

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

**Касимова Яна Александровна**

**Выпускная квалификационная работа  
бакалавра**

**Теоретико-игровые модели  
конкуренции в области исследований и  
разработок**

Направление 010400

Прикладная математика, фундаментальная информатика  
и основы программирования

Научный руководитель,  
кандидат физ.-мат. наук,  
доцент  
Зенкевич Н. А.

Санкт-Петербург  
2016

# Содержание

Введение . . . . .	4
Постановка задачи . . . . .	6
Обзор литературы . . . . .	8
Глава 1. Теоретико-игровая модель патентной гонки пуассоновского типа . . . . .	10
1.1. Стохастический процесс . . . . .	10
1.2. Пуассоновский процесс . . . . .	11
1.3. Построение и анализ модели . . . . .	12
1.4. Решение числового примера . . . . .	18
1.5. Результаты численного анализа чувствительности равновесия по Нэшу от параметров модели . . . . .	20
Глава 2. Стохастическая дифференциальная игра патентной гонки . . . . .	23
2.1. Экспоненциальная игра . . . . .	24
2.2. Постановка стохастической дифференциальной игры . . . . .	25
2.3. Решение дифференциальной игры . . . . .	27
2.4. Результаты численного анализа чувствительности равновесия по Нэшу от параметров модели . . . . .	30
Глава 3. Стохастическая многошаговая игра патентной гонки . . . . .	34
3.1. Постановка стохастической многошаговой игры . . . . .	34

3.2. Решение стохастической многошаговой игры . . . . .	37
3.3. Решение числового примера стохастической многошаговой игры . . . . .	41
Выводы . . . . .	46
Заключение . . . . .	47
Список литературы . . . . .	48
Приложение 1. Программа решения игры патентной гонки пуассоновского типа. . . . .	52
Приложение 2. Программа решения дифференциальной игры. . . . .	53
Приложение 3. Программа решения стохастической игры. . . . .	54

# Введение

Инновации в наше время являются главным конкурентным преимуществом на рынке среди фирм, нацеленных на постоянное развитие, устойчивый рост, и лидерство на рынке. Это объясняется, в частности, ускорением темпа изменений, которые происходят в глобальной экономике. Новые технологии быстро устаревают, а вкусы потребителей меняются, что вынуждает фирмы постоянно пересматривать стратегии управления исследованиями и разработками. В наше время остро стоит проблема эффективного управления исследованиями и разработками при различных условиях на рынке. Положим, что на рынке есть монополист и фирма, которая только собирается выйти на рынок. Тогда инновации могут послужить фирме-новичку пропуском на рынок, и в тоже время пошатнуть позиции фирмы-монополиста. Но может быть и наоборот инновации укрепят позиции монополиста и обеспечат уверенностью, что на рынок не проникнет ни одна фирма-новичок. Таким образом монополист и фирма-новичок будут вовлечены в соревнования, целью которых является сделать открытие, которое приведет к производству нового продукта или улучшит технологию производства старого продукта. Такие соревнования мы будем называть патентными гонками. Но патентные гонки ведутся не только

между монополистом и новичком, в них могут участвовать и фирмы, которые занимают схожие позиции на рынке. Тогда открытие может привести к укреплению позиций на рынке фирмы победителя.

Патентные гонки, чаще всего, обладают неопределенной продолжительностью и требуют больших вложений, поэтому фирме, участнику патентной гонки, важно найти оптимальную стратегию, которая максимизирует прибыль фирмы. В связи с этим исследуются модели патентных гонок, наиболее приближенных к реальным условиям, которые помогают найти оптимальную стратегию поведения фирмы в области исследований и разработок.

## Постановка задачи

Целью моего исследования является анализ математических моделей патентных гонок, а именно: патентной гонки пуассоновского типа, патентной гонки в форме стохастической дифференциальной игры, и дискретной модели патентной гонки марковского типа. Для достижения поставленных целей предполагается решить следующие задачи

1. Исследование модели патентной гонки пуассоновского типа и нахождение равновесия по Нэшу;
2. Исследование модели патентной гонки в форме стохастической дифференциальной игры, численный анализ чувствительности равновесия по Нэшу от параметров модели;
3. Построение дискретной модели патентной гонки в форме стохастической многошаговой игры марковского типа;
4. Разработка и программная реализация решения стохастической многошаговой игры.
5. Расчет численных примеров, решения моделей.

Работа организована следующим образом: Глава 1 посвящена описанию модели патентной гонки пуассоновского типа, ее анализу и поиску равновесия по Нэшу в этой модели. В Главе 2

описана модель патентной гонки в форме стохастической дифференциальной игры и проведен численный анализ чувствительности решения от параметров задачи. Глава 3 посвящена построению дискретной модели патентной гонки марковского типа и алгоритма нахождения равновесия по Нэшу. В приложении приведены программы численных расчетов примеров.

## Обзор литературы

В данной работе рассматриваются две модели патентных гонок: патентная гонка пуассоновского типа и патентная гонка как стохастическая дифференциальная игра. Концептуально модель пуассоновской патентной гонки впервые была сформулирована Tirole в [1] и развита в работе Agnion [2]. Эта модель основана на более ранних работах по анализу конкуренции в области исследований и разработок (ИР) [3–5]. В последнее время интерес к такого рода моделям опять вырос, поскольку они лежат в основе стохастических дифференциальных игр в области исследований и разработок [6–8]. Также использовалась работа Шевкопляс Е.В.[9] по исследованию стохастических дифференциальных игр со случайной продолжительностью, в которой подробно разобрана функция выигрыша игрока. Вторая модель патентной гонки в форме стохастической дифференциальной игры между  $N$ -лиц была предложена Dockner [10]. Данная модель основана на ранних работах Reinganum[11], где игроки выбирали свои стратегии независимо от знаний, которые игроки накопили за прошедшее время. Однако в работах Mehlmann[12] стратегии игроков уже зависели от накопленного опыта и знаний. Данная модель получила развитие в работе Keller [13] и послужила построению новой модели, описанной в



работе Doraszleski[14], где время уже имело распределение, отличное от экспоненциального. Различные модификации данной модели патентной гонки были рассмотрены в [15], где прибыль участников патентной гонки зависела от времени, т.е. имела ставку дисконтирования. Построенная в третьей главе модель патентной гонки в форме стохастической многошаговой игры марковского типа была основана на работах Fudenbergа [16], где патентная гонка состояла из двух стадий. Результатом первой стадии являлось промежуточное (вспомогательное) открытие. Затем гонка переходила на вторую стадию, окончанием которой было само открытие и получение патента. При разработке алгоритма использованы результаты работы [17], в которой построено равновесие для игры марковского типа.

# Глава 1. Теоретико-игровая модель патентной гонки пуассоновского типа

В данной главе мы рассмотрим простую модель патентной гонки пуассоновского типа, решим числовой пример и проведем анализ чувствительности равновесия по Нэшу от параметров модели. Для построения модели патентной гонки пуассоновского типа нам потребуются понятия стохастического [13] и пуассоновского процесса.

## 1.1. Стохастический процесс

**Определение 1.1** Дано вероятностное пространство  $\langle \Omega, F, P \rangle$ , где  $\Omega$  — это множество элементарных событий,  $F$  — это  $\sigma$ -алгебра подмножеств множества  $\Omega$ , а  $P$  — это вероятностная мера, заданная на  $F$ . Пусть  $T$  — это временной промежуток,  $T \subset R^+$ . Тогда стохастический процесс это набор случайных величин  $\{x_t\}_{t \in \tilde{T}}$ , определенных на  $\langle \Omega, F, P \rangle$ , и  $x_t \in R^n$ . Стохастический процесс можно представить при помощи случайной функции:

$$(t, \omega) \mapsto x(t, \omega) : T \times \Omega \mapsto R^n.$$

где  $t \in \tilde{T}$ .

**Определение 1.2** Если мы зафиксируем  $t \in T$  то получим

случайную величину:

$$\omega \mapsto x_t(\omega), \omega \in \Omega.$$

Для каждого элементарного события  $\omega \in \Omega$ , траектория случайного процесса определяется по формуле:

$$t \mapsto x_t(\omega), t \in T.$$

**Определение 1.3** Стохастический процесс  $x_t$  — стационарный, если  $x_{t_1}, \dots, x_{t_n}$  и  $x_{t_{(1+s)}}, \dots, x_{t_{(n+s)}}$  имеют одинаковые вероятностные характеристики (плотность распределения, функция распределения) для любого  $n$  и  $s$ , т.е. вероятностные характеристики не меняются при изменении начала отсчета времени наблюдения на произвольную величину  $s$ .

## 1.2. Пуассоновский процесс

Рассмотрим стационарный случайный процесс — пуассоновский процесс [18].

**Определение 1.4** Пуассоновским процессом с параметром  $\lambda > 0$  называют случайный процесс  $x(t, \omega)$ ,  $t \in T = [0, \infty)$ , обладающий свойствами:

1.  $x(0, \omega) \equiv 0$ ;

2. для любых  $N > 1$  и  $t_k \in T$ ,  $k = 1, \dots, N$ :  $0 = t_0 < t_1 < \dots < t_N$  случайные величины  $x(t_k, \omega) - x(t_{k-1}, \omega)$ , будут независимыми

3. для любых  $t_1, t_2 \in T$ , таких, что  $0 \leq t_1 < t_2$ , случайная величина  $x(t_2, \omega) - x(t_1, \omega)$  распределена по закону Пуассона с параметром  $\lambda(t_2 - t_1)$ , т.е.

$$P[x(t_2, \omega) - x(t_1, \omega) = K] = \frac{[\lambda(t_2 - t_1)]^k e^{-\lambda(t_2 - t_1)}}{k!}, \quad k = 0, \dots, N.$$

### 1.3. Построение и анализ модели

В данной главе рассмотрена модель патентной гонки пуассоновского типа между монополистом (игрок 1, А) и новичком (игрок 2, Е). В ней мы учитываем затраты на текущие ИР и не берем в рассмотрение опыт, накопленный двумя игроками ранее, что существенно упрощает нам анализ данной модели. Патентная гонка пуассоновского типа дает ответ на вопрос, является ли монополист более склонным к инновациям, чем новичок. Сам патент, за который борются игроки, может послужить как улучшению технологии производства определенного продукта, так и созданию нового товара в сегменте рынка, который до инновации занимала первая фирма - монополист. Предполагается, что фирма, которая первой осваивает

новую технологию, приобретает и использует патент, который имеет неограниченный срок действия. Конкуренция в области ИР между двумя фирмами характеризуется интенсивностями инвестиций на исследования, заданными в виде функций времени  $x_1(t) \in R_+^1$  и  $x_2(t) \in R_+^1$ ,  $t \in [0, \infty)$ . В каждый момент  $t$ , если ни одна из фирм не сделала открытие, игра, начавшаяся в этот момент, идентична первоначальной. Поэтому стратегии игроков  $x_1$  и  $x_2$  не зависят от времени. Так как мы рассматриваем патентную гонку без памяти, то время, когда фирма  $i$  сделает открытие будет иметь экспоненциальное распределение и вероятность для фирмы сделать открытие в момент  $t$  будет зависеть только от интенсивности ее инвестиций в этот период. Будем предполагать, что функция распределения  $F_i(x, t)$  того, что фирма  $i$  сделает открытие до момента  $t$  при уровне инвестиций  $x$  имеет вид:

$$F_i(x_i, t) = P(T_i(x) \leq t) = 1 - e^{-h(x)t}, t \in [0, \infty), \quad (1.1)$$

где  $T_i(x_i)$  - время, когда фирма  $i$  сделает открытие,  $h(x)$  — заданная дважды дифференцируемая функция, удовлетворяющая следующим условиям:

$$h(x) \geq 0, x \geq 0, \quad (1.2)$$

$$h''(x) \leq 0, x \geq 0, \quad (1.3)$$

$$h'(x) \geq 0, x \geq 0, \quad (1.4)$$

$$\lim_{x \rightarrow \infty} h'(x) = 0 = h(0). \quad (1.5)$$

В предположениях модели:

$$P(T_i(x_i) \in (t, t + dt | T_i(x_i) > t)) = h(x_i) dt, t \in [0, \infty). \quad (1.6)$$

Формула (1.6) следует из свойства экспоненциального распределения, позволяющего не учитывать опыт и знания, которые фирмы накопили ко времени  $t$ , т.е.  $h(x_i) dt$  — это условная вероятность того, что открытие произошло во временном промежутке  $(t, t + dt)$  если до момента  $t$  открытия не было.

Пусть игра начинается в момент  $t_0$  и заканчивается, когда одна из фирм сделала первой открытие, т.е. выиграла патентную гонку. Пусть  $\widehat{T}_i$ -время, когда фирма  $i$  первой сделает открытие в патентной гонке, т.е.  $\widehat{T}_i = \min_{j \neq i} T(x_j)$ . Тогда  $\widehat{T}_i$  также имеет экспоненциальное распределение  $F(t)$ . По определению функции распределения:

$$F(t) = P(\widehat{T}_i < t).$$

По свойству распределения вероятностей

$$P(\widehat{T}_i < t) = 1 - P(\widehat{T}_i \geq t)$$

По условию, что  $\widehat{T}_i = \min_{j \neq i} T(x_j)$ , получаем

$$P(\widehat{T}_i \geq t) = P(\min_{j \neq i} T(x_j) \geq t).$$

$$\begin{aligned}
& P(\min_{j \neq i} T(x_j) \geq t) = \\
& = P(T(x_1) \geq t) P(T(x_2) \geq t) \dots (P(T(x_{i-1}) \geq t))(P(T(x_{i+1}) \geq t)) \dots \\
& \qquad \qquad \qquad P(T(x_n) \geq t).
\end{aligned}$$

Так как  $T_i(x_i)$  — независимые величины. Используем снова определение функции распределения

$$\begin{aligned}
& P(T(x_1) \geq t) P(T(x_2) \geq t) \dots (P(T(x_{i-1}) \geq t))(P(T(x_{i+1}) \geq t)) \dots P(T(x_n) \geq t) \\
& = (1 - F_1(t))(1 - F_2(t)) \dots (1 - F_{i-1}(t))(1 - F_{i+1}(t)) \dots (1 - F_n(t)).
\end{aligned}$$

т. е

$$F(t) = 1 - \prod_{j \neq i} (1 - F_j(t))$$

Тогда

$$\begin{aligned}
& F(t) = 1 - \prod_{j \neq i} (1 - (1 - e^{-h(x_j)t})) \\
& F(t) = P(\widehat{T}_i \leq t) = 1 - e^{-\sum_{j \neq i} h(x_j)t}. \tag{1.7}
\end{aligned}$$

Пусть  $P_1^A$  прибыль монополиста до получения патента,  $P_1^W$  его прибыль, если он выиграл патентную гонку и  $P_1^L$  если проиграл,  $P_2^E$  прибыль новичка, если он первый сделал открытие. Тогда интегральный выигрыш в игре с предписанной продолжительностью  $[t_0, T]$ , выражается следующим образом:

$$\int_{t_0}^T \left[ P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r} \right] e^{-rt} dt. \tag{1.8}$$

где  $r \geq 0$  - заданная процентная ставка.

Так как в нашей постановке момент окончания игры  $\widehat{T}_i$  — величина случайная, то под выигрышем в игре будем понимать математическое ожидание от (1.8), т. е:

$$E \left( \int_{t_0}^T \left[ P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r} \right] e^{-rt} dt \right). \quad (1.9)$$

Таким образом, ожидаемый выигрыш представляет собой следующий интегральный функционал:

$$V^1(x_1, x_2) = \int_{t_0}^{\infty} \int_{t_0}^t \left[ P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r} \right] e^{-rt} dt dF(t). \quad (1.10)$$

Далее, как было показано в [9], с помощью перестановки интегралов в интегральном функционале (1.10), ожидаемый выигрыш может быть представлен в виде:

$$\int_{t_0}^{\infty} \left[ P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r} \right] e^{-rt} (1 - F(t)) dt. \quad (1.18)$$

т. е:

$$V^1(x_1, x_2) = \int_0^{\infty} e^{-(h(x_1)+h(x_2))t} \times \left[ P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r} \right] \times e^{-rt} dt. \quad (1.11)$$

Рассуждая аналогично, ожидаемая прибыль новичка вычисляется следующим образом



Интегральный выигрыш новичка:

$$\int_{t_0}^T (h(x_2) \frac{P_2^E}{r} - x_2) e^{-rt} dt$$

Ожидаемый выигрыш новичка:

$$E \left( \int_{t_0}^T (h(x_2) \frac{P_2^E}{r} - x_2) e^{-rt} dt \right) \Rightarrow$$

$$V^2(x_1, x_2) = \int_{t_0}^{\infty} \int_{t_0}^T (h(x_2) \frac{P_2^E}{r} - x_2) e^{-rt} dt dF(t)$$

Переходим от двойного интеграла к следующему виду:

$$\int_{t_0}^{\infty} (h(x_2) \frac{P_2^E}{r} - x_2) e^{-rt} (1 - F(t)) dt$$

т. е:

$$V^2(x_1, x_2) = \int_0^{\infty} e^{-(h(x_1)+h(x_2))t} \times (h(x_2) \frac{P_2^E}{r} - x_2) \times e^{-rt} dt. \quad (1.12)$$

Вычисляя интегралы в (1.11) и (1.12), мы получаем выигрыши игроков в явном виде:

$$V^1(x_1, x_2) = \frac{P_1^A - x_1 + h(x_1) \frac{P_1^W}{r} + h(x_2) \frac{P_1^L}{r}}{r + h(x_1) + h(x_2)}, \quad (1.13)$$

$$V^2(x_1, x_2) = \frac{h(x_2) \frac{P_2^E}{r} - x_2}{r + (h(x_1) + h(x_2))}. \quad (1.14)$$

В итоге мы построим бесконечную игру двух лиц в нормальной форме:  $G_2 = \langle X, X, V^1, V^2 \rangle$ , где  $X = \{x | x \geq 0\}$  - множества стратегий игрока 1, и 2, а функции выигрыша  $V^1, V^2$  заданы

(1.13) и (1.14) соответственно.

Рассмотрим вторые частные производные от  $V^1$  и  $V^2$ :

$$\frac{\partial^2 V^1(x_1, x_2)}{\partial^2 x_1} = \frac{h''(x_1)(P_1^W + h(x_2) \frac{P_1^W}{r} - P_1^A + x_1 - h(x_2) \frac{P_1^L}{r})}{(r + h(x_1) + h(x_2))^4}, \quad (1.15)$$

$$\frac{\partial^2 V^2(x_1, x_2)}{\partial^2 x_2} = \frac{h''(x_2)(P_2^E + h(x_1) \frac{P_2^E}{r} - x_2)}{(r + h(x_1) + h(x_2))^4}, \quad (1.16)$$

Учитывая полученные результаты (1.15) - (1.16) и предположения (1.2) - (1.5), а также теорему [17], в игре  $G_2$  существует равновесие по Нэшу в чистых стратегиях, которое находится из условий первого порядка:

$$\begin{aligned} (h'(x_1^*) + \frac{1}{r} h'(x_1^*) h(x_2^*)) P_1^W - h'(x_1^*) P_1^A - r - \\ - h(x_1^*) - h(x_2^*) - h(x_2^*) \frac{P_1^L}{r} + x_1 h'(x_1^*) = 0; \end{aligned} \quad (1.17)$$

$$(h'(x_2^*) + \frac{1}{r} h'(x_2^*) h(x_1^*)) P_2^E - r - h(x_1^*) - h(x_2^*) + x_2^* h'(x_2^*) = 0 \quad (1.18)$$

В общем случае система, из условий (1.17) - (1.18) не имеет аналитического решения.

#### 1.4. Решение числового примера

В данном подразделе мы рассмотрим базовый пример при значениях постоянных задачи, удовлетворяющих всем ограниче-

ниям, которые были получены при нахождении ее решений. Подсчеты, результаты которых будут приведены в этом разделе, осуществлялись при помощи программы, написанной в среде MATLAB (см.приложение 1).

Свойствам (1.2)-(1.5), заданным для  $h(x)$  удовлетворяет класс функций:  $h(x) = a x^{\frac{1}{k}}$ , при  $k \in Z, k > 1, a > 0$ , который и выбран в качестве базового примера:

$$h(x) = a x^{\frac{1}{k}} \geq 0, x \geq 0, \quad (1.19)$$

$$h'(x) = \frac{a}{k x^{-\frac{k-1}{k}}} \geq 0, x \geq 0, \quad (1.20)$$

$$h''(x) = -\frac{a(k-1)}{k^2 x^{-\frac{2k-1}{k}}} \leq 0, x \geq 0, \quad (1.21)$$

$$\lim_{x \rightarrow \infty} h'(x) = \frac{a}{k x^{-\frac{k-1}{k}}} = 0. \quad (1.22);$$

Для данного класса функций  $h(x)$  численно найдено равновесие по Нэшу.

В частности, если:  $a= 1; k= 2; P_1^W= 11; P_2^E= 6; P_1^L= 4; P_1^A= 5; r= 0,9;$

то равновесие по Нэшу имеет вид  $\hat{x}_1= 1,6985, \hat{x}_2= 0,9864$ , а выигрыши в равновесии равны:  $\hat{V}^1= 0, \hat{V}^2= 4,6947$  соответственно (см.приложение 1).

Прибыль монополиста, если он сделает открытие, больше чем его прибыль до открытия и больше чем прибыль новичка, если

открытие сделает новичок. Это значит, что расходы монополиста на ИР будут больше чем у новичка. Но так как процентная ставка близка к единице, выигрыш новичка в ситуации равновесия по Нэшу будет больше, чем у монополиста.

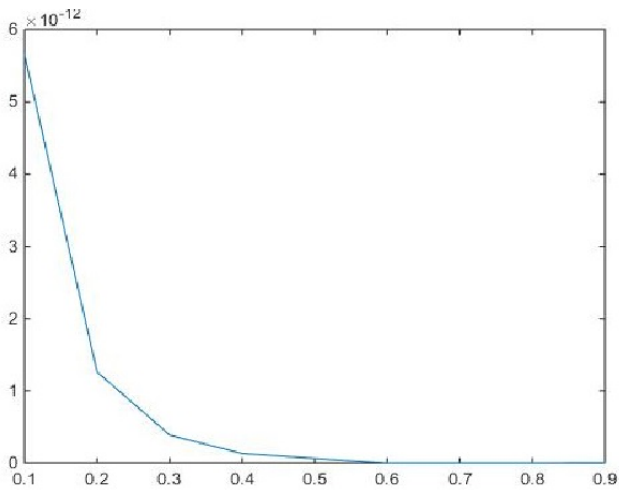
### **1.5. Результаты численного анализа чувствительности равновесия по Нэшу от параметров модели**

Анализ чувствительности представляет собой оценку влияния параметров модели на ожидаемые выигрыши игроков в положении равновесия по Нэшу. Для численного анализа мы выбрали параметры  $r$ ,  $k$ , поскольку от них величины  $V^1(r, k)$  и  $V^2(r, k)$  зависят не линейно. В этом параграфе численный анализ чувствительности проведен для изложенного базового примера в §1.4. Рассмотрим зависимость между прибылью монополиста в положении равновесия по Нэшу  $V^1$  от процентной ставки  $r$ , т. е. зафиксируем все остальные параметры модели и будем изменять только параметр  $r$ . Пусть  $r$  у нас изменяется от 0.1 до 0.9, по определению процентной ставки, при ее увеличении уменьшается ожидаемый доход, и, действительно, при возрастании процентной ставки мы видим как прибыль монополиста в положении равновесия по Нэшу уменьшается. Это можно заметить на рис. 1.1(a). Прибыль новичка в положении

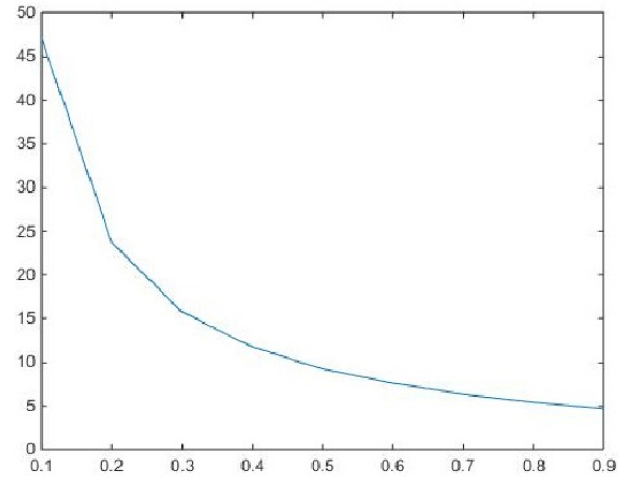
равновесия по Нэшу, аналогично прибыли монополиста в положении равновесия, уменьшается при возрастании процентной ставки  $r$  от 0.1 до 0.9, что видно на рис.1.1(b). Изучим зависимость прибыли монополиста в положении равновесия по Нэшу от параметра  $k$ , напомним что параметр  $k$  входит в состав функции  $h(x)$

$$h(x) = a x^{1/k}$$

Учитывая ограничения на параметр  $k$ , что  $k \in \mathbb{Z}$ ,  $k > 1$ , зафиксируем все остальные параметры модели и будем изменять  $k$  от 2 до 10. Зависимость прибыли монополиста в положении равновесия от параметра  $k$  представляет собой кусочно-линейную функцию, как видно на рис.1.2(a), но поскольку масштаб оси ординат —  $10^{-11}$ , то колебания незначительны и прибыль монополиста в положении равновесия приблизительно равна нулю. Рассмотрим зависимость прибыли новичка в положении равновесия по Нэшу от параметра  $k$ . Также, учитывая ограничения на  $k$ , будем изменять данный параметр в таком же диапазоне, как и в случае с прибылью монополиста, т. е от 2 до 10. Заметим, что при увеличении параметра  $k$  уменьшается прибыль новичка в положении равновесия по Нэшу, что видно на рис.1.2(b).

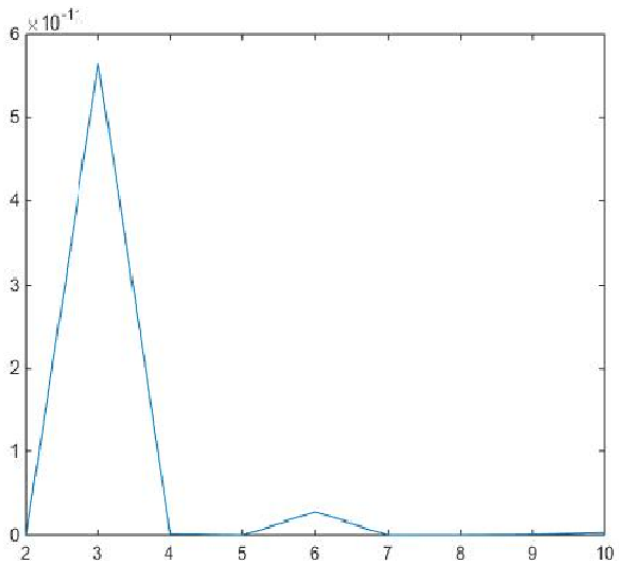


a)

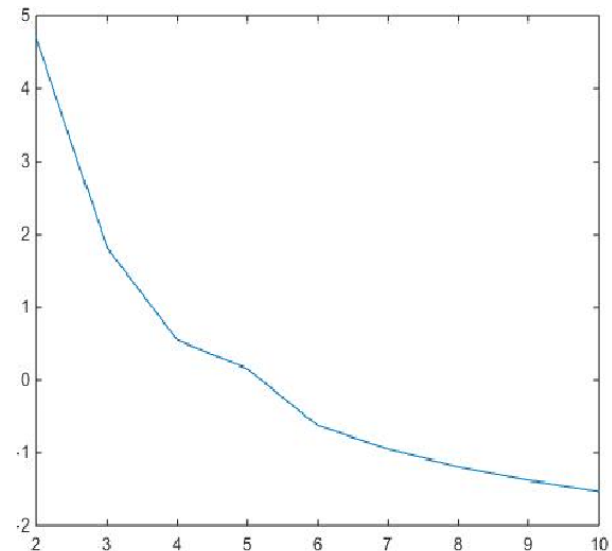


b)

Рис. 1.1: Зависимость  $V^1(r)$  и  $V^2(r)$  от параметра  $r$



a)



b)

Рис. 1.2: Зависимость  $V^1(k)$  и  $V^2(k)$  от параметра  $k$

## Глава 2. Стохастическая дифференциальная игра патентной гонки

Теоретико-игровой подход к ИР рассматривает инновации как нечто развивающееся в конкурентной среде. Иногда одна мысль, порожденная ИР служит началом технологическому прорыву. Усилия, которые фирма тратит на ИР, влияют на вероятность, что данная фирма первой сделает открытие. Когда фирма выигрывает патентную гонку предполагается, что она занимает позицию монополиста на рынке[10].

Стохастическая дифференциальная игра в области ИР основывается на трех предположениях. Во-первых, ни одна из фирм заранее не знает сколько она должна потратить на ИР. Во-вторых, возможны несколько путей, приводящих к победе в патентной гонке. В-третьих, область ИР требует затрат, но она приводит к накоплению опыта и знаний, которые повышают вероятность победы.

В данной главе мы рассмотрим стохастическую дифференциальную игру в области ИР, модель которой была предложена в [10], а затем развита в [13], найдем ее решение и проведем анализ чувствительности равновесия по Нэшу от параметров модели. Для анализа модели нам потребуется понятие экспоненциальной игры[10].

## 2.1. Экспоненциальная игра

Рассмотрим подкласс стохастических дифференцированных игр — экспоненциальные игры. Это класс дифференциальных игр, в которых уравнение, описывающее состояние системы представимо в виде:

$$\dot{x}(t) = f(u^1(t), \dots, u^N(t), t), \quad (2.1)$$

где  $u^i(t) \in R^m$ ,  $\forall i = 1, \dots, n$  — это управления (положим, что в игре участвуют  $N$  игроков),  $x(t) \in R^n$  — вектор, описывающий состояние системы в момент времени  $t$ . Функция выигрыша игрока  $i$  равна:

$$J^i = \int_0^T e^{-rt} L^i(u^1(t), \dots, u^N(t), t) e^{-\lambda_i x(t)} dt, \quad (2.2)$$

где  $\lambda^i \in R^n$  — постоянный вектор. Особенность экспоненциальных игр, заключается в том, что вектор, описывающий состояние системы  $x(t)$  входит в функцию выигрыша в качестве степени экспоненты. Экспоненциальную игру можно преобразовать в линейную игру, т. е. вектор  $x(t)$  будет входить в функцию выигрыша линейно:

$$y^i(t) = e^{-\lambda_i x(t)}, \quad i = 1, 2. \quad (2.3)$$



Дифференцируя (2.1) по времени, получим функцию выигрыша в виде:

$$J^i = \int_0^T e^{-rt} L^i(u^1(t), \dots, u^N(t), t) y^i(t) dt. \quad (2.4)$$

Таким образом, из экспоненциальной игры мы получили линейную игру.

## 2.2. Постановка стохастической дифференциальной игры

Пусть  $N$  фирм(игроков) участвуют в патентной гонке.

Время, когда  $i$ -ая фирма сделает открытие — это случайная величина  $\tau_i$ , имеющая распределение вероятностей  $F_i(t) = \mathbf{P}(\tau_i \leq t)$ . Предположим, что между фирмами не происходит обмена знаний и опыта в области ИР, тогда  $\tau_i$  стохастически независима. Обозначим время, когда одна из фирм сделала первой открытие  $\tau = \min(\tau_i \mid i=1,2,\dots,N)$ . Фирма, пусть будет под номером  $k$ , у которой  $\tau_k = \tau$ , является победителем в патентной гонке:

$$\mathbf{P}(\tau \leq t) = 1 - \prod_{i=1}^N [1 - F_i(t)]. \quad (2.5)$$

Пусть  $u_i(t) \geq 0$  описывает сколько усилий фирма  $i$  потратила на ИР. И рассмотрим (условную) вероятность что фирма  $i$  первой сделает открытие в момент  $t$ , при условии, что ранее ни одна из

фирм не сделала открытие, также предполагается, что данная вероятность пропорциональна  $u_i(t)$ . Тогда имеем:

$$\frac{\dot{F}_i(t)}{[1 - F_i(t)]} = \lambda u_i(t) = h_i(t), \quad F_i(0) = 0 \quad (2.6)$$

где  $\lambda \geq 0$  константа.

Пусть  $P_I$  — это чистая прибыль, которую получит фирма, сделавшая открытие первой, предполагается что  $P_I$  константа, следовательно не зависит от момента времени, когда произошло открытие.  $P_F$  — это чистая прибыль, которую получают остальные участники патентной гонки после открытия, предполагается что  $P_F$  также константа.

Предположим, что  $P_I \geq P_F$ , а момент окончания патентной гонки фиксирован и равен  $T$ . Затраты на ИР равны  $u_i(t)^2$ , тогда ожидаемая прибыль игрока  $i$  равна:

$$E \left( \int_0^T \left[ P_I h_i(t) + P_F \sum_{j \neq i} h_j(t) - \frac{e^{-rt}}{2} u_i(t)^2 \right] dt \right). \quad (2.7)$$

Преобразуем по теореме о двойных интегралах [3] и подставим (2.6) в (2.7) получим:

$$\int_0^T \left[ \lambda P_I u_i(t) + \lambda P_F \sum_{j \neq i} u_j(t) - \frac{e^{-rt}}{2} u_i(t)^2 \right] \prod_{i=1}^N [1 - F_i(t)] dt. \quad (2.8)$$

Первое слагаемое в формуле (2.8) отражает чистую прибыль игрока при условии, что он сделал открытие первым, второе

слагаемое отражает чистую прибыль игрока, если он проиграл, а третье слагаемое — это затраты игрока на ИР.  $r$  — это процентная ставка.

Таким образом мы получили стохастическую дифференциальную игру, где  $u_1, u_2, \dots, u_N$  — это управления (стратегии).

Предположим, что в самом начале патентной гонки игроки находились в состоянии равном нулю, затем когда  $i$ -ая фирма сделала первой открытие, состояние перешло в  $i$  и осталось там навсегда. Время переключения из одного состояния в другое —  $\tau$ .

### 2.3. Решение дифференциальной игры

Проинтегрировав (2.6) получаем:

$$-\ln(1 - F_i(t)) = \lambda m_i(t). \quad (2.9)$$

где:

$$\dot{m}_i(t) = u_i(t), \quad m_i(0) = 0. \quad (2.10)$$

подставляя (2.9) в (2.8) получим:

$$\int_0^T e^{-\lambda \sum_{j=1}^N m_j(t)} \left[ \lambda P_I u_i(t) + \lambda P_F \sum_{j \neq i} u_j(t) - \frac{e^{-rt}}{2} u_i(t)^2 \right] dt \quad (2.11)$$

$m_i(t)$  описывает состояние  $i$ -ого игрока в момент времени  $t$ , т.е. сколько опыта и знаний накопил  $i$ -ый игрок ко времени  $t$ .

Введем новую переменную  $y(t)$ :

$$y(t) = e^{-\lambda \sum_{j=1}^N m_j(t)}. \quad (2.12)$$

Игроки не знают в каком состоянии находятся их соперники, т.е. сколько опыта они накопили к моменту  $t$ , но они знают значение  $y(t)$  в каждый момент  $t$ , которое описывает сколько в целом игроки накопили опыта и знаний в заданной области ИР.

Дифференцируя (2.12) по  $t$  получаем:

$$\dot{y}(t) = -\lambda y(t) \sum_{j=1}^N u_j(t), y(0) = 1. \quad (2.13)$$

Таким образом мы перешли от экспоненциальной игры к линейной следующего вида:

$$\int_0^T y(t) \left[ \lambda P_I u_i(t) + \lambda P_F \sum_{j \neq i} u_j(t) - \frac{e^{-rt}}{2} u_i(t)^2 \right] dt,$$

$$\dot{y}(t) = -\lambda y(t) \sum_{j=1}^N u_j(t), y(0) = 1.$$

Для нахождения равновесия по Нэшу воспользуемся принципом максимума Понтрягина[7]

Построим гамильтониан для  $i=1,2,\dots,N$ :

$$H^i(y(t), u_i(t), \mu_i(t), t) = y(t) \left[ \lambda P_I u_i(t) + \lambda P_F \sum_{j \neq i} u_j(t) - \frac{e^{-rt}}{2} u_i(t)^2 \right]$$

$$-\mu_i(t) \lambda y(t) \left[ u_i(t) + \sum_{j \neq i}^N u_j(t) \right] \quad (2.14)$$

Так как по построению гамильтониана  $H^i(y(t), u_i(t), \mu_i(t), t)$  видно, что он вогнут по  $u_i(t)$  то мы можем найти максимум из условий первого порядка.

Тогда используем условие оптимального управления:

$$\frac{\partial H(y(t), u_i(t), \mu_i(t), t)}{\partial u_i(t)} = 0 \quad (2.15)$$

Продифференцировав  $H^i(y(t), u_i(t), \mu_i(t), t)$  по  $u_i(t)$  получим:

$$u_i(t) = \lambda e^{rt} (P_I - \mu_i(t)), \quad (2.16)$$

Уравнения Эйлера-Лагранжа:

$$\dot{\mu}_i(t) = -\frac{\partial H^i(y(t), u_i(t), \mu_i(t), t)}{\partial y(t)}, \quad \mu_i(T) = 0. \quad (2.17)$$

$$u_i(t) = -b(t) \lambda e^{rt}. \quad (2.18)$$

Таким образом мы получили  $2N + 1$  уравнений для  $2N + 1$  переменных  $y, \mu_i, u_i, i = 1, 2, \dots, N$ .

Подставим (2.17) в (2.13) и получим:

$$\dot{y}(t) = -\lambda^2 y(t) N b(t) e^{rt}, \quad y(0) = 1. \quad (2.19)$$

Используя (2.17), (2.18) и (2.19) приходим к уравнению Бернул-

ли:

$$\dot{b}(t) = -\frac{\lambda^2 e^{rt}}{2} \{(2N-1)b^2(t) + 2b(t)(1-N)(P_F - P_I)\}, \quad b(T) = -P_I. \quad (2.20)$$

которое решаем при помощи замены  $g(t) = \frac{1}{b(t)}$ . Подставим  $b(t) = \frac{1}{g(t)}$  в уравнение (2.20), получаем:

$$\dot{g}(t) - 2(1-N)(P_F - P_I) \frac{\lambda^2 e^{rt}}{2} g(t) = \frac{\lambda^2 e^{rt}}{2} (2N-1). \quad (2.21)$$

приходим к линейному дифференциальному уравнению первого порядка, решив которое методом вариации постоянной, получаем решение:

$$b(t) = -\frac{2P_I(P_I - P_F)(N-1)}{(2N-1)P_I - [P_I + 2(N-1)P_F]e^{\frac{1}{r}(P_I - P_F)(N-1)\lambda^2(e^{rt} - e^{rT})}}. \quad (2.22)$$

Подставив данное решение в (2.17) получим:

$$u^*(t) = \frac{2\lambda P_I(P_I - P_F)(N-1)e^{rt}}{(2N-1)P_I - [P_I + 2(N-1)P_F]e^{\frac{1}{r}(P_I - P_F)(N-1)\lambda^2(e^{rt} - e^{rT})}}. \quad (2.23)$$

Из (2.23) видно, что стратегии у игроков одинаковые в ситуации равновесия по Нэшу.

## 2.4. Результаты численного анализа чувствительности равновесия по Нэшу от параметров модели

Получив формулу (2.23) мы можем провести анализ зависимости  $u^*(t)$ ,  $\mu^*(t)$  от времени, а также проанализировать зависи-

мость  $u^*(r, P_I, P_F)$  от параметров модели  $r$ ,  $P_I$  и  $P_F$ . Рассмотрим зависимость  $u_i(t)$  от времени  $t$ . Изменяя  $t$  0.1 до 1, при фиксированных параметрах модели:  $P_I = 50$ ,  $P_F = 2$ ,  $r = 0.1$ ,  $T = 1$ ,  $\lambda = 7$ , а количество игроков  $N = 5$ . На рис. 2.1(а) видно, что  $u^*(t)$  возрастает, а поскольку расходы игроков на ИР равны  $u^*(t)^2$ , то тогда можно сделать вывод, что расходы игроков на ИР возрастают с увеличением времени. Теперь по формуле (2.16):

$$\mu^*(t) = P_I - \frac{u^*(t)}{\lambda e^{rt}}. \quad (2.24)$$

поскольку мы знаем  $u^*(t)$ , подставим его в (2.24):

$$\mu^*(t) = \frac{-2 P_I (P_I - P_F) (N - 1)}{(2 N - 1) P_I - [P_I + 2 (N - 1) P_F] e^{\frac{1}{r} (P_I - P_F) (N - 1) \lambda^2 (e^{rt} - e^{rT})}} + P_I \quad (2.31)$$

Рассмотрим зависимость  $\mu^*(t)$  от  $t$ , изменяя  $t$  0.1 до 1, при аналогичных значениях параметров модели. На рис.2.1(б) видно, что  $\mu^*(t)$  убывает при увеличении времени, действительно, так как:

$$u^*(t) = -b(t) \lambda e^{rt},$$

$$\mu^*(t) = b_i(t) + P_I.$$

то при увеличении времени  $t$   $u^*(t)$  будет возрастать, а  $\mu^*(t)$  будет убывать. Проанализируем зависимость  $u^*(\lambda, P_I, P_F)$  от параметров  $\lambda$ ,  $P_I$  и  $P_F$ . Пусть параметр  $\lambda$  изменяется от 1 до 10,

а другие параметры модели имеют значения:  $P_I = 50$ ,  $P_F = 2$ ,  $r = 0.1$ ,  $T = 1$ ,  $t = 0.1$ ,  $N = 5$ . Воспользуемся формулой (2.23) и получим, что при увеличении  $\lambda$  увеличивается и  $u^*(\lambda)$ , что видно на рис. 2.2(a). Пусть теперь изменяется параметр  $P_F$  от 10 до 40, при заданных значениях других параметров модели:  $P_I = 50$ ,  $r = 0.1$ ,  $T = 1$ ,  $\lambda = 7$ ,  $N = 5$ . На рис. 2.2(b) видно, что при увеличении данного параметра, уменьшается  $u^*(P_F)$ , т. е уменьшаются расходы игроков на ИР. Положим, что теперь изменяется параметр  $P_I$ , при заданных значениях других параметров модели:  $P_F = 2$ ,  $r = 0.1$ ,  $T = 1$ ,  $\lambda = 7$ ,  $N = 5$ . На рис.2.3 видно, что при увеличении параметра  $P_I$ , увеличивается и  $u^*(P_I)$ , действительно, при увеличении выигрыша, увеличивается и заинтересованность игроков в победе и, следовательно, они тратят на ИР больше.

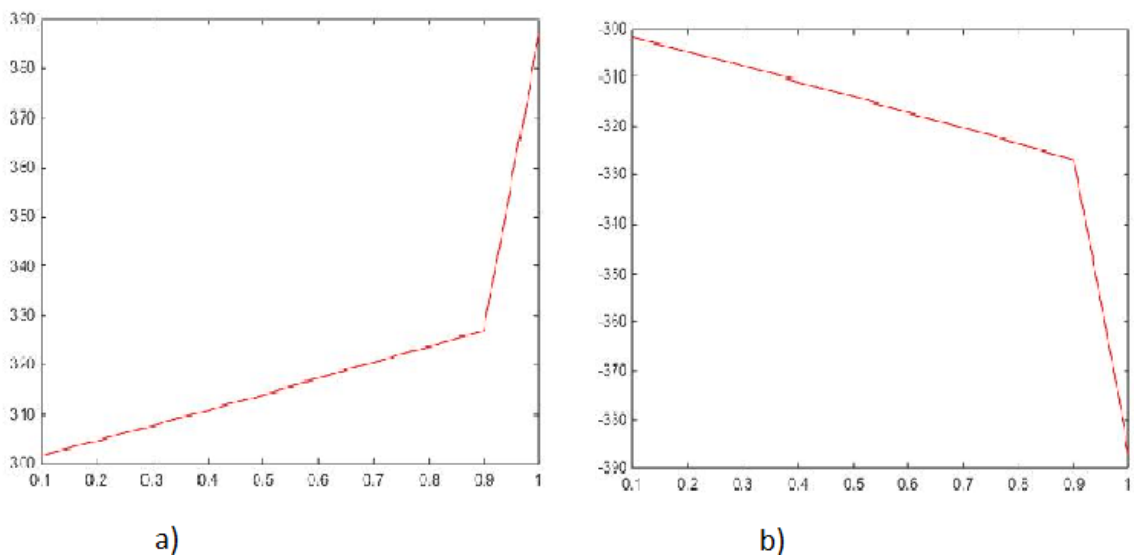


Рис. 2.1: Зависимость  $u^*$ ,  $\mu^*$  от  $t$



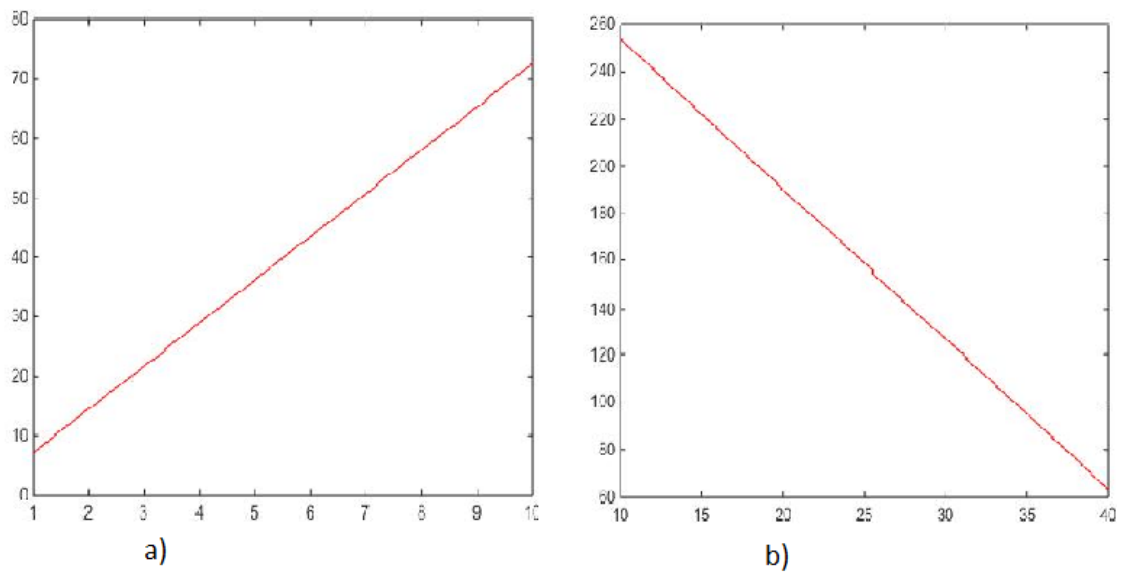


Рис. 2.2: Зависимость  $u^*(t)$  от  $r, P_F$

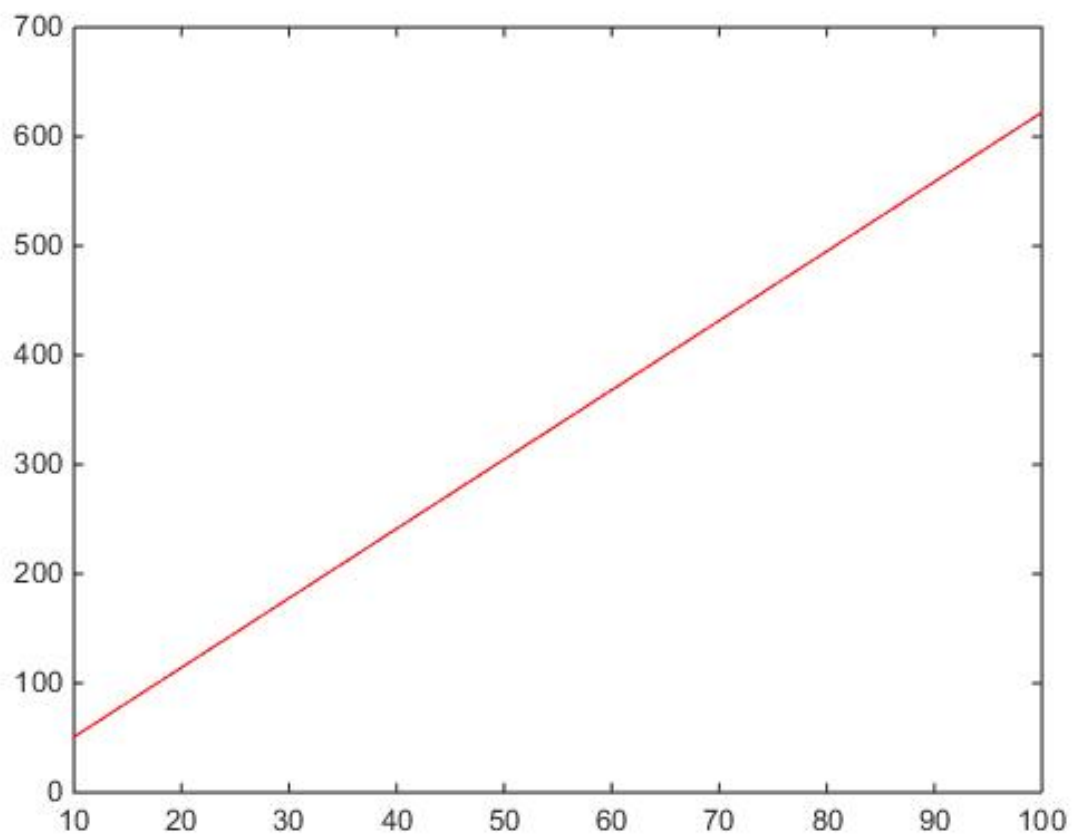


Рис. 2.3: Зависимость  $u^*(t)$  от  $P_I$

# Глава 3. Стохастическая многошаговая игра патентной гонки

В данной главе представлена модель патентной гонки в форме стохастической многошаговой игры марковского типа между  $N$  игроками, построенная на основании рассмотренных моделей патентных гонок и модели, описанной в [16]. Также представлен разработанный алгоритм для нахождения равновесия по Нэшу. В конце главы представлен пример игры двух лиц, для которой получена в явном виде ситуация равновесия по Нэшу.

## 3.1. Постановка стохастической многошаговой игры

Рассмотрим стохастическую многошаговую игру марковского типа на графе [17]. Для этого введем несколько определений.

**Определение 3.1** Пара  $(X, E)$  называется графом, если  $X$  — конечное множество, элементы которого называют вершинами или узлами, а  $E$  — множество упорядоченных пар на  $X$ , т.е. подмножество множества  $X \times X$ , элементы которого называют ориентированными ребрами или дугами.

Ребро, концы которого совпадают, т.е.  $e = (x, x)$ , где  $e \in E$ , называется петлей. Граф будем обозначать символом  $G$

**Определение 3.3** Последовательность  $e = (e_1, e_2, \dots, e_k, \dots)$

дуг, такая, то конец каждой предыдущей дуги совпадает с началом следующей называется путем в графе  $G$ . Длина пути  $e = (e_1, e_2, \dots, e_k)$  — это число  $l(e) = k$  равное количеству дуг в последовательности  $e$ .

**Определение 3.5** Взвешенный граф — это граф, каждому ребру которого поставлено в соответствие некоторое значение (вес ребра)

Связным графом будем называть граф, в котором между любой парой вершин существует как минимум один путь.

Рассмотрим патентную гонку в форме стохастической многошаговой игры марковского типа  $\tilde{\Gamma}$ , на связном взвешенном графе  $G$ , в котором существует единственная вершина  $x_0$ , такая, что не существует ребра  $e = (y, x_0)$ , для  $\forall y \in X$ . Причем будем предполагать, что  $D = \{0, \varepsilon, 2\varepsilon, 3\varepsilon, \dots, 1\}$  — это множество состояний игроков, где  $d_l^{c_k} \in D$  — состояние игрока  $l$  на  $c$ -ом шаге в  $k$ -ом узле графа  $G$ . На первом шаге игры все игроки находятся в узле  $x_0$  и в состоянии  $d_i^{00} = 0$ , для  $\forall i = \{1, \dots, n\}$ , т.е. они не имеют накопленного опыта и знаний в области ИР. Игра состоит из  $\theta$  шагов, т.е. максимальное количество шагов, которое может быть в игре. Получаем, что  $\theta$  равна:

$$\theta = \frac{1}{\varepsilon} 2 - 1. \quad (3.1)$$

На первом шаге, как говорилось ранее, состояния игроков

$d_i^{00} = 0$ . Однако при каждом переходе на следующий шаг игрок может изменить свое текущее состояние увеличив его на  $\varepsilon$  либо остаться в текущем состоянии, т.е. игрок либо получает опыт и знания необходимые для патента, либо нет. Когда у игрока  $i$  состояние  $d_i^{ck} = 1$  то значит на  $c$ -ом шаге он сделал открытие и выиграл патентную гонку. На каждом шаге игры игроки узнают информацию о состоянии своих соперников, т.е. рассматривается игра с полной информацией. Целью каждого игрока, как и в прошлых моделях патентных гонок, будет являться максимизация ожидаемой прибыли.

Рассмотрим поподробнее структуру графа  $G$ .

Каждый  $j$ -ый узел графа  $G$  ( $j = 1, 2, 3, \dots$ ) состоит из:

1. Игрового элемента — игры  $n$  лиц в нормальной форме  $\Gamma^j$

$$\Gamma^j = \langle \bar{N}, U_1^j, \dots, U_n^j, V_1^j, \dots, V_n^j \rangle,$$

где  $\bar{N}$  — множество игроков, одинаковое для всех игр  $\Gamma^j$ ,  $j = 0, 1, 2, \dots, m$ , где  $m$  — количество узлов в графе  $G$ ,  $U_i^j$  ( $i = 1, \dots, n$ ) — множество стратегий  $i$ -ого игрока в игре  $\Gamma^j$ ,  $V_i^j$  — функция выигрыша  $i$ -ого игрока в игре  $\Gamma^j$ . Будем полагать, что  $u_i^j$  — это расход  $i$ -ого игрока на ИР,  $U_i^j$  конечно для любого игрока и любого игрового элемента.

2. Информации о состоянии игроков в патентной гонке —

$$d_1^{cj}, \dots, d_n^{cj}.$$

Будем предполагать, что самый длинный путь в графе  $G$  называется длиной игры  $\Gamma$ , т.е. максимальное количество шагов в игре  $\Gamma$

### 3.2. Решение стохастической многошаговой игры

Для построения алгоритма нахождения равновесия по Нэшу [177] в стохастической многошаговой игре марковского типа нам потребуется ввести несколько определений

**Определение 3.6** Информационное множество  $L_{i+1}$  — множество узлов графа  $G$ , в которые игроки могут перейти при переходе из  $i$ -ого шага игры в  $i + 1$  шаг.

**Определение 3.7** Ситуация в игре  $\Gamma$   $u^* = (u_1^*, \dots, u_n^*)$ , где  $u_1^* = (u_1^{0*}, \dots, u_1^{m*})$  называется равновесием по Нэшу, если имеет место неравенство

$$K_i((u_1^*, \dots, u_i^*, \dots, u_n^*)) \geq K_i((u_1^*, \dots, u_i, \dots, u_n^*))$$

для всех  $u_i \in U_i$ ,  $i \in N$ . Поскольку мы рассматриваем многошаговую игру, то нам необходимо ввести определение подыгры [17]

**Определение 3.8** Подыгра  $\hat{\Gamma}^z$  это часть игры  $\tilde{\Gamma}$ , удовлетворяющая следующим условиям:

1.  $\hat{\Gamma}^z$  начинается в узле  $x_z$ , где  $x_z \in E$ ;
2. Подыгра  $\hat{\Gamma}^z$  разыгрывается на графе, который содержит все узлы исходного графа, в которые можно попасть из узла  $x_z$  за произвольное число шагов.

**Определение 3.9** Ситуация в игре  $\tilde{\Gamma}$   $u^* = (u_1^*, \dots, u_n^*)$ , где  $u_1^* = (u_1^{0*}, \dots, u_1^{m*})$  — абсолютное равновесие по Нэшу[17] в игре  $\tilde{\Gamma}$ , где  $m$  — количество узлов с игровыми элементами, если  $u^{*z} = (u_1^{*z}, \dots, u_n^{*z})$  — равновесие по Нэшу в подыгре  $\hat{\Gamma}^z$ , где  $u_i^{*z}$  — сужение стратегии  $u_i^*$  на подыгру  $\hat{\Gamma}^z$ .

Поскольку мы рассматриваем многошаговую игру с полной информацией на конечном графе, то по теореме [19] в игре будет существовать ситуация абсолютного равновесия по Нэшу.

Если в  $j$ -ом узле хотя бы одна из  $m_i^{cj} = 1$  ( $i = 1, 2, \dots, n$ ), то в данном узле не разыгрывается игра  $\Gamma^j$ , а сразу приписываются выигрыши каждому из игроков, исходя из состояния игроков в  $j$ -ом узле:

$$V_i^j = \begin{cases} \frac{V}{c}, & m_i^{cj} = 1 \\ 0, & m_i^{cj} \leq 1. \end{cases} \quad (3.2)$$

где  $c$  — это количество игроков, у которых  $m_i^{cj} = 1$ , т.е. количество игроков, которые сделали открытие.  $V$  — это прибыль от открытия.

Вес ребра, соединяющего вершины  $x_s$  и  $x_k$  — это вероятность перехода игроков из  $s$ -ой вершины графа, находящейся в информационном множестве  $L_c$  в  $k$ -ую вершину, обозначим эту вероятность  $p_{sk}^c$ .  $p_{sk}^c$  зависит от стратегий, которые игроки выбрали в игре  $\Gamma^s$ , т.е.  $p_{sk}^c = p_{sk}^c(u_1^s, \dots, u_n^s)$ , причем при заданном наборе стратегий  $u_1^s, \dots, u_n^s$  в  $s$ -ой вершине вероятности перехода из этой вершины в сумме дают единицу.

$$\sum_{k \in L_{c+1}} p_{sk}^c = 1.$$

Заметим, что если  $k$ -ый узел не принадлежит  $L_{c+1}$ , то вероятность перехода из вершины  $x_s \in L_c$  в вершину  $x_k$   $p_{sk}^c = 0$ . Поскольку в данной главе мы рассматриваем патентную гонку как многошаговую игру, то воспользуемся принципом оптимальности Беллмана: управление, в нашем случае это стратегия игрока  $u_i^j$ , на каждом шаге должно быть оптимальным с точки зрения процесса в целом, т.е. с какой бы вершины мы не начали, оптимальное управление останется оптимальным. На основании принципа оптимальности, абсолютное равновесие по Нэшу определяется методом обратной индукции начиная с  $j$ -ого узла, который принадлежит информационному множеству  $L_{\theta-1}$ , в котором игроки находятся в состояниях —  $(d_1^{\theta-1j} = 1 - \varepsilon, \dots, d_n^{\theta-1j} = 1 - \varepsilon)$ . В нем разыгрывается игровой

элемент  $\Gamma^j$ . Функция выигрыша игрока  $i$  будет иметь вид

$$V_i^j = \sum_{q \in L_\theta} p_{jq}^{\theta-1} (u_1^j, \dots, u_n^j) (V_i^q - u_i^j) \quad (3.3)$$

где  $V_i^q$  определяются по формуле (3.2). Решая игру  $\Gamma^j$  в результате получаем ситуацию равновесия по Нэшу в  $\Gamma^j$  и выигрыши в ситуации равновесия  $V_1^{j*}, \dots, V_n^{j*}$

$$V_i^{j*} = \sum_{q \in L_\theta} p_{jq}^{\theta-1} (u_1^{j*}, \dots, u_n^{j*}) (V_i^q - u_i^{j*}). \quad (3.4)$$

где  $u_i^{j*}$  — это оптимальная по Нэшу стратегия игрока  $i$ . Затем рассматриваем узел  $x_k \in L_{\theta-2}$ , в котором у игрока  $i$  состояние  $d_i^{\theta-2j} = 1 - 2\varepsilon$ , а у остальных игроков состояния идентичны их состояниям в узле  $x_j$ , т.е. игроки находятся в состояниях —  $(d_1^{\theta-2j} = 1 - \varepsilon, \dots, d_{i-1}^{\theta-2j} = 1 - \varepsilon, d_i^{\theta-2j} = 1 - 2\varepsilon, d_{i+1}^{\theta-2j} = 1 - \varepsilon, \dots, d_n^{\theta-2j} = 1 - \varepsilon)$ . В узле  $x_k$  функция выигрыша игрока  $i$  будет иметь вид

$$V_i^k = \sum_{q \in L_{\theta-1}} p_{jq}^{\theta-2} (u_1^k, \dots, u_n^k) (V_i^q - u_i^k) \quad (3.5)$$

где  $V_i^q$  определяются по формуле (3.2) и (3.4), т.е. выигрыш игрока  $i$  в ситуации равновесия по Нэшу в узле  $x_j$  входит в формулу (3.5) в качестве  $V_i^q$ , т.е.  $V_i^q = V_i^{j*}$ , где  $q = k$ . Решаем игру  $\Gamma^k$  и получаем ситуацию равновесия по Нэшу и выигрыши в ситуации равновесия  $V_1^{k*}, \dots, V_n^{k*}$ . Аналогично рассматриваем все узлы графа  $G$ , которые принадлежат информационному



множеству  $L_{\theta-2}$ , затем переходим к информационному множеству  $L_{\theta-3}$  и т.д. В итоге приходим в вершину  $x_0 \in L_0$  и решая игру  $\Gamma^0$  получаем ситуацию абсолютного равновесия по Нэшу  $(u_1^*, \dots, u_n^*)$  в игре  $\tilde{\Gamma}$  и выигрыши игроков в ситуации абсолютного равновесия  $V_1^{0*}, \dots, V_n^{0*}$ .

### 3.3. Решение числового примера стохастической многошаговой игры

Рассмотрим пример стохастической многошаговой игры между двумя игроками на графе  $G$ , при условии, что  $\varepsilon = 0.5$

Положим, что у игроков есть две стратегии: низкие и высокие инвестиции — это  $\bar{s}$  и  $\underline{s}$  соответственно, причем  $V \gg \bar{s}$ . Следовательно множества стратегий 1 и 2-ого игрока:  $U_1^j = (\bar{s}, \underline{s})$ ,  $U_2^j = (\bar{y}, \underline{y})$ . Вероятности перехода из вершины  $x_0$ , которая принадлежит  $L_0$  в вершины  $x_1, x_2, x_3 \in L_1$ , в нашем случае, равны:

$$p_{01}^0 = \frac{p(u_2^0)(1 - p(u_1^0))}{p(u_1^0) + p(u_2^0) - p(u_2^0)p(u_1^0)},$$

где  $p(u_i^j)$  — вероятность что состояние  $i$ -ого игрока увеличится на  $\varepsilon$ . Здесь есть вероятность  $1 - p(u_1^0)$ , поскольку первый игрок не изменил своего состояния при переходе из вершины  $x_0$  в  $x_1$ :

$$p_{02}^0 = \frac{(1 - p(u_2^0))p(u_1^0)}{p(u_1^0) + p(u_2^0) - p(u_2^0)p(u_1^0)}.$$

$$p_{03}^0 = \frac{p(u_2^0) p(u_1^0)}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)}.$$

Аналогично вычисляются вероятности перехода и из других вершин. На рис.3.1 изображен граф  $G$ , где в каждом  $j$ -ом узле ( $j = 1, 2, \dots$ ) представлены  $(m_1^{cj}, m_2^{cj})$ .

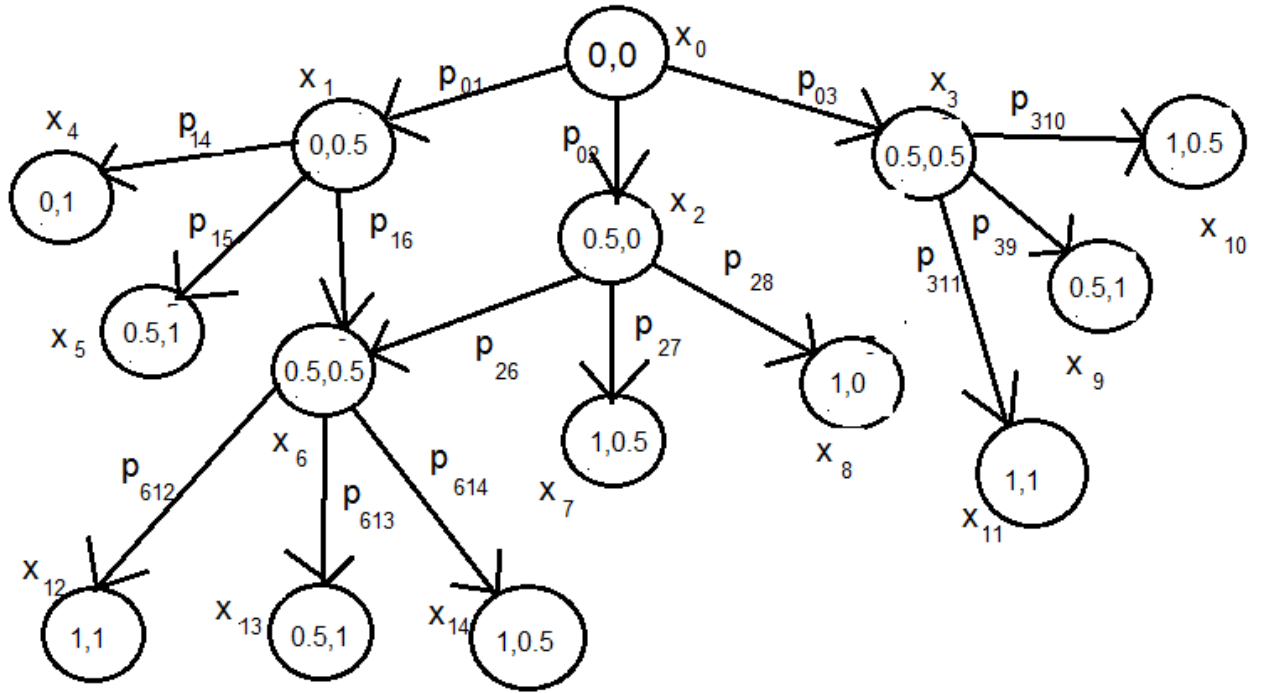


Рис. 2.4: Граф  $G$

Здесь, максимальное количество  $\theta = 4$ , следовательно у нас четыре информационных множества. Рассмотрим функции выигрыша 1 и 2 игрока во всех узлах графа  $G$ , заметим, что игровые элементы в узле  $x_6$  и  $x_3$  совпадают.

В узле  $x_6$  функции выигрыша равны:

$$V_1^6 = \frac{p(u_1^6)(1 - p(u_2^6))}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}(V - u_1^6) + \\ + \frac{p(u_2^6)(1 - p(u_1^6))}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}(-u_1^6) + \\ + \frac{p(u_1^6)p(u_2^6)}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}\left(\frac{V}{2} - u_1^6\right),$$

$$V_2^6 = \frac{p(u_1^6)(1 - p(u_2^6))}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}(-u_2^6) + \\ + \frac{p(u_2^6)(1 - p(u_1^6))}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}(V - u_2^6) + \\ + \frac{p(u_1^6)p(u_2^6)}{p(u_1^6) + p(u_2^6) - p(u_2^6)p(u_1^6)}\left(\frac{V}{2} - u_2^6\right),$$

В узле  $x_2$ :

$$V_1^2 = \frac{p(u_1^2)(1 - p(u_2^2))}{p(u_1^2) + p(u_2^2) - p(u_2^2)p(u_1^2)}(V - u_1^1) + \\ + \frac{p(u_2^2)(1 - p(u_1^2))}{p(u_1^2) + p(u_2^2) - p(u_2^2)p(u_1^2)}(V_1^6 - u_1^2) + \\ + \frac{p(u_1^2)p(u_2^2)}{p(u_1^2) + p(u_2^2) - p(u_2^2)p(u_1^2)}(V - u_1^2),$$

$$\begin{aligned}
V_2^2 = & \frac{p(u_1^2) (1 - p(u_2^2))}{p(u_1^2) + p(u_2^2) - p(u_2^2) p(u_1^2)} (-u_2^2) + \\
& + \frac{p(u_2^2) (1 - p(u_1^2))}{p(u_1^2) + p(u_2^2) - p(u_2^2) p(u_1^2)} (V_2^6 - u_2^2) + \\
& + \frac{p(u_1^2) p(u_2^2)}{p(u_1^2) + p(u_2^2) - p(u_2^2) p(u_1^2)} (-u_2^2),
\end{aligned}$$

Рассмотрим функции выигрыша игроков в узле  $x_1$ :

$$\begin{aligned}
V_1^1 = & \frac{p(u_1^1) (1 - p(u_2^1))}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (V_1^6 - u_1^1) + \\
& + \frac{p(u_2^1) (1 - p(u_1^1))}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (-u_1^1) + \\
& + \frac{p(u_1^1) p(u_2^1)}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (-u_1^1),
\end{aligned}$$

$$\begin{aligned}
V_2^1 = & \frac{p(u_1^1) (1 - p(u_2^1))}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (V_2^6 - u_2^1) + \\
& + \frac{p(u_2^1) (1 - p(u_1^1))}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (V - u_2^1) + \\
& + \frac{p(u_1^1) p(u_2^1)}{p(u_1^1) + p(u_2^1) - p(u_2^1) p(u_1^1)} (V - u_2^1),
\end{aligned}$$

В узле  $x_0$ :

$$\begin{aligned}
V_1^0 = & \frac{p(u_1^0) (1 - p(u_2^0))}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_1^2 - u_1^0) + \\
& + \frac{p(u_2^0) (1 - p(u_1^0))}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_1^1 - u_1^1) + \\
& + \frac{p(u_1^0) p(u_2^0)}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_1^3 - u_1^0),
\end{aligned}$$

$$\begin{aligned}
V_2^0 = & \frac{p(u_1^0) (1 - p(u_2^0))}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_2^2 - u_1^0) + \\
& + \frac{p(u_2^0) (1 - p(u_1^0))}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_2^1 - u_1^1) + \\
& + \frac{p(u_1^0) p(u_2^0)}{p(u_1^0) + p(u_2^0) - p(u_2^0) p(u_1^0)} (V_2^3 - u_1^0),
\end{aligned}$$

Задав, числовые значения параметрам модели, найдем численное решение многошаговой игры между двумя игроками по построенному алгоритму и при помощи алгоритма Лемке-Хаусона [20], который будем использовать при нахождении равновесия по Нэшу в игровых элементах  $\Gamma^j$  (см. Приложение 3). Зададим  $\varepsilon = 0.5$ :  $p(\bar{s}) = p(\bar{y}) = 0.9$ ,  $p(\underline{s}) = p(\underline{y}) = 0.25$ ,  $V = 0.9$ ,  $\bar{s} = \bar{y} = 0.8$ ,  $\underline{s} = \underline{y} = 0.2$ . Равновесие по Нэшу будет иметь вид:  $u_1^* = (\bar{s}, \underline{s}, \underline{s}, \underline{s}, \underline{s})$ ,  $u_2^* = (\bar{y}, \underline{y}, \underline{y}, \underline{y}, \underline{y})$ , а выигрыши в ситуации абсолютного равновесия по Нэшу равны:  $V_1 = 0.25$ ,  $V_2 = 0.25$

## Выводы

На данный момент существует множество различных моделей патентных гонок, начиная от простых моделей, которые не учитывают ранее накопленный опыт в области ИР и до моделей со сложной структурой, предполагающей несколько стадий развития патентных гонок. В последних моделях учитывается и опыт и ограниченность ресурсов фирм-участников патентных гонок и другие факторы, влияющие на состояние фирм. На основании проанализированных в данной работе моделей патентных гонок была построена новая, отличная от существующих модель патентной гонки. Данная модель представляет из себя стохастическую многошаговую игру, в которой учитывается опыт, приобретенный ранее фирмами-участниками. Однако, построенную модель можно развить, добавив ограничение на ресурсы фирм. Также интересно проанализировать построенную модель без ограничения на состояния игроков, т.е. что при переходе на следующий шаг, игроки могут остаться в том же состоянии, в котором были. Можно было бы попробовать ввести ставку дисконтирования в построенную модель.

## Заключение

В работе исследованы несколько моделей патентных гонок. Первая — патентная гонка без памяти (пуассоновского типа). Для нее было доказано, что в явном виде невозможно найти равновесие по Нэшу. Проведен анализ модели, разработан и программно реализован алгоритм численного решения задачи нахождения равновесия по Нэшу. Для базового примера проведен анализ чувствительности равновесия в зависимости от параметров модели. Вторая модель патентной гонки представляет собой стохастическую дифференциальную игру. Для данной модели был проведен анализ чувствительности равновесия в зависимости от параметров модели. На основании рассмотренных моделей была построена стохастическая многошаговая игра, моделирующая патентные гонки. Для нее был разработан и программно реализован алгоритм нахождения равновесия по Нэшу, также рассчитан базовый пример.

## Список литературы

1. Tirole J. The Theory of Industrial Organization. London: The MIT Press, 1990. P. 619–623.
2. Agnion P., Howwit P. A model of growth through creative destruction // *Econometrica*, 1992. Vol. 60, No 2. P. 323–351.
3. Lee T., Wilde L. Market structure and innovation: a reformulation // *The Quart. J. Econ*, 1980. Vol. 194. P. 429–436.
4. Reinganum J. F. A dynamic game of R and D: patent protection and competitive behavior // *Econometrica*, 1982. Vol. 50, No 3. P. 71–688.
5. Dasguspta P., Stiglitz J. Uncertainty, industrial structure and the speed of reseaching and development // *Bell J. Econom*, 1980. No 11. P. 1–28.
6. Sennewald K. Controlled stochastic differential equations under poisson uncertainty and with unbounded utility // *J. Econ. Dynam. Control*, 2007. No 31. P. 1106–1131.
7. Gayle P. G. Market structure and product innovation // Boulder (Colorado) / University of Colorado, 2001. P. 1–15.



8. Sutton J. Technology and Market Structure. London: The MIT Press, 2000. P 692.
9. Шевкопляс Е. В, Костюнин С. Ю Об упрощении интегрального выигрыша в дифференциальных играх со случайной продолжительностью // Вестник Санкт-Петербургского университета. Серия 10: Прикладная математика. Информатика. Процессы управления. 2011. е 4. С. 47–56
10. Dockner E. J. Differential Games in Economics and Management Science // Cambridge / Cambridge University Press, 2000. P. 721.
11. Reinganum J. F. A dynamic game of R and D: patent protection and competitive behavior // Econometrica, 1982. Vol. 50, No 3. P. 71–688.
12. Mehlmann A. Applied Differential Games. New York: Plenum Press, 1988. P 201.
13. Keller A. A. Stochastic differential games and queueing models to innovation and patenting // Contributions to game theory and management / Ed. by L. A. Petrosjan and N. A. Zenkevich. St. Petersburg: Graduated School of Management St. Petersburg State University, 2007. P. 245–269.

14. Doraszleski U. Rent Dissipation in R and D Races // The Economics of Innovation, 2008. No 286. P. 3-13
15. Reinganum J.F. "The timing of Innovation: Research, Development, and Diffusion" in R // Schmalensee and R.D. Willig: Handbook of Industrial Organization(I). Amsterdam, 1989.
16. Fudenberg D., Gilbert R., Stiglitz J., Tirole J. Preemption, leapfrogging and competition in patent races. // European Economic Review, 1983. No 22. P. 3-31.
17. Петросян Л. А., Зенкевич Н. А., Шевкопляс Е. В. Теория игр. СПб.: БХВ-Петербург, 2012. 432 с.
18. Петросян Л. А., Седаков А. А. Многошаговые сетевые игры с полной информацией. // Управление большими системами. 2009. № 26-1. С. 121–132.
19. Волков И. К., Зуев С. М., Цветкова Г. М. Случайные процессы: Учеб. для вузов / Под ред. Зарубина В. С., Крищенко А. П.-М:Изд-во МГТУ им. Баумана Н. Э., 1999. 448 с.
20. Петросян Л. А., Зенкевич Н. А., Янг Д. В. К. Динамические игры и их приложения в менеджменте. СПб: Изд-во "Высшая школа менеджмента 2009. 415 с.

21. Набатова Д. С. О формировании процесса обучения решению биматричных игр с помощью алгоритма Лемке-Хоусона // Труды МАИ, 2009. № 35. С. 1-14.

# Приложение 1

```
x=fsolve(@t1,[1 1])
[k1 k2]=t1(x)
l=[1 2 3 4 5 6 7 8 9 10 ];
w1=[3.9968*10^(-15) ];
w2=[4.6947 ];
function [ y ] = h(x)
b=1;
y = b*x^(1/2);
end
function [ V1,V2 ] = t1(x)
r1=9/10;
b=11*(1/r1);
c=4*(1/r1);
d=6*(1/r1);
a=5;
s=1;
k=2;
V1=(r1*(s*(1/k))/(x(1))^(1/k)+((s*(1/k))/(x(1))^(1/k)*h1(x(2)))*b-s*(1/k)/(x(1))^(1/k)*a-r1-
-h1(x(1))-h1(x(2))-h1(x(2))*c+x(1)*s*(1/k)/(x(1))^(1/k);
V2=(r1*s*(1/k)/(x(2))^(1/k)+s*(1/k)/(x(2))^(1/k)*s*(x(1))^(1/k)*d-r1-
-s*(x(1))^(1/k)-s*(x(2))^(1/k)+x(2)*s*(1/k)/(x(2))^(1/k);
end
```

## Приложение 2

```
p_i= 50;
p_f=2;
N=5;
r=0.1;
T=1;
lambda=7;
t=0;
u=(2*lambda*p_i*(p_i-p_f)*(N-1)*exp(r*t))/
((2*N-1)*p_i-(p_i+2*(N-1)*p_f)*exp(((p_i-p_f)*(N-1)*lambda^2*(exp(r*t)-exp(r*T)))/r))
s=zeros(1,10);
t=zeros(1,10);
for i=1:10
    r1=i*0.1;
    t(1,i)=r1;
    s(1,i)=(2*lambda*p_i*(p_i-p_f)*(N-1)*exp(r*t(1,i)))/
    ((2*N-1)*p_i-(p_i+2*(N-1)*p_f)*exp(((p_i-p_f)*(N-1)*lambda^2*(exp(r*t(1,i))-exp(r*T)))/r));
end
s
plot(t,s,'r')
```

# Приложение 3

```
p1=0.9;
p2=0.25;
V=0.9;
x1=0.8;
x2=0.2;
r11=(1/(2-p1))*(p1*((V/2)-x1)+(1-p1)*(V-2*x1));
q11=(1/(2-p1))*(p1*((V/2)-x1)+(1-p1)*(V-2*x1));
r12=(1/(p1+p2-p1*p2))*(-(1-p1)*p2*(x1)+p1*(1-p2)*(V-x1)+p1*p2*((V/2)-x1));
q21=(1/(p1+p2-p1*p2))*((1-p2)*p1*(-x2)+p2*(1-p1)*(V-x2)+p1*p2*((V/2)-x2));
r22=(1/(2-p2))*(p2*((V/2)-x2)+(1-p2)*(V-2*x2));
q22=(1/(2-p2))*(p2*((V/2)-x2)+(1-p2)*(V-2*x2));
r21=q21;
q12=r12;

A=[r11 r12; r21 r22]
B=[q11 q12; q21 q22]

a1=0;
b1=0;

a=LemkeHowson(A,B)
b=a{2}
a=a{1}
m=r22;
k=q22;
m11=((1-p1)/(2-p1))*(-x1)+((1-p1)/(2-p1))*m+p1*(-x1)/(2-p1);
n11=((1-p1)/(2-p1))*(V-x1)+((1-p1)/(2-p1))*k+p1*(V-x1)/(2-p1);
m12=((1-p1)*p2)/(p1+p2-p1*p2))*(-x1)+(1-p2)*p1*m/(p1+p2-p1*p2)+p1*p2*(-x1)/(p1+p2-p1*p2);
n12=((1-p1)*p2)/(p1+p2-p1*p2))*(V-x2)+(1-p2)*p1*k/(p1+p2-p1*p2)+p1*p2*(V-x2)/(p1+p2-p1*p2);
m21=((1-p2)*p1)/(p1+p2-p1*p2))*(-x2)+(1-p1)*p2*m/(p1+p2-p1*p2)+p1*p2*(-x2)/(p1+p2-p1*p2);
n21=((1-p2)*p1)/(p1+p2-p1*p2))*(V-x1)+(1-p1)*p2*k/(p1+p2-p1*p2)+p1*p2*(V-x1)/(p1+p2-p1*p2);
m22=((1-p2)/(2-p2))*(-x2)+((1-p2)/(2-p2))*m+p2*(-x2)/(2-p2);
n22=((1-p2)/(2-p2))*(V-x2)+((1-p2)/(2-p2))*k+p2*(V-x2)/(2-p2);
A=[m11 m12; m21 m22];
B=[n11 n12; n21 n22];
a=LemkeHowson(A,B)
b1=a{2}
a1=a{1}
m1=m22;
k1=n22;
w11=((1-p1)/(2-p1))*m+((1-p1)/(2-p1))*(V-x1)+(p1/(2-p1))*(V-x1);
s11=((1-p1)/(2-p1))*k+((1-p1)/(2-p1))*(-x1)+(p1/(2-p1))*(-x1);
w12=((1-p1)*p2/(p1+p2-p1*p2))*m+((1-p2)*p1/(p1+p2-p1*p2))*(V-x1)+(p1*p2/(p1+p2-p1*p2))*(V-x1);
s12=((1-p1)*p2/(p1+p2-p1*p2))*k+((1-p2)*p1/(p1+p2-p1*p2))*(-x2)+(p1*p2/(p1+p2-p1*p2))*(-x2);
w21=((1-p2)*p1/(p1+p2-p1*p2))*m+((1-p1)*p2/(p1+p2-p1*p2))*(V-x2)+(p2*p1/(p1+p2-p1*p2))*(V-x2);
s21=((1-p2)*p1/(p1+p2-p1*p2))*k+((1-p1)*p2/(p1+p2-p1*p2))*(-x1)+(p2*p1/(p1+p2-p1*p2))*(-x1);
w22=((1-p2)/(2-p2))*m+((1-p2)/(2-p2))*(V-x2)+(p2/(2-p2))*(V-x2);
s22=((1-p2)/(2-p2))*k+((1-p2)/(2-p2))*(-x2)+(p2/(2-p2))*(-x2);
A=[w11 w12; w21 w22];
```

```

B=[s11 s12; s21 s22];

a=LemkeHowson(A,B)
b2=a{2}
a2=a{1}
m2=w22;
k2=s22;

p11=((1-p1)/(2-p1))*m1+((1-p1)/(2-p1))*m2+(p1/(2-p1))*m;
d11=((1-p1)/(2-p1))*k1+((1-p1)/(2-p1))*k2+(p1/(2-p1))*k;
p12=((1-p1)*p2/(p1+p2-p1*p2))*m1+((1-p2)*p1/(p1+p2-p1*p2))*m2+(p1*p2/(p1+p2-p1*p2))*m;
d12=((1-p1)*p2/(p1+p2-p1*p2))*k1+((1-p2)*p1/(p1+p2-p1*p2))*k2+(p1*p2/(p1+p2-p1*p2))*k;
p21=((1-p2)*p1/(p1+p2-p1*p2))*m1+((1-p1)*p2/(p1+p2-p1*p2))*m2+(p2*p1/(p1+p2-p1*p2))*m;
d21=((1-p2)*p1/(p1+p2-p1*p2))*k1+((1-p1)*p2/(p1+p2-p1*p2))*k2+(p2*p1/(p1+p2-p1*p2))*k;
p22=((1-p2)/(2-p2))*m1+((1-p2)/(2-p2))*m2+(p2/(2-p2))*m;
d22=((1-p2)/(2-p2))*k1+((1-p2)/(2-p2))*k2+(p2/(2-p2))*k;
A=[p11 p12; p21 p22]
B=[d11 d12; d21 d22]

a=LemkeHowson(A,B)
b0=a{2}
a0=a{1}
m0=p11
k0=d11
function nashEqbm = LemkeHowson(varargin)

if nargin < 2 || nargin > 4
    error('This function takes between two and four arguments');
end

A = varargin{1};
B = varargin{2};

if any(size(A) ~= size(B))
error('Matrices must have same dimension');
end

[m,n] = size(A);
size_ = [m,n];

if nargin > 2
    k0 = varargin{3};
    if k0 < 1 || k0 > m+n
        error(['Initial pivot must be in {1,...,' num2str(n+m) '}']);
    end
else
    k0 = 1;
end

if nargin == 4

```

```

maxPivots = varargin{4};
if maxPivots < 1
    error('Maximum pivots parameter must be a positive integer!');
end
else
maxPivots = 500000;
end

minVal = min( min(min(A)), min(min(B)) );
if minVal <= 0
    A = A + ones(size(A))*(1-minVal);
    B = B + ones(size(A))*(1-minVal);
end

Tab = cell(2,1);
Tab{1} = [B', eye(n), ones(n,1)];
Tab{2} = [eye(m), A, ones(m,1)];

rowLabels = cell(2,1);
rowLabels{1} = m+1:m+n;
rowLabels{2} = 1:m;

k = k0;
if k0 <= m
    player = 1;
else
    player = 2;
end

numPiv = 0;
while numPiv < maxPivots

    numPiv = numPiv+1;

    LP = Tab{player};
    [m_, ~] = size(LP);

    max_ = 0;
    ind = -1;
    for i = 1:m_
        t = LP(i,k) / LP(i, m+n+1);
        if t > max_
            ind = i;
            max_ = t;
        end
    end

    if max_ > 0
        Tab{player} = pivot(LP, ind, k);
    else
        break;
    end
end

```



```

end

temp = rowLabels{player}(ind);
rowLabels{player}(ind) = k;
k = temp;

if k == k0
    break;
end
if player == 1
    player = 2;
else
    player = 1;
end

end

if numPiv == maxPivots
    error(['Maximum pivot steps (' num2str(maxPivots) ') reached!']);
end

nashEqbm = cell(2,1);

for player = 1:2

    x = zeros(size_(player), 1);
    rows = rowLabels{player};
    LP = Tab{player};

    for i = 1:length(rows)
        if player == 1 && rows(i) <= size_(1)
            x(rows(i)) = LP(i,m+n+1) / LP(i,rows(i));
        elseif player == 2 && rows(i) > size_(1);
            x(rows(i)-size_(1)) = LP(i,m+n+1) / LP(i,rows(i));
        end
    end

    nashEqbm{player} = x/sum(x);

end

end

function B = pivot(A,r,s)

[m,~] = size(A);
B = A;

for i = 1 : m
    if i == r
        continue;

```

```
    else
        B(i,:) = A(i,:) - A(i,s) / A(r,s) * A(r,:);
    end
end
end

end
```