ldots, z_n(t), z(t)) \triangleq u_S$. Эта функция зависит от времени, местоположений игроков и удовлетворяет системе (1). Множество допустимых стратегий S обозначим $\mathcal{U}_S$. Учитывая условие полной информации, стратегии игрока C_j $(j = 1,\ldots,n)$ помимо времени и местоположений зависят также от вектора скорости убегающего S, то есть $u_{C_j}(t,z_1(t),\ldots,z_n(t),,z(t),\mathrm{u}_S^t) \triangleq u_{C_j}$. Пусть $\mathcal{U}_{C_j}$ множество допустимых стратегий игрока C_j $(j = 1,\ldots n)$.

Игра заканчивается, когда S пойман одним из преследователей, под поимкой будем понимать совпадение местоположений S и C_j в некоторый момент времени. Тогда время поимки убегающего S преследователем C_j определяется следующим образом:

$$t_{C_j}(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = \min\{t : z_j(t) = z(t)\},$$

А момент окончания игры, то есть момент первой встречи убегающего S с одним из преследователей:

$$t_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = \min\{t_{C_1},\ldots,t_{C_n}\}.$$

Так как убегающий стремится как можно дольше оставаться непойманным, а каждый преследователь – минимизировать время поимки, функции выигрыша игроков можно определить следующим образом:

$$K_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = t_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S)$$
$$K_{C_j}(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = -t_{C_j}(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S).$$

Каждый участник стремится к достижению такой ситуации, чтобы значение его функции выигрыша было наибольшим. Таким образом рассматриваемую игру можно формализовать в виде неантоганистической игры в нормальной форме:

$$\Gamma(z_1^0,\ldots,z_n^0,z^0) = \langle N,\{\mathcal{U}_j\}_{j\in N},\{K_j\}_{j\in N}\rangle \tag{2}$$

где $N = \{C_1,\ldots,C_n,S\}$ множество игроков, а $\mathcal{U}_j$ и K_j множество стратегий j-ого игрока $(j \in N)$ и его функция выигрыша соответственно.

Существуют найденные решения данной классической постановки игры преследования n лиц, в частности известно множество ситуаций равновесия по Нэшу [2][7].

1.2. Неантагонистическая игра о встрече

Одной из поставленных задач данной работы было применение имеющейся теорию дифференциальных игр преследования к задаче, в которой все игроки преследуют цель наискорейшей встречи. Практический смысл такой задачи можно описать ситуацией, в которой один игрок обладает некой информацией или товаром, который он хотел бы распространить, а остальные игроки стремятся этот товар как можно быстрее получить.

Поэтому переформулируем описанную в предыдущем параграфе игру, руководствуясь практическим смыслом нашей задачи.

Убегающего S будем называть продавцом (Salesman), а n преследователей $C_1, C_2, \dots, C_n$ покупателями (Customer).

Начальные положения и множество допустимых управлений остаются прежними, то есть перемещения игроков можно также описать системой дифференциальных уравнений (1).

Продолжаем рассматривать случай полной информации, поэтому стратегии продавца и покупателей по-прежнему можно описать функциями

$$u_S(t, z_1(t), \dots, z_n(t), z(t)) \triangleq u_S$$
$$u_{C_j}(t, z_1(t), \dots, z_n(t), z(t), \mathrm{u}_S^t) \triangleq u_{C_j}.$$

Ключевым отличием является ход игры и момент ее окончания. Игра происходит следующим образом: в начальный момент времени продавец S объявляет покупателям $C_1, \dots, C_n$ направление своего движения, а затем, двигаясь по заданному пути, осуществляет встречи с покупателями $C_1, \dots, C_n$. Под встречей игроков понимается совпадение их местоположений. Игра заканчивается, когда S встретил всех C_j $(j = 1, \dots, n)$.

Каждый игрок C_j стремится минимизировать свое время встречи

$$t_{C_j}(z_1^0, \dots, z_n^0, z^0, u_{C_1}, \dots, u_{C_n}, u_S) = \min\{t : z_j(t) = z(t)\},$$

а игрок S – минимизировать общее время всех встреч (считаем, что на проведение самой встречи время не тратится), то есть S стремится минимизировать время окончания игры (время последней встречи)

$$t_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = \max\{t_{C_1},\ldots,t_{C_n}\}.$$

Таким образом ставится задача о встрече на быстродействие с функциями выигрыша

$$K_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = -t_S(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S)$$
$$K_{C_j}(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S) = -t_{C_j}(z_1^0,\ldots,z_n^0,z^0,u_{C_1},\ldots,u_{C_n},u_S)$$

Как и в предыдущем параграфе рассматриваемую игру можно формализовать в виде неантоганистической игры в нормальной форме:

$$\Gamma(z_1^0,\ldots,z_n^0,z^0) = \langle N,\{\mathcal{U}_j\}_{j\in N},\{K_j\}_{j\in N}\rangle \qquad (3)$$

где $N = \{C_1,\ldots,C_n,S\}$ множество игроков, а $\mathcal{U}_j$ и K_j множество стратегий j-ого игрока ($j \in N$) и его функция выигрыша соответственно.

В следующем параграфе найдем решение сформулированной игры.

1.3. Решение игры о встрече

1.3.1. Ситуация равновесия по Нэшу

Решение описанной неантагонистической игры о встрече будем искать в виде ситуации равновесия по Нэшу. Дадим определение ситуации равновесия по Нэшу.

Рассмотрим неантагонистическую игру в нормальной форме:

$$\Gamma(z_1^0,\ldots,z_n^0,z^0) = \langle N,\{\mathcal{U}_j\}_{j\in N},\{K_j\}_{j\in N}\rangle$$

Каждый игрок i ($i \in N$) выбирает некоторую стратегию u_i из своего множества допустимых стратегий, и таким образом в игре складывается ситуация $u = (u_1,\ldots,u_{i-1},u_i,u_{i+1},\ldots,u_{n+1})$. Построим ситуацию, которая отличается от u только выбором i-ого игрока. То есть стратегия u_i в ней заменена на стратегию u_i'. В таком случае получим ситуацию $(u_1,\ldots,u_{i-1},u_i',u_{i+1},\ldots,u_{n+1}) \triangleq (u||u_i')$.

Определение 1. [6] *Ситуация* $u^* = (u_1^*,\ldots,u_{i-1}^*,u_i^*,u_{i+1}^*,\ldots,u_{n+1}^*)$ *называется ситуацией равновесия по Нэшу, если для всех* $i \in N$ *и* $u_i \in U_i$ *имеет место неравенство*

$$K_i(u^*) \geq K_i(u^*||u_i')$$

Из определения следует, что не один игрок не заинтересован в отклонении от стратегии u_i^*, входящей в ситуацию равновесия, так как его выигрыш только уменьшится, в случае, если остальные игроки придерживаются стратегий, образующих ситуацию равновесия по Нэшу u^*.

1.3.2. Основные понятия и определения

Опишем понятия теории дифференциальных игр, необходимые для построения ситуации равновесия по Нэшу в описанной в параграфе 1.2 неантагонистической игре о встрече (3).

Так как перед нами стоит задача на быстродействие в условиях полной информации, нам понадобится определение стратегии параллельного сближения, которое было введено Л.А.Петросяном в 1965 году [3].

Определение 2. [3][4] *Параллельным сближением (П-стратегией) называется способ преследования точкой C_j точки S, при котором управление игрока C_j в каждый момент времени совпадает с управлением, гарантирующим ему быстродействие в точку встречи .*

При использовании игроком C_j П-стратегии отрезок $C_j^t S^t$ в каждый момент времени до момента встречи параллелен отрезку $C_j^0 S^0$, то есть перемещается параллельно самому себе. И его длина строго убывает . (Здесь $C_j^t = z_j(t)$, а $S^t = z(t)$.)

Определение 3. [4] *Окружность Аполлония $A(z_j^0, z^0)$ преследователя C_j и убегающего S это множетсво точек A, для которых выполняется условие*

$$\frac{|C_j^0 A|}{\alpha_j} = \frac{|S^0 A|}{\beta}. \tag{4}$$

Множество точек, ограниченных окружностью Аполлония $A(z_j^0, z^0)$, будем называть кругом Аполлония и обозначать A_j (См. Рис. 1).

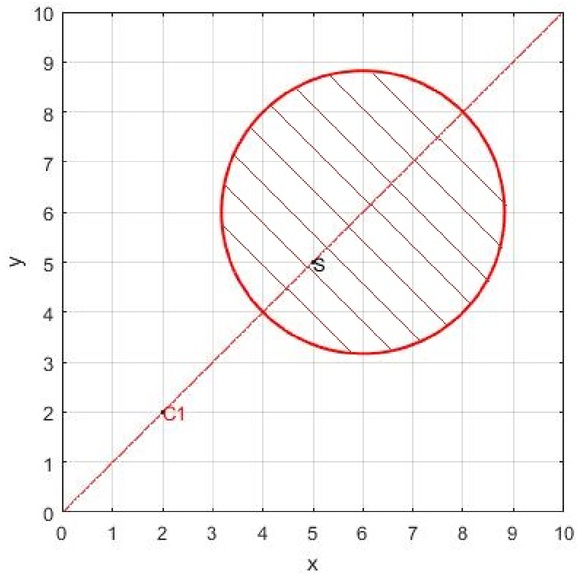

Рис. 1: Окружность Аполлония $A(z_1^0, z^0)$. Заштрихованная область круг Аполлония A_1

В случае использования стратегии параллельного сближения игроком C_j встречи происходят внутри круга Аполлония, если в дополнении к этому игрок S движется прямолинейно, встреча с игроком C_j произойдет на окружности Аполлония [4].

Введем также обозначение границы области, образованной объединением всех кругов Апполония $\mathcal{D}A = \mathcal{D}(A_1 \cup \ldots \cup A_n)$ (См. Рис. 2).

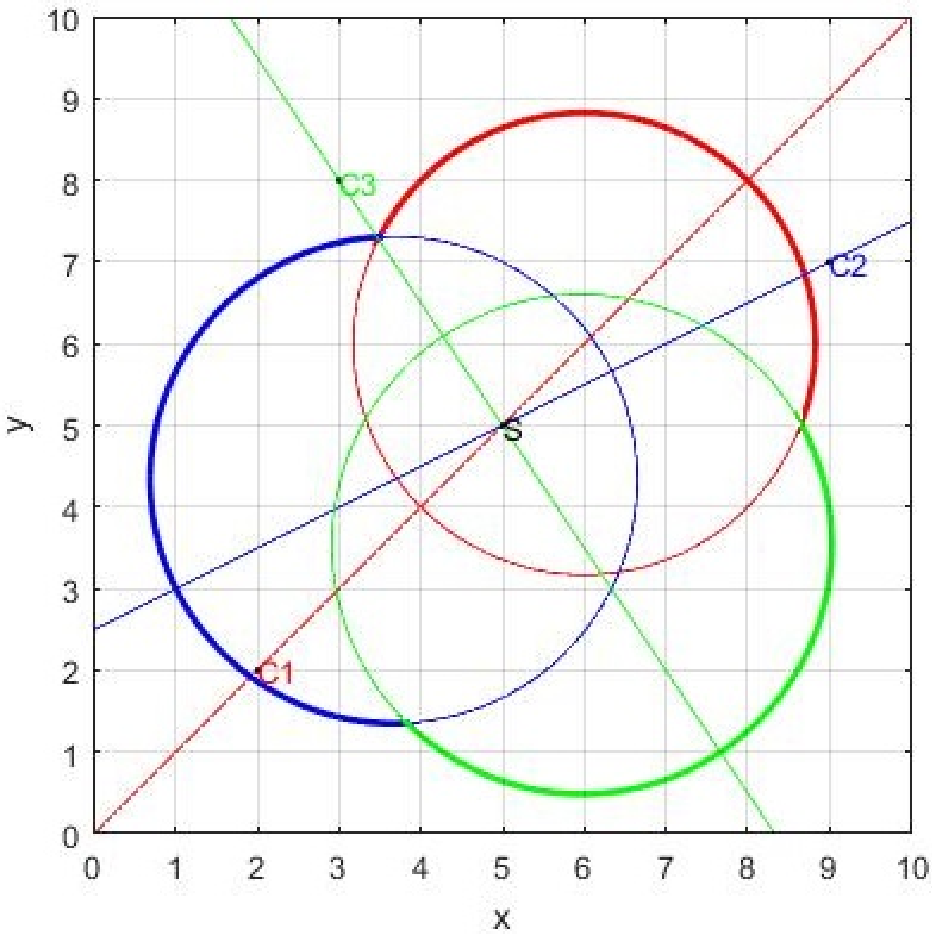

Рис. 2: Граница $\mathcal{D}A$ выделена жирным

1.3.3. Поведение игрока C_j

Так как перед нами ставится задача на быстродействие в условиях полной информации, игроки C_j $(j = 1, \ldots, n)$ максимизируют свои выигрыши, используя стратегию параллельного сближения, которая по определению гарантирует им быстродействие в точку встречи.

1.3.4. Поведение игрока S

Таким образом, построив окружности Аполлония для всех игроков, можно определить поведение всех покупателей C_j $(j = 1, \ldots, n)$, которое гарантирует минимальное время встречи при указанном (выбранном) S направлением движения, отклонятся от которого каждому покупателю по отдельности не выгодно. И вопрос поиска ситуации равновесия по Нэшу сводится к поиску поведения игрока S, максимизирующему его выигрыш. Опишем несколько типов поведения игрока S:

1. Игрок S использует тип поведения u_S^1, согласно которому движется прямолинейно по направлению к игроку C_j, то есть в ближайшую точку окружности Аполлония A_j $(j = 1, \ldots, n)$, которая принадлежит границе $\mathcal{D}A$ (См. Рис. 3).

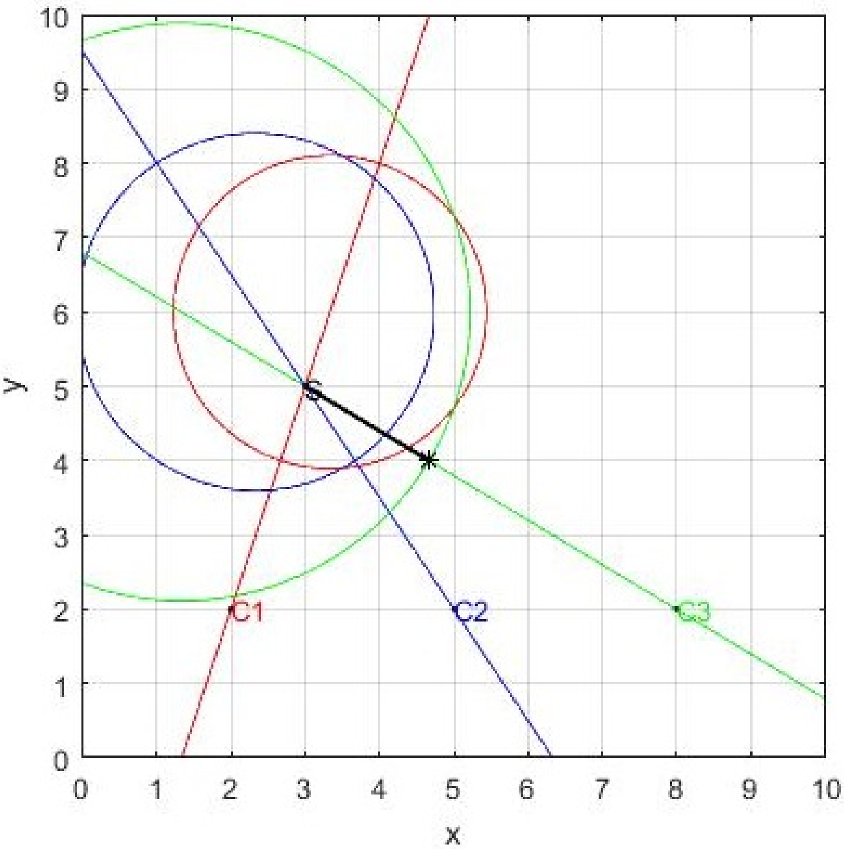

Рис. 3: Движение согласно типу поведения u_S^1

2. Игрок S использует тип поведения u_S^2, в соответствие с которым движется по прямой в ближайшую точку пересечения окружностей Аполлония $A(z_j^0,z^0)$ и $A(z_k^0,z^0)$ $(j \neq k)$, которая принадлежит границе $\mathcal{D}A$ (См. Рис. 4).

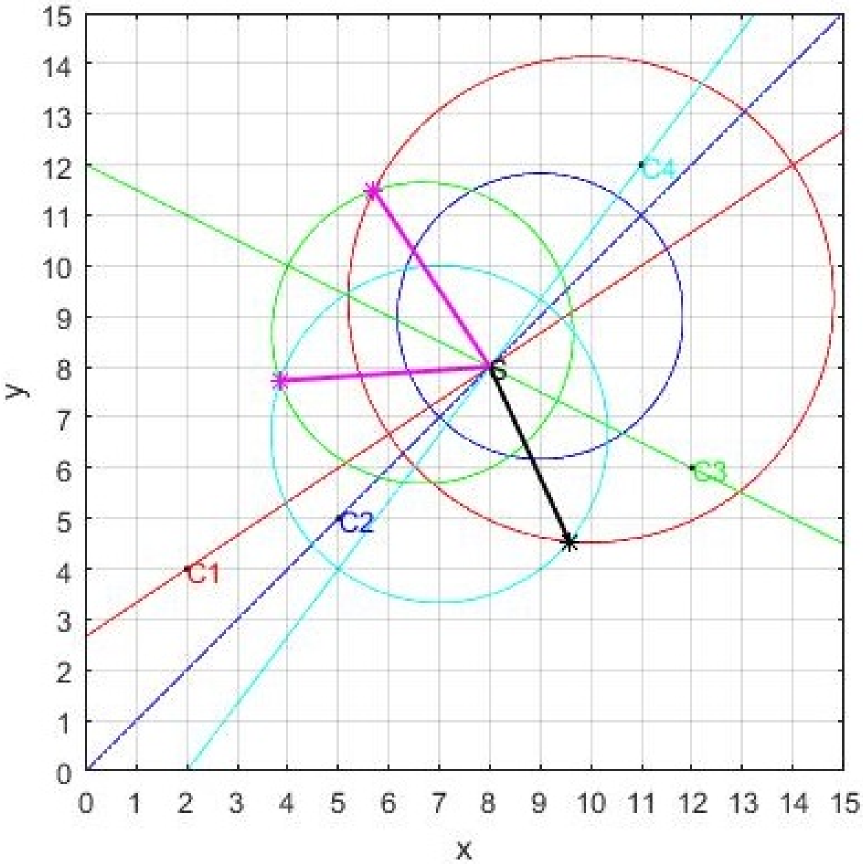

Рис. 4: Движение согласно типу поведения u_S^2

3. Игрок S использует тип поведения u_S^3, в соответствие с которым движется по кратчайшей ломаной: в начале в точку пересечения окружностей Аполлония $A(z_j^0,z^0)$ и $A(z_k^0,z^0)$ $(j \neq k)$, которая окажется на границе после исключения текущей границы $\mathcal{D}A$, а затем по направлению к последнему оставшемуся игроку C_i (См. Рис. 5).

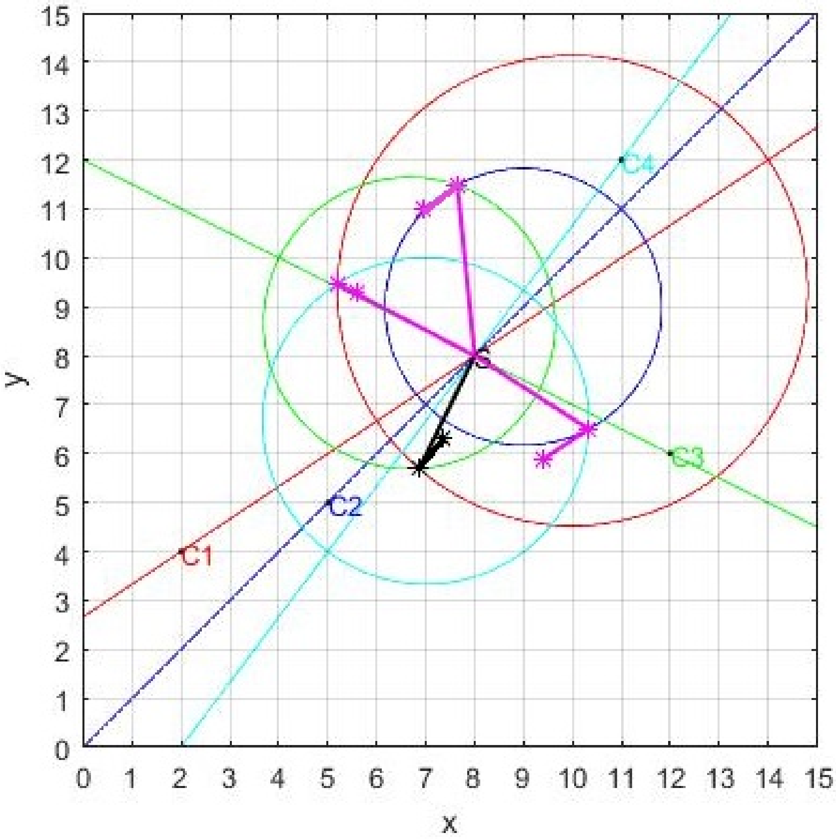

Рис. 5: Движение согласно типу поведения u_S^3

Теорема 1 (Я.Б.Панкратова, С.И.Тарашнина 2016).
В игре $\Gamma(z_1^0, \ldots, z_n^0, z^0) = \langle N, \{\mathcal{U}_j\}_{j \in N}, \{K_j\}_{j \in N} \rangle$ *существует ситуация равновесия по Нэшу, которая строится следующим образом:*

1. *Игрок S выбирает стратегию u_S^*, которая предписывает ему один из трех типов поведения и дает минимальное время встречи t_S.*

2. *Игроки C_j $(j = 1, \ldots, n)$ двигаются согласно стратегии параллельного сближения $u_{C_j}^* = u_{C_j}^{\Pi}$.*

Замечание. *Если существует тип поведения u_S^1, то его использование будет давать минимальное время, и нет смысла рассматривать типы поведений u_S^2 и u_S^3.*

Глава 2. Программная реализация

2.1. Описание программной реализации

Программная реализация была выполнена в среде MATLAB. Эта среда была выбранна в силу обилия удобных графических средств, необходимых для графической визуализации начальных условий поставленной задачи и найденных решений (построенных стратегий). В результате работы программы строится ситуация равновесия по Нэшу в заданной неантагонистической игре о встрече (то есть определяются стратегии игроков, которые создают эту ситуацию), а также вычисляются выигрыши игроков в построенной ситуации. Кроме того строится график со всеми окружностями Апполония в начальный момент времени, на котором найденное решение изображено в виде траектории игрока S.

Программа написана (Приложение 1), основываясь на Теореме 1. То есть в начале программа рассчитывает, возможно ли из данных начальных положений движение согласно типу поведения u_S^1, если да, то ситуация равновесия найдена. Для проверки этого условия находятся ближайшие точки встречи с каждым из покупателей C_j $(j = 1, \ldots, n)$ (то есть ближайшие точки каждой окружности Аполлония $A(z_j^0, z^0)$ $(j = 1, \ldots, n)$), а затем реализуется движение игрока S в самую дальнюю из найденных точек. При движение по выбранному пути записывается время осуществления встреч с покупателями. Если было проведено n встреч (то есть были встречены все покупатели), то стратегия, задающая это движение, входит в ситуацию равновесия, а запомненное время встречи с каждым C_j с отрицательным знаком является выигрышем C_j. Выигрыш S – время последней встречи с обратным знаком (Приложение 2).

Если же по пути, заданному движением в дальнюю из ближайших точек окружностей, не удалось встретить всех покупателей, то осуществляется расчет времени встреч при поведение u_S^2 и u_S^3. Нахождение стратегий, согласованных с данными типами поведения, реализуется также за счет проверки количества встреч при движение в некоторую точку пересечения окружностей Аполлония $A(z_j^0,z^0)$ и $A(z_k^0,z^0)$ $(j \neq k)$.

Если было проведено n встреч, значит движение в эту точку соответствует поведению u_S^2 и время последней встречи записывается (запоминается) для дальнейшего сравнения.

Если же по пути было встречено $n-1$ покупателей, то пересчитываются местоположения игроков в момент последней (то есть $(n-1)$-ой) встречи и рассчитывается дальнейшее прямолинейное движение по направлению к последнему оставшемуся покупателю. Такая стратегия соответствует поведению u_S^3 и время последней встречи также запоминается для дальнейшего сравнения.

Иначе, то есть встреч было меньше $n-1$, движение в такую точку пересечения прерывается и в дальнейшем не рассматривается (Приложение 3).

После проверки всех точек пересечения находится минимальное время последней встречи (из запомненного), и стратегия игока S, соответствующая движению, дающему это минимальное время, как раз входит в искомую ситуацию равновесия. Выигрыши игроков получаются из времени встреч, также расчитанном на этапе проверки движения в точки пересечения.

Также для наглядной визуализации строится график, на котором изображены: начальные местоположения игроков, окружности Аполлония $A(z_j^0,z^0)$ $(j = 1,\ldots,n)$, прямые $S^0C_j^0$ и траектория S, движение по которой соответствует его стратегии, которая дает минимальное время последней встречи и входит в ситуацию равновесия по Нэшу. В случае использования игроком третьего типа поведения также строится окружность Апполония последнего покупателя в момент $(n-1)$-ой встречи и изображается его местоположение в этот момент.

2.2. Численные примеры программной реализации

2.2.1. Иллюстрация первого типа поведения игрока S

Пример 1. Рассмотрим игру 5 лиц (1 продавец S и 4 покупателя C_j) с начальными местоположениями игроков $S^0 = (8{,}8)$; $C_1^0 = (7{,}1)$; $C_2^0 = (3{,}5)$; $C_3^0 = (9{,}2)$; $C_4^0 = (12{,}4)$ и скоростями $\beta = 2$; $\alpha_1 = 3.5$; $\alpha_2 = 4$; $\alpha_3 = 4$; $\alpha_4 = 4$ соответственно.

На Рис. 6 отображено найденное в результате работы пргограммы решение.

```
S встретил C3 в точке 7.706431 5.945020 через 1.037922 единиц времени
S встретил C4 в точке 7.671192 5.698343 через 1.162512 единиц времени
S встретил C2 в точке 7.666667 5.666667 через 1.178511 единиц времени
S встретил C1 в точке 7.636364 5.454545 через 1.285649 единиц времени
Произошли все встречи в порядке 3421 за время 1.285649

Равновесная стратегия S - движение согласно 1 типу поведения по направлению к C1;
соответсвующий порядок встреч 3421 за время 1.285649
```

Рис. 6: Результат работы программы на Примере 1.

Таким образом, в данной игре существует ситуация равновесия по Нэшу, в которой игрок S согласно теореме использует тип поведения u_S^1 и будет двигаться по направлению к C_1 в точку $(7.636, 5.454)$, траектория его движения на Рис. 7 показана черным отрезком. А покупатели C_j будут двигаться согласно стратегии параллельного сближения: C_1 в точку $(7.636, 5.454)$, C_2 в точку $(7.666, 5.666)$, C_3 в точку $(7.706, 5.945)$ и C_4 в $(7.671, 5.698)$.

Встречи состоятся в порядке C_3, C_4, C_2, C_1.

А выигрыши игроков в сложившейся ситуации отражены в Таблице 1.

Таблица 1: Выигрыши игроков в Примере 1.

K_{C_1}	-1.285649	K_{C_2}	-1.178511
K_{C_3}	-1.037922	K_{C_4}	-1.162512
K_S		-1.285649	

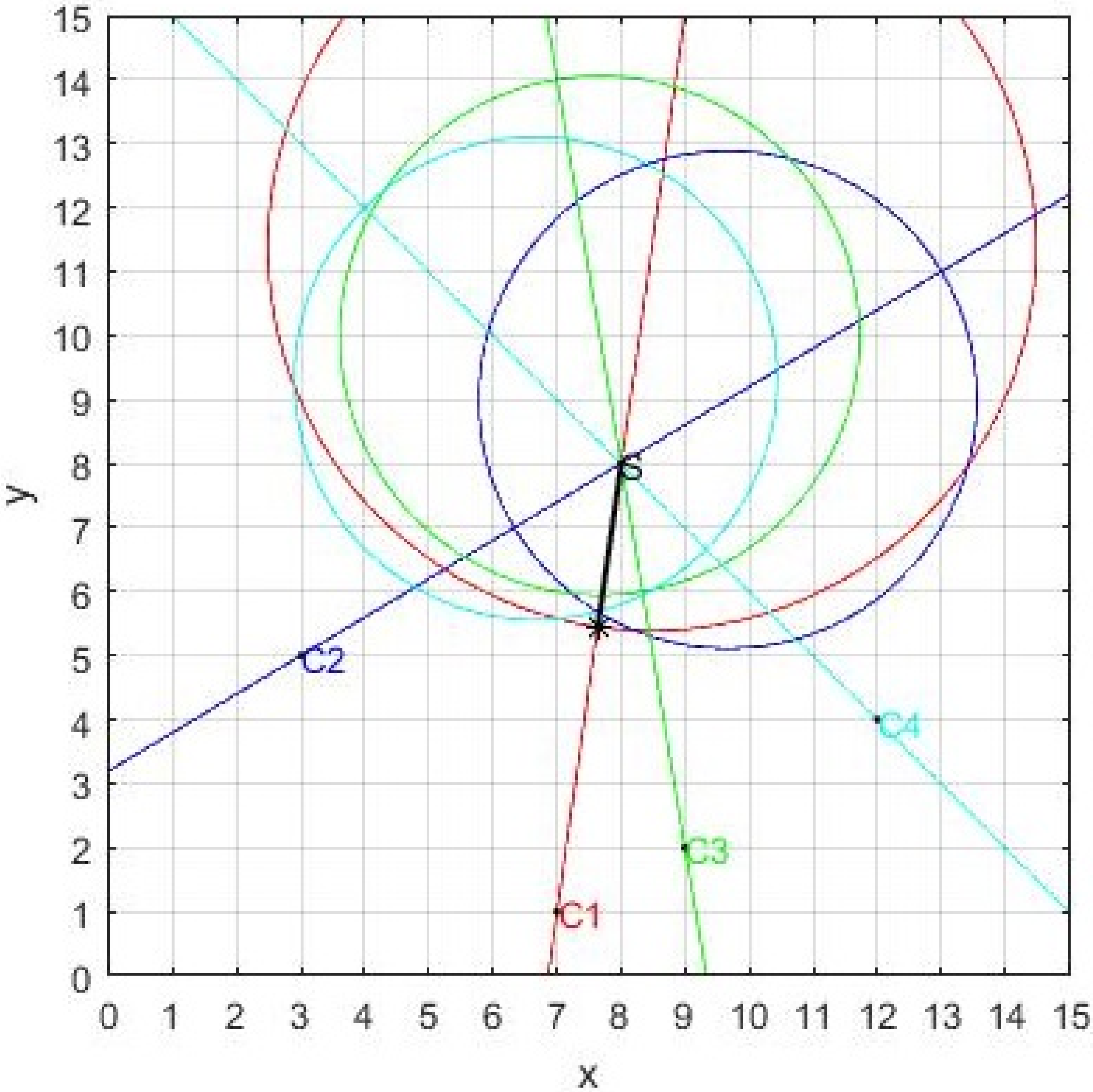

Рис. 7: Иллюстрация типа поведения u_S^1

2.2.2. Иллюстрация второго типа поведения игрока S

Пример 2. Рассмотрим игру 5 лиц (1 продавец S и 4 покупателя C_j) с начальными местоположениями игроков $S^0 = (7{,}8)$; $C_1^0 = (2{,}11)$; $C_2^0 = (3{,}6)$; $C_3^0 = (4{,}14)$; $C_4^0 = (5{,}3)$ и скоростями $\beta = 2$; $\alpha_1 = 4$; $\alpha_2 = 4$; $\alpha_3 = 4$; $\alpha_4 = 4$ соответственно.

В данной игре существует ситуация равновесия по Нэшу, в которой игрок S согласно теореме использует тип поведения u_S^2 и будет двигаться прямолинейно в точку пересечения окружностей $A(z_3^0, z^0)$ и $A(z_4^0, z^0)$

с координатами $(4.278, 8.479)$, траектория его движения на Рис. 8 показана черным отрезком. А покупатели C_j будут двигаться согласно стратегии параллельного сближения: C_1 в точку $(5.02, 8.348)$, C_2 в точку $(5.376, 8.286)$, C_3 и C_4 в точку $(4.278, 8.479)$.

В этом случае встречи состоятся в порядке C_2, C_1, C_3 и C_4.

Выигрыши игроков в такой ситуации составят:

Таблица 2: Выигрыши игроков в Примере 2.

K_{C_1}	-1.004803	K_{C_2}	-0.824385
K_{C_3}	-1.381790	K_{C_4}	-1.381790
K_S		-1.381790	

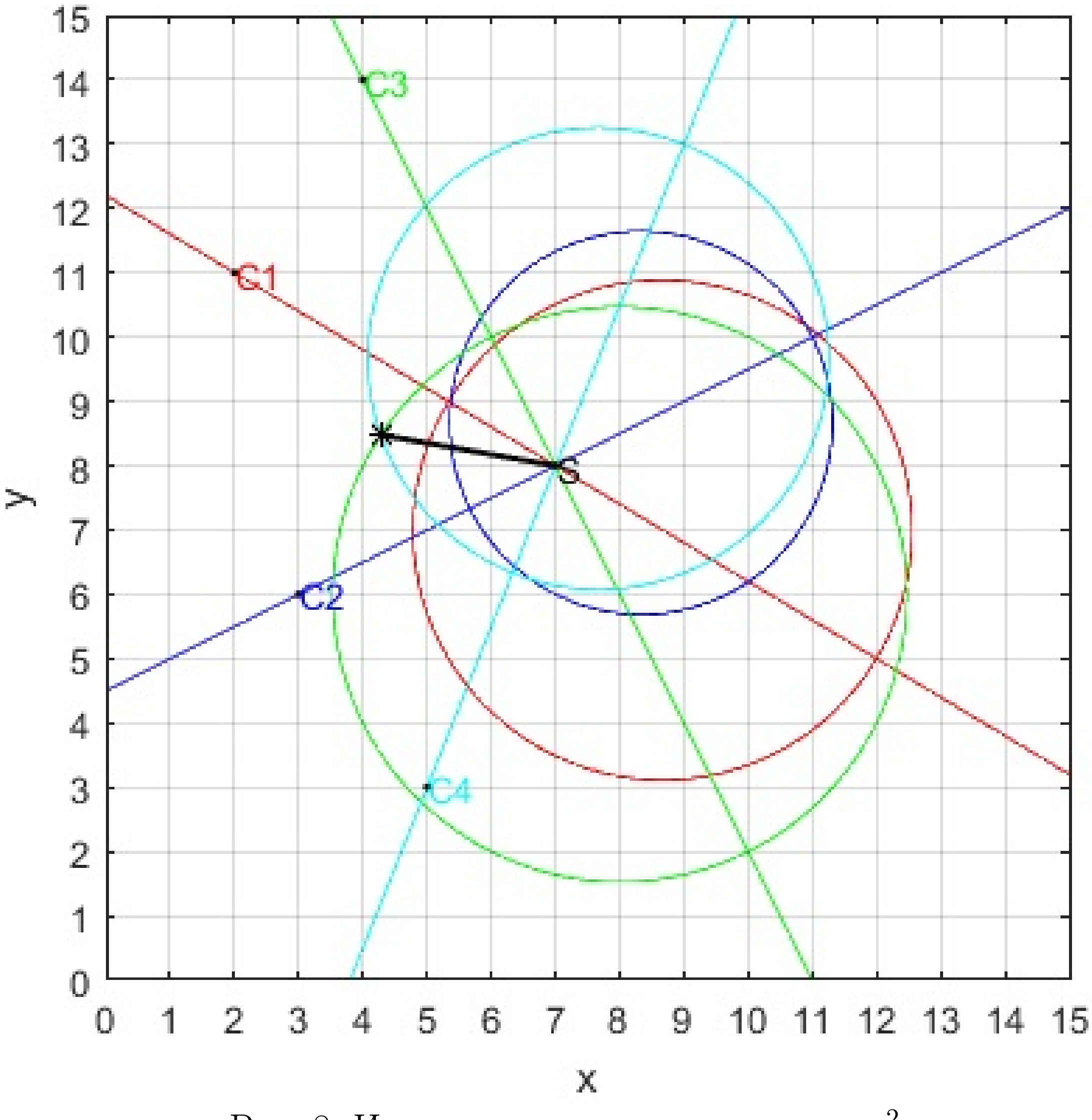

Рис. 8: Иллюстрация типа поведения u_S^2

На Рис.9 показано, как данный результат выводится программно.

```
S идет в точку пересечения окружностей Аполлония(34)
S встретил C2 в точке 5.376274 8.286284 через 0.824385 единиц времени
S встретил C1 в точке 5.020919 8.348938 через 1.004803 единиц времени
S встретил C3 и C4 в точке 4.278398 8.479854 через 1.381790 единиц времени
Произошли все встречи в порядке 2134 за время 1.381790

Равновесная стратегия S - движение согласно 2 типу поведения в точку пересечения окружностей (34);
соответсвующий порядок встреч 2134 за время 1.381790
```

Рис. 9: Результат работы программы на Примере 2.

2.2.3. Иллюстрация третьего типа поведения игрока S

Пример 3. Рассмотрим игру 5 лиц (1 продавец S и 4 покупателя C_j) с начальными местоположениями игроков $S^0 = (7{,}8)$; $C_1^0 = (2{,}11)$; $C_2^0 = (13{,}6)$; $C_3^0 = (4{,}14)$; $C_4^0 = (5{,}3)$ и скоростями $\beta = 2$; $\alpha_1 = 4$; $\alpha_2 = 4$; $\alpha_3 = 4$; $\alpha_4 = 4$ соответственно.

В данной игре существует ситуация равновесия по Нэшу, в которой игрок S согласно теореме использует тип поведения u_S^3 и будет двигаться в начале в точку пересечения окружностей $A(z_3^0, z^0)$ и $A(z_4^0, z^0)$ с координатами $(4.278, 8.479)$, а затем по направлению к C_2 в точку $(5.399, 8.106)$, траектория его движения на Рис. 10 изображена черной ломаной. На рисунке также, помимо окружностей Апполония в начальный момент времени, изображена окружность Апполония последнего покупателя в момент смены игроком S направления движения, а также местоположение этого покупателя в этот момент отмечено точкой NC_2. Покупатели же в сложившейся ситуации равновесия будут двигаться согласно стратегии параллельного сближения: C_1 в точку $(4.830, 8.382)$, C_2 в точку $(5.399, 8.106)$, C_3 и C_4 в точку $(4.278, 8.479)$.

Выигрыши игроков в этой ситуации отражены в Таблице 3.

Таблица 3: Выигрыши игроков в Примере 2.

K_{C_1}	-1.004803	K_{C_2}	-1.972784
K_{C_3}	-1.381790	K_{C_4}	-1.381790
K_S		-1.972784	

Встречи будут осуществлены в порядке C_1, C_3 и C_4,C_2.

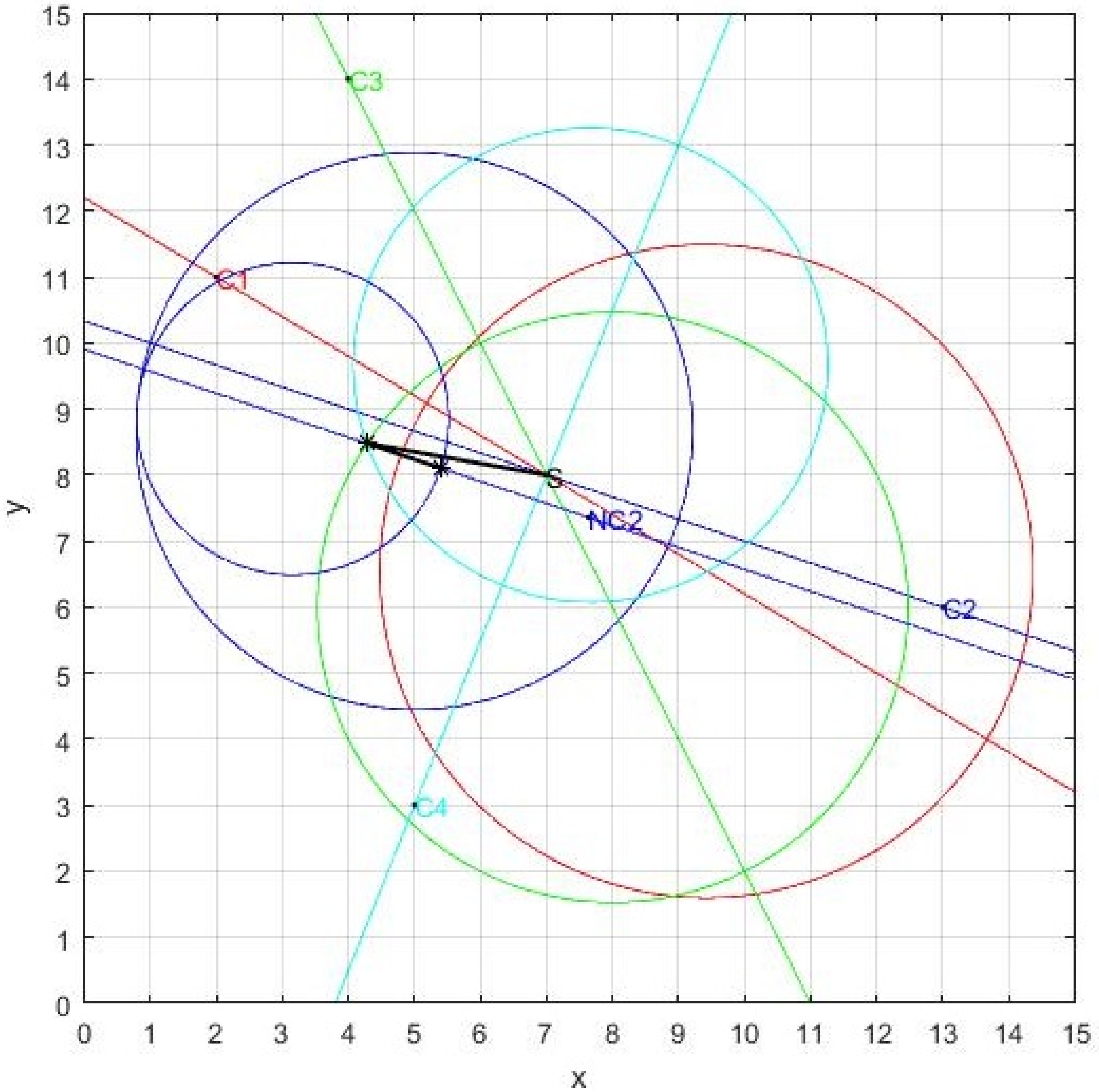

Рис. 10: Иллюстрация типа поведения u_S^3

Программно результаты выводятся следующим образом:

```
S идет в точку пересечения окружностей Аполлония(34)
S встретил C1 в точке 4.830474 8.382516 через 1.101495 единиц времени
S встретил C3 и C4 в точке 4.278398 8.479854 через 1.381790 единиц времени
S встретил C2 в точке 5.399730 8.106077 через 1.972784 единиц времени
Произошли все встречи в порядке 1342 за время 1.972784

Равновесная стратегия S - движение согласно 3 типу поведения в точку пересечения окружностей (34);
соответсвующий порядок встреч 1342 за время 1.972784
```

Рис. 11: Результат работы программы на Примере 3.

Пример 4. Интересным для рассмотрения является случай симметричного расположения покупателей относительно S. В такой игре существует несколько ситуаций равновесия по Нэшу.

Рассмотрим игру 5 лиц (1 продавец S и 4 покупателя C_j) с начальными местоположениями игроков $S^0 = (7{,}8)$; $C_1^0 = (11{,}12)$; $C_2^0 = (3{,}4)$; $C_3^0 = (3{,}12)$; $C_4^0 = (11{,}4)$ и скоростями $\beta = 2$; $\alpha_1 = 3.5$; $\alpha_2 = 4$; $\alpha_3 = 3.5$; $\alpha_4 = 4$ соответственно.

Так как интерес представляет не просто симметричное начальное местоположение игроков, а симметрия окружностей Аполлония, то, учитывая начальные условия и скорости игроков, симметричными можно назвать пары покупателей $C_1 - C_3$ и $C_2 - C_4$.

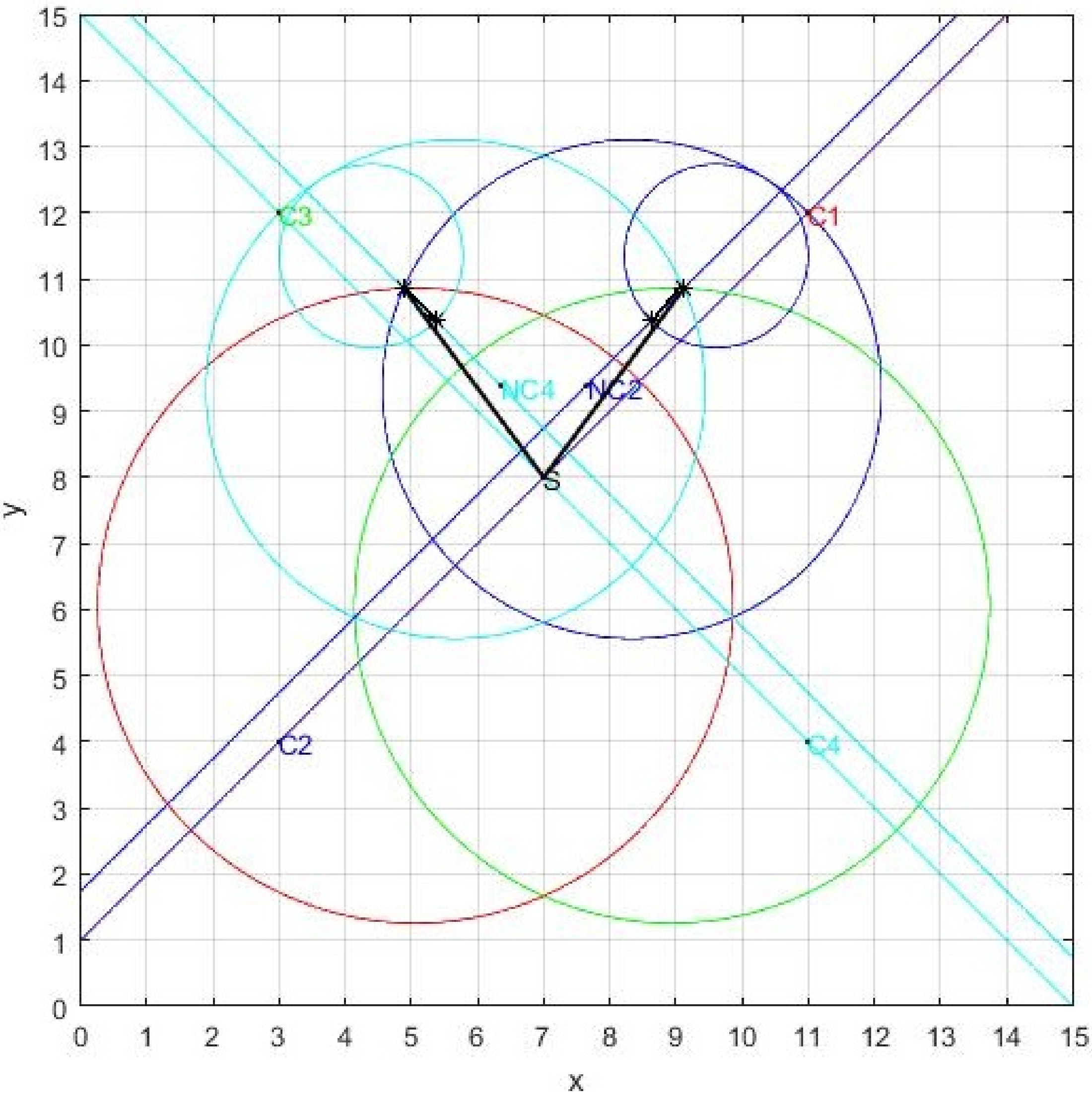

Рис. 12: Иллюстрация типа поведения u_S^3 в симметричной ситуации

В данной игре существует две ситуации равновесия по Нэшу, обе задаются поведением u_S^3 игрока S:

1. В начале S движется прямолинейно в точку пересечения окружностей $A(z_3^0,z^0)$ и $A(z_4^0,z^0)$ с координатами $(9.116, 10.857)$, а затем по направлению к C_2 в точку $(8.625, 10.366)$. Встречи состоятся в порядке C_1, C_3 и C_4,C_2.

2. В начале S движется прямолинейно в точку пересечения окружностей $A(z_1^0,z^0)$ и $A(z_2^0,z^0)$ с координатами $(4.883, 10.857)$, а затем по направлению к C_4 в точку $(5.374, 10.366)$. Встречи состоятся в порядке C_3, C_1 и C_2,C_4.

Траектории игрока S при использование описанных стратегий изображены на Рис. 12 черными ломаными. На рисунке также, помимо окружностей Апполония в начальный момент времени, изображены: окружность Апполония последнего покупателя в момент смены игроком S направления движения (синяя окружность меньшего радиуса при использовании первой описанной стратегии; голубая – второй) и местоположение этого покупателя в этот момент (точка NC_2 для первой стратегии;NC_4 для второй).

Программно полученные решения описываются следующим образом:

```
S идет в дальнюю точку пересечения окружностей Аполлония(12)
S встретил C3 в точке 5.767928 9.663307 через 1.034963 единиц времени
S встретил C1 и C2 в точке 4.883643 10.857098 через 1.777778 единиц времени
S встретил C4 в точке 5.374333 10.366407 через 2.124748 единиц времени
Произошли все встречи в порядке 3124 за время 2.124748

S идет в точку пересечения окружностей Аполлония(34)
S встретил C1 в точке 8.232072 9.663307 через 1.034963 единиц времени
S встретил C3 и C4 в точке 9.116357 10.857098 через 1.777778 единиц времени
S встретил C2 в точке 8.625667 10.366407 через 2.124748 единиц времени
Произошли все встречи в порядке 1342 за время 2.124748

Равновесная стратегия S - движение согласно 3 типу поведения в точку пересечения окружностей (34);
соответсвующий порядок встреч 1342 за время 2.124748

Равновесная стратегия S - движение согласно 3 типу поведения в дальнюю точку пересечения окружностей (12);
соответсвующий порядок встреч 3124 за время 2.124748
```

Рис. 13: Результат работы программы на Примере 4.

Можно заметить (См. Таблицу 4), что выигрыши игроков, обозначенных ранее симметричными, попарно меняются в зависимости от выбора стратегии игроком S, при этом выигрыш самого S сохраняется независимо от его выбора.

Таблица 4: Выигрыши игроков в симметричной ситуации

	Ситуация равновесия 1.	Ситуация равновесия 2.
K_{C_1}	-1.034963	-1.777778
K_{C_2}	-2.124748	-1.777778
K_{C_3}	-1.777778	-1.034963
K_{C_4}	-1.777778	-2.124748
K_S	-2.124748	

Замечание. *Таким образом, если в классической постановке игрок S был дискриминирован и, хотя выбор стратегий преследователей напрямую зависел от выбора стратегии игрока S, считался слабым, то в данной постановке игрок S является самым влиятельным, так как по сути выбирает выигрыши покупателей.*

2.3. Возможные интересные задачи для дальнейшего исследования

Интересно в данном случае не только наличие нескольких ситуаций равновесия, но и тот факт, что в условиях другой постановки задачи (а именно ограничение на максимальный модуль скорости, а не на его постоянство) один из симметричных игроков C_j, замедлившись, лишил бы игрока S выбора между несколькими ситуациями равновесия с равным для S выигрышем и вынуждал бы его двигаться в ту сторону, где встреча с C_j происходит раньше. Таким образом, введение продавца S покупателем C_j в заблуждение о его максимальной по модулю скорости и дальнейшее использование заявленной скорости (для сокрытия обмана и сохранения торговых отношений) выгодно игроку C_j в случае, если:

$$\frac{1}{2}K_{C_j}(z_1^0,\ldots,u_{C_j}^{max},\ldots,u'_S)+\frac{1}{2}K_{C_j}(z_1^0,\ldots,u_{C_j}^{max},\ldots,u''_S)<K_{C_j}(z_1^0,\ldots,u_{C_j}^{slow},\ldots,u'_S)$$

где u'_S стратегия игрока S при которой он движется в сторону, где встреча с C_j происходит раньше, а u''_S стратегия, при которой движение происходит в обратную сторону. $u_{C_j}^{max}$ здесь означает стратегию, при которой C_j не обманывает игрока S о значение своей максимальной скорости, а $u_{C_j}^{slow}$ описывает движение C_j с намеренно заявленной чуть меньшей максимально возможной скоростью.

Таким образом, случай симметрии в условиях ограничения на максимальную по модулю скоростью является крайне интересным, но при его изучении появляется ряд проблем, как например поведения игрока симметричного C_j. В условиях же нашей постановки симметричные случаи не менее интересны, так как возможны не только две ситуации равновесия, но и большее их число в зависимости от количества игроков, их начальных местоположений и скоростей.

Заключение

В ходе данной работы была изучена теория дифференциальных игр преследования, формализована задача о встрече, заданы типы поведения игроков, создающие ситуацию равновесия по Нэшу, сформулированные в Теореме 1.

Также была выполнена программная реализация для нахождения ситуаций равновесия по Нэшу и выигрышей игроков в этих ситуациях. Были рассмотрены различные численные примеры, в том числе получен интересный результат для случая симметричных начальных местоположений.

Список литературы

[1] Айзекс Р. Дифференциальные игры.- Москва: Издательство "Мир 1967. - 480 С.

[2] Панкратова Я.Б. Решение кооперативной дифференциальной игры группового преследования // Дискретный анализ и исследования операций. 2010. том. 17, номер 2. с.57-78.

[3] Петросян Л.А. Об одном семействе дифференциальных игр на выживание в пространстве R^n // Доклады академии наук СССР, 1965. — Т. 161, — № 1. с.52–54.

[4] Петросян Л.А., Томский Г.В. Геометрия простого преследования. — Новосибирск: Издательство:Наука, 1983. — 143 С.

[5] Петросян Л.А., Ширяев В.Д. Простое преследование одним преследователем двух преследуемых.-В кн.:Некоторые вопросы дифференциальных и интегральных уравнений и их приложения. Вып.3, Якутск. ун-т. 1978. с.103-108.

[6] Nash J.F. Equilibrium points in n-person games// Proc. Nat.Acad. Sci. USA. 1950. Vol.36. p.48-49.

[7] Tarashnina S. Nash equilibria in a differential pursuit game with one pursuer and m evaders. Game Theory and Applications. N.Y. Nova Science Publ. 1998. Vol. III, p.115-123.

Приложение

Приложение 1. Программная реализация нахождения ситуации равновесия

```
S=[7 8];       %начальное местоположение и скорость продавца
vs=2;
A=[11 3 3 11;  %начальное местоположение и скорости покупателей
   12 4 12 4];
vA=[ 3.5 4 3.5 4];
forprint=['%d','%d','%d','%d','%d','%d','%d','%d','%d','%d'];
%для корректного вывода при любом n
n=length(vA);
[TAmaxmin,meetordermaxmin,stratmaxmin] = StoAmaxmin(S,vs,A,vA,n);
%реализация первого типа поведения
   if (TAmaxmin)==inf
      [Tsumsoch,meetordersoch,stratsoch ] =
      StoAperesechenie(S,vs,A,vA,n);
      %реализация второго и третьего типа поведения при движение
      %в точку пересечения окружностей ближайшую из двух
      [Tsumsochmax,meetordersochmax,stratsochmax ] =
      StoAperesecheniemax(S,vs,A,vA,n);
      %реализация второго и третьего типа поведения при движение
      %в точку пересечения окружностей дальнюю из двух
      Topt=min([Tsumsoch,Tsumsochmax]);   %поиск максимального
      выигрыша S среди стратегий, соответствующих второму и третьему
      типу поведения
         if (Tsumsoch==Tsumsochmax)    %поиск стратегии,
         соответствующей этому выигышу
            bestorder=[meetordersoch;meetordersochmax];
            beststrat=cell(1,size(bestorder,1));
```

```
        for i=1:1:size(stratsoch,1)
        beststrat{i}=['в точку пересечения (',
        num2str(stratsoch(i,1)*10+stratsoch(i,2)),')'];
        end
        for i=1:1:size(stratsochmax,1)
        beststrat{i+size(stratsoch,1)}=['в дальнюю точку
        пересечения (',num2str(stratsochmax(i,1)*10+
        stratsochmax(i,2)),')'];
        end
     elseif (Tsumsoch==Topt)
        bestorder=meetordersoch;
        beststrat=cell(1,size(bestorder,1));
        for i=1:1:size(stratsoch,1)
        beststrat{i}=['в точку пересечения (',
        num2str(stratsoch(i,1)*10+stratsoch(i,2)),')'];
        end
     else
        bestorder=meetordersochmax;
        beststrat=cell(1,size(bestorder,1));
        for i=1:1:size(stratsochmax,1)
        beststrat{i}=['в дальнюю точку
        пересечения (',num2str(stratsochmax(i,1)*10+
        stratsochmax(i,2)),')'];
        end
     end
else %возможен первый тип поведения
   bestorder=meetordermaxmin;
   beststrat=['по направлению к ',num2str(stratmaxmin)];
 Topt=TAmaxmin;
end
```

```
for i=1:1:size(bestorder,1)
fprintf(['Равновесная стратегия S - движение ',beststrat{i},'
соответсвующий порядок встреч ',forprint(1:2*n),'
за время %f\n\n'],bestorder(i,:),Topt)
end
forngrafic(S,vs,A,vA) %визуализация начальных данных
```

Приложение 2. Программная реализация первого типа поведения игрока S

```
function [T,meetorder,imaxminA] = StoAmaxmin(S,vs,A,vA,n)
%движение в самую дальнюю из ближайших точек встреч каждого A и
%S(пересечения прямой SAi и i-ой окр)
NaprA=zeros(2,n);
TimeA=zeros(1,n);
SNAi=zeros(1,n);
RA=vpa(zeros(1,n));
SA=vpa(zeros(1,n));
forprint=['%d','%d','%d','%d','%d','%d','%d','%d','%d','%d'];
for i=1:1:n
   RA(i)=krug(S,A(:,i),vs,vA(i));   %строим окружность Апполония
   i-ого покупателя
   SA(i)=mline(S,A(:,i));
   [NaprA(:,i)]=vstrecha(RA(i),SA(i),S,A(:,i));   %находим точки
   пересечения окружности и прямой, выбираем ближайшую к покупателю
   SNAi(i)=mlength(S,NaprA(:,i));
   TimeA(i)=SNAi(i)/vs;
end
T=max(TimeA);
imaxminA=find(TimeA==T);   %покупатель, точка встречи с которым
может находиться на границе
```

```
for i=1:1:n    %поиск точек встреч и времени встреч с каждым
покупателем при движение к выбранному покупателю
   [NaprA(:,i)]=vstrecha(RA(i),SA(imaxminA),S,A(:,imaxminA));
   SNAi(i)=mlength(S,NaprA(:,i));
   TimeA(i)=SNAi(i)/vs;
end
meetorder=zeros(1,n);
i=1;
while (~isempty(find(TimeA==T))) %осуществление встреч пока
не произошла встреча с выбранным покупателем
   imeetA=find(TimeA==min(TimeA));
   fprintf('встретил C%d в точке %f %f через %f единиц времени\n',
   imeetA,NaprA(1,imeetA),NaprA(2,imeetA),min(TimeA))
   TimeA(imeetA)=[inf];
   meetorder(i)=imeetA;
   i=i+1;
end
   if i==n+1
      fprintf(['Встретил всех в порядке ',forprint(1:2*n),'
      за время %f\n\n'],meetorder,T )
   else
      fprintf('Не удалось встретить всех до встречи с последним,
      данная стратегия не является равновесной\n\n')
      T= inf;
   end
end
```

Приложение 3. Программная реализация второго и третьего типа поведения игрока S

```
function [Tsumsoch,meetorder,stratsoch] =
StoAperesechenie(S,vs,A,vA,n)
%движение в точку пересечения окружностей ближайшую из двух
sochetania =nchoosek([1:n],2);
Tsumsoch=zeros(1,size(sochetania,1));
meetorder=zeros(n,size(sochetania,1));
forprint=['%d','%d','%d','%d','%d','%d','%d','%d','%d','%d'];
for i=1:1:size(sochetania,1)
    VstA=A;
    VstS=S;
    meetA=zeros(1,n);
    %запишем сюда порядок встреч для i-ого варианта сочетаний
    NaprA=zeros(2,n);
    Tsum=0;
    order=[1:n];
    soch=sochetania(i,:);
    fprintf('идет в точку пересечения окружностей
    Аполлония(%d%d)\n',soch)
    [SAA,Achoice] = StodoubleA(VstS,vs,VstA,vA,soch );
    %строим прямую, проходящую через S и ближайшую точку пересечения
    %выбранных (soch) окружностей Аполлония
       if SAA==0 %выбранные окружности не пересекаются
          Tsumsoch(i)=inf;
       else
          TimeA=zeros(1,n);
          SNAi=zeros(1,n);
          for k=1:1:n
          RA = krug(S,VstA(:,k),vs,vA(k));
          %поиск точек встреч с каждым покупателем при движение
```

```
%в найденную точку пересечения
[NaprA(:,k)]=vstrecha(RA,SAA,S,Achoice);
SNAi(k)=mlength(S,NaprA(:,k));
TimeA(k)=SNAi(k)/vs;
end
VstS=NaprA(:,soch(1));
j=1;
Tsum=TimeA(soch(1));   %время до встречи в точке
пересечения

while (~isempty(find(TimeA==Tsum)))   %осуществление
встреч до встречи в точке пересечения
T=min(TimeA);
meetAi=find(round(TimeA*10^10)==min(round(TimeA*10^10)));
%округление для избежания проблем с точностью
   if length(meetAi)==2
      meetA(j)=meetAi(1);
      meetA(j+1)=meetAi(2);
      order(find(order==meetAi(1)))=[];
      order(find(order==meetAi(2)))=[];
      VstA(:,meetAi)=NaprA(:,meetAi);
      j=j+2;
      fprintf('встретил C%d и C%d в точке %f %f
      через %f единиц времени\n',meetAi(1),meetAi(2),
      VstA(1,meetAi(1)),VstA(2,meetAi(1)),T)
   else
      meetA(j)=meetAi;
      VstA(:,meetAi)=NaprA(:,meetAi);
      j=j+1;
      order(find(order==meetAi))=[];
      fprintf('встретил C%d в точке %f %f через %f
      единиц времени\n',meetAi(1),VstA(1,meetAi(1)),
```

```
        VstA(2,meetAi(1)),T)
      end
    TimeA(meetAi)=[inf];
    end
end
if length(order)>1   %движение в эту точку пересечения
не соответсвует ни второму, ни третьему типу поведения
    fprintf('Недостаточно встреч до встречи в точке
    пересечения, данная стратегия не является равновесной\n\n');
    Tsumsoch(i)=inf;
elseif isempty(order)   %движение в эту точку пересечения
соответствует второму типу поведения
    Tsumsoch(i)=Tsum;
    fprintf(['Встретил всех в порядке ',forprint(1:2*n),'
    за время %f\n\n'],meetA,Tsumsoch(i))
else %движение в эту точку пересечения соответствует
третьему типу поведения; осталась последняя встреча
    a=(NaprA(:,order)-A(:,order))/
    mlength(NaprA(:,order),A(:,order));
    VstA(:,order)=A(:,order)+a*vA(order)*Tsum;
    %поиск положения оставшегося игрока в момент встречи
    %в точке пересечения
    RA=krug(VstS,VstA(:,order),vs,vA(order));
    SA=mline(VstS,VstA(:,order));
    [NaprA]=vstrecha(RA,SA,VstS,VstA(:,order));
    SNAi=mlength(VstS,NaprA);
    T=SNAi/vs;
    fprintf('встретил C%d в точке %f %f через %f единиц
    времени\n',order,NaprA(1),NaprA(2),Tsum+T)
    meetA(n)=order;
    Tsumsoch(i)=Tsum+T;
```

```
            fprintf(['Встретил всех в порядке ',forprint(1:2*n),'
            за время %f\n\n'],meetA,Tsumsoch(i))
        end
    meetorder(:,i)=meetA;
end
Tsumsoch=round(Tsumsoch*(10^10));
j=find(Tsumsoch==min(Tsumsoch));
stratsoch=sochetania(j,:);
meetorder=meetorder(:,j)';
Tsumsoch=min(Tsumsoch)/(10^10);
%поиск максимального выигрыша S среди стратегий, соответствующих
%второму и третьему типу поведения при движении в ближайшую из точек
%пересечения, и соответствующих этому выигрышу стратегий
end
```

Функция, описывающая движение согласно второму и третьему типу поведения дальние точки пересечения двух окружностей, строится аналогичным образом, за исключением нахождения точек пересечения, в которые происходит движение.

```
function [Tsumsoch,meetorder,stratsoch] =
StoAperesecheniemax(S,vs,A,vA,n)
...
[SAA,Achoice] = StodoubleAmax(VstS,vs,VstA,vA,soch );
%строим прямую, проходящую через S и дальнюю точку пересечения
%выбранных (soch) окружностей Аполлония
...
end
```