

Министерство Образования Украины

Учреждение образования

Одеський національний університет імені І. І. Мечникова

Институт Інститут математики, економіки та механіки

Факультет Прикладная математика

Одесса 2016

СОДЕРЖАНИЕ

Вступление	3
Раздел 1 Основные понятия и определения	4
1.1 Элементы теории игр	4
1.2 Элементы анализа на временных шкалах	4
Раздел 2 Макроэкономические модели на временных шкалах	6
2.1 Классическая модель Барро-Гордона	6
2.2 Теоретико-игровая интерпретация модели Барро-Гордона	10
2.3 Устойчивость равновесных стратегий в модели Барро-Гордона на временных шкалах	12
2.4 Классическая модель профсоюз—монополист	16
2.5 Модель профсоюз—монополист на временных шкалах	17
Раздел 3 Компьютерная имитация макроэкономических моделей	20
3.1 Анализ устойчивости стратегий в модели Барро-Гордона	20
3.2 Анализ устойчивости стратегий в модели профсоюз—монополист	24
Выводы	29
Список использованных источников	30
Додаток А	31

ВСТУПЛЕНИЕ

Актуальность теории игры заключается в широком спектре применения: в экономике, социальных наук, биологии, компьютерных науках. Тем более важно изучение теории игр на временных шкалах, так как модели построенные на временных шкалах являются особенно приближенные к реальной жизни.

В данной дипломной работе будут:

- рассмотрена игровая интерпретация модели Барро-Гордона на временных шкалах
- построена игровая интерпретация модели "профсоюз—монополист"
- расширена модель "профсоюз—монополист" на временных шкалах
- написано программное обеспечение имитирующее игру на временных шкалах

РАЗДЕЛ 1 ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

1.1 Элементы теории игр

Определение 1. (S, H) — не кооперативная игра n лиц в нормальной форме, где S — набор чистых стратегий, а H — набор выигрышей. Когда каждый игрок $i \in \{1, \dots, n\}$ выбирает стратегию $x_i \in S$ в профиле стратегий $x = (x_1, \dots, x_n)$, игрок i получает выигрыш $H_i(x)$. Профиль стратегий $x^* \in S$ является равновесием по Нэшу, если изменение своей стратегии с x_i^* на x_i не выгодно ни одному игроку i , то есть $\forall i : H_i(x^*) \geq H_i(x_i, x_{-i}^*)$.

Определение 2. Равновесие по Нэшу называется безупречным по под-играм, если и только если оно является равновесием по Нэшу для каждой под-игры.

Определение 3. Парето-оптимальность в смысле теории игр характеризует такую ситуацию, при которой невозможно улучшить исход игры для одного игрока без его ухудшения для других игроков.

Определение 4. Инфляция — повышение общего уровня цен на товары и услуги.

Определение 5. Индексация заработной платы — повышение заработных плат с целью частичной защиты населения от роста потребительских цен на товары и услуги.

Определение 6. Любое совершенное равновесие по под-играм (SPNE), в котором оба игрока выбирают стратегию L во всех своих ходах, назовём **совершенным равновесием Рамсея по под-играм (Ramsey SPNE)**

1.2 Элементы анализа на временных шкалах

Определение 7. Под временной шкалой понимается непустое замкнутое подмножество множества вещественных чисел, она обозначается символом \mathbb{T} . Свойства временной шкалы определяются тремя функциями:

1) оператор перехода вперед:

$$\sigma(t) = \inf \{s \in \mathbb{T} : s > t\};$$

2) оператор перехода назад:

$$\rho(t) = \sup \{s \in \mathbb{T} : s < t\},$$

(при этом полагается $\inf \emptyset = \sup \mathbb{T}$ и $\sup \emptyset = \inf \mathbb{T}$);

3) функция зернистости

$$\mu(t) = \sigma(t) - t.$$

Поведение операторов перехода вперед и назад в конкретной точке временной шкалы определяет тип этой точки. Соответствующая классификация точек представлена в таблице (1.1).

t справа рассеянная	$t < \sigma(t)$
t справа плотная	$t = \sigma(t)$
t слева рассеянная	$\rho(t) < t$
t слева плотная	$\rho(t) = t$
t изолированная	$\rho(t) < t < \sigma(t)$
t плотная	$\rho(t) = t = \sigma(t)$

Таблица 1.1 – Классификация точек временной шкалы

Определение 8. Множество \mathbb{T}^κ определим как:

$$\mathbb{T}^\kappa = \begin{cases} \mathbb{T} \setminus \{M\}, & \text{если } \exists \text{ справа рассеянная точка } M \in \mathbb{T} : \\ & M = \sup \mathbb{T}, \sup \mathbb{T} < \infty \\ \mathbb{T}, & \text{в противном случае.} \end{cases}$$

Далее полагаем $[a, b] = \{t \in \mathbb{T} : a \leq t \leq b\}$.

Определение 9. δ -разбиение отрезка временной шкалы $[t_0, t_f]$ определим следующим образом: задавшись диаметром разбиения δ , положим первую точку разбиения равной t_0 , а дальнейшие точки определим по формуле

$$t_i = \begin{cases} \sup(t_{i-1}, t_{i-1} + \delta], & \text{если } t_{i-1} + \delta \in \mathbb{T}^\kappa, \\ \sigma(t_{i-1}), & \text{если } t_{i-1} + \delta \notin \mathbb{T}^\kappa. \end{cases}$$

РАЗДЕЛ 2 МАКРОЭКОНОМИЧЕСКИЕ МОДЕЛИ НА ВРЕМЕННЫХ ШКАЛАХ

2.1 Классическая модель Барро-Гордона

Классическая модель Барро-Гордона рассматривает правительство (в лице некоторого одного политика), которое принимает решение о мерах воздействия на экономику основываясь на двух факторах. Первый фактор является собой суммарную функцию благосостояния вида

$$W = \sum_{t=0}^{\infty} \beta^t w_t = \sum_{t=0}^{\infty} \beta^t \left(y_t - \frac{\lambda \pi^2}{2} \right), \quad 0 < \beta < 1, \quad (2.1)$$

где β^t - значимость t -ого периода для благосостояния политика, w_t - функция благосостояния политика в t -ом периоде. При этом взаимосвязь между инфляцией и безработицей задается кривой Лукаса

$$y = y^* + \alpha(\pi - \pi^\beta). \quad (2.2)$$

Второй фактор принятия решения – это оценка общественностью действий правительства в виде поддержки или недоверия. Такая оценка в модели задается через инфляционные ожидания, которые формируются в период t следующим образом:

$$\pi_t = \begin{cases} \pi, & \forall t : \pi_t = \tilde{\pi}_t, \\ \frac{\alpha}{\lambda}, & \exists s < t : \pi_s \neq \tilde{\pi}_s, \end{cases} \quad (2.3)$$

где $\tilde{\pi}$ - уровень инфляции, который политики обещают достичь в результате воздействия на экономику.

Как следует из (2.3), если политик ведет себя честно, то общественность ему доверяет и ожидает обещанный им уровень инфляции. Если политик хотя бы раз, обманув ожидания, провел не ту политику, которую обещал, то общественность ему не верит. В таком случае она ожидает отличный от обещанного уровень инфляции, который составляет $\frac{\alpha}{\lambda}$, поскольку полагает, что в случае обмана политик будет стремиться к тому, чтобы в экономике установился именно этот уровень инфляции.

Действительно, в случае обмана политик выберет такой уровень инфляции, который будет максимизировать его функцию благосостояния в этом

периоде (допустим, в периоде s). Поскольку, согласно (2.1)-(2.3), функция благосостояния периода s при $\pi_s \neq \tilde{\pi}$ имеет вид

$$w_s = y^* \alpha \left(\pi_s - \frac{\lambda \pi_s^2}{2} \right), \quad (2.4)$$

то максимизирующий её уровень инфляции будет равен величине $\frac{\alpha}{\lambda}$, что следует из первого условия максимизации.

Так как репутация политика влияет на инфляционные ожидания общественности, то, согласно (2.1)-(2.2), репутация влияет также и на его общую функцию благосостояния. Предположим, что политик ведет себя честно, то есть проводит ту политику, которую обещал. Тогда его общая функция благосостояния принимает вид

$$w^{fair} = \frac{1}{(1-\beta)} \left(y^* - \frac{\lambda \pi^2}{2} \right), \quad (2.5)$$

поскольку в этом случае, согласно (2.1)-(2.3), функция благосостояния для любого периода t имеет вид

$$w_t^{fair} = y^* - \frac{\lambda \pi^2}{2}. \quad (2.6)$$

Если политик решит обмануть ожидания общественности в какой-то (допустим в нулевой) период времени, то в этом случае, согласно (2.1)-(2.3), его функция благосостояния в нулевой период составит

$$w_0^{deception} = y^* - \frac{\alpha^2}{2\lambda} - \alpha \tilde{\pi}, \quad (2.7)$$

а во все последующие периоды вследствие недоверия населения общественности

$$w_0^{deception} = y^* - \frac{\alpha^2}{2\lambda}. \quad (2.8)$$

Следовательно, общая функция благосостояния политика в случае нарушения данных обещаний имеет вид

$$W^{deception} = \frac{1}{(1-\beta)} y^* + \frac{(1-2\beta)\alpha^2}{(1-\beta)2\lambda} - \alpha \tilde{\pi}, \quad (2.9)$$

Очевидно, что политику имеет смысл вести себя честно только при условии $W^{fair} > W^{deception}$, в противном случае он выигрывает больше от невыполнения данных ранее обещаний.

Выведем условия, при которых политик ведет себя честно, не обманывая ожидания населения. Воспользуемся для этого графической интерпретацией этой проблемы: построим графики выигрыша и проигрыша политика в случае обмана и найдем области, в которых выигрыш будет больше проигрыша и наоборот.

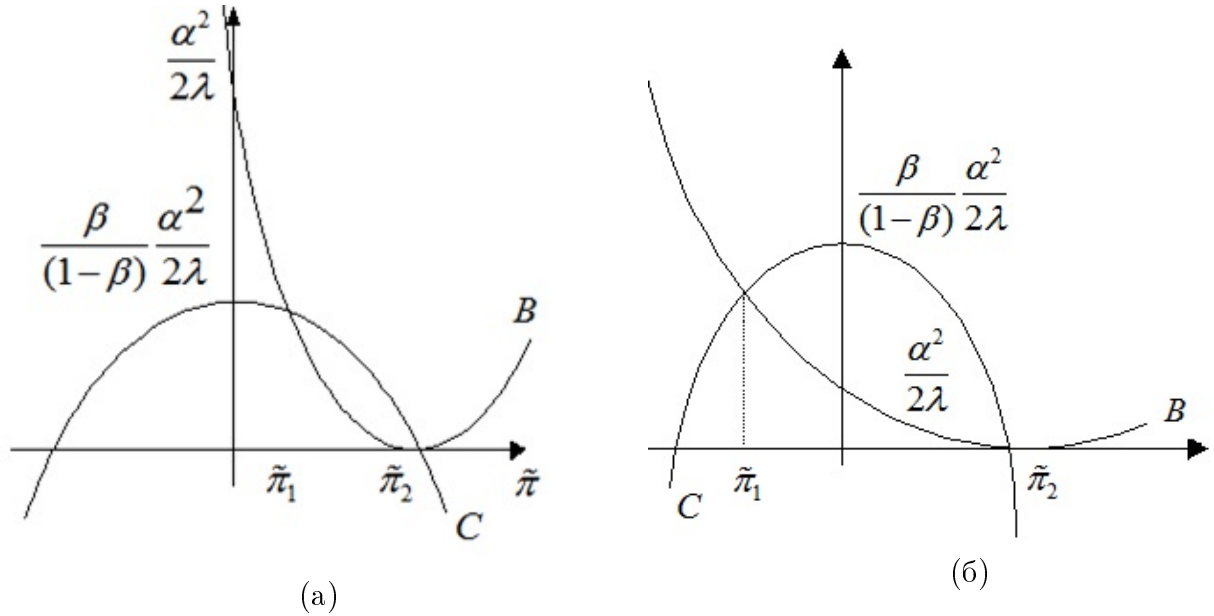


Рис 2.1

Если политик обманывает ожидание общественности в нулевой момент времени, то выигрыш он получает за счет того, что в нулевой момент времени обманул ожидания общественности, а проигрыш – что в дальнейшем общественность перестает ему доверять и всегда ожидает больший уровень инфляции, чем было объявлено.

Следовательно, выигрыш обманывающего население в нулевой момент времени политика

$$B = w_0^{deception} - w_0^{fair} = \frac{\lambda \pi^2}{2} + \frac{\alpha^2}{2\lambda} - \alpha \tilde{\pi} \quad (2.10)$$

является квадратичной функцией от объявляемого уровня инфляции. Эта функция достигает минимума $B = 0$ в точке $\tilde{\pi} = \frac{\alpha}{\lambda}$. При $\tilde{\pi} = 0$ выигрыш равен $\frac{\alpha^2}{\lambda}$.

Проигрыш обманывающего население в нулевой момент времени политика также является квадратичной функцией от объявляемого уровня инфляции, поскольку

$$C = \sum_{t=1}^{\infty} \beta^t (w_t^{deception} - w_t^{fair}) = -\frac{\lambda \tilde{\pi}^2}{2} + \frac{\alpha^2}{2\lambda}. \quad (2.11)$$

Функция, задаваемая выражением (2.11), описывает параболу, достигающую максимума $C = \frac{\beta}{(1-\beta)} \frac{\alpha^2}{2\lambda}$ в точке $\tilde{\pi} = 0$. Значение функции равняется нулю, когда уровень инфляции составляет $\tilde{\pi} \pm \frac{\alpha}{\lambda}$.

Взаиморасположение графиков потерь и выигрыша обманывающего общественность в нулевой момент времени политика зависит от того, больше или меньше единицы величина $\frac{\beta}{1-\beta}$.

Если $\frac{\beta}{1-\beta} < 1$, то есть $\beta < \frac{1}{2}$, то функция потерь и функция выигрыша пересекаются в точках $\tilde{\pi}_1 > 0$ и $\tilde{\pi}_2 > 0$ (2.1а), причем $\tilde{\pi}_1 = \frac{\alpha(1-2\beta)}{\lambda}$, $\tilde{\pi}_2 = \frac{\alpha}{\lambda}$. Если $\frac{\beta}{1-\beta} > 1$, то есть $\beta > \frac{1}{2}$, то функция потерь и функция выигрыша пересекаются в точках $\tilde{\pi}_1 < 0$, $\tilde{\pi}_2 > 0$ (2.1б).

Таким образом, из проведенного анализа следует, что поведение политика (будет он выполнять обещания или нет) зависит от его отношения к репутации. Если для него репутация не важна, будущее мало влияет на его функцию благосостояния $\beta < \frac{1}{2}$, то политику не выгодно вести себя честно. Поскольку, как видно из (2.1а), существует достаточно невысокий уровень инфляции $\tilde{\pi} \in \left(0; \frac{\alpha(1-2\beta)}{\lambda}\right)$, обещая который он может выиграть больше в результате обмана (по сравнению с честным поведением). Если для политика его репутация важна, будущее значимо для его благосостояния $\beta < \frac{1}{2}$, то ему имеет смысл вести себя честно. Поскольку, как видно из (2.1б), не существует уровня инфляции близкого к нулю, обещая который он может получить большую реализацию своих целей в результате обмана по сравнению с честным поведением.

2.2 Теоретико-игровая интерпретация модели Барро-Гордона

В игре участвуют два игрока: p - правительство, q - общественность, которые оперируют инфляцией π и индексированием заработной платы ω соответственно. Для упрощения модели положим, что оба не могут прогнозировать будущее. Каждый игрок выбирает из следующих стратегий: низким L и высоким H уровнем повышения. В общем виде игра может быть задана в виде матрицы выигрышей,

		Общественность	
		L	H
Правительство	L	a, q	b, v
	H	c, x	d, z

где параметры a, b, c, d, q, v, x, z - выигрыши удовлетворяющие следующим ограничениям

$$c > a > d > b, q > v, q \geq z > x. \quad (2.12)$$

В данном случае проще всего описать экономику через функцию совокупного предложения Лукаса

$$y_t - Y = \lambda(\pi_t - \omega_t) + \varepsilon_t, \quad (2.13)$$

где $\lambda > 0$, y - производительность, Y - естественный уровень производительности, а ε - макроэкономический шок близкий к нулю. Коэффициенты дисконтирования игроков составляют β_g и β_p , а их функции полезности имеют следующий вид

$$u_t^g = -(\pi_t - \tilde{\pi})^2 + \alpha y_t - \beta(y_t - Y)^2, \quad (2.14)$$

$$u_t^p = -(\pi_t - \omega)^2, \quad (2.15)$$

где $\tilde{\pi}$ - оптимальный уровень инфляции, а $\alpha > 0, \beta > 0$ описывают относительный вес целей правительства (стабильной инфляции, высокой и стабильной производительности). Общественность озабочена верным ожиданием уровня инфляции для того, чтобы определить уровень зарплат на рынке.

Так как нас интересует эффект от выбранной политики, то сфокусируемся на долгосрочном исходе игры. Для этого однозначно определим экономику положив $\forall t, \varepsilon_t = 0$, что подразумевает, что мы можем положить

$\beta = 0$ без потери общности. Из этого следует, что инструмент правительства π представляет собой выбор средней инфляции.

В стандартной пошаговой игре, в которой игроки могут менять свое поведение в каждый период, мы используем (2.13)-(2.14) для получения равновесия

$$\pi_t^* = \tilde{\pi} + \frac{\alpha\lambda}{2} = \omega_t^*, \quad (2.16)$$

что является известным результатом $\pi_t^* > \tilde{\pi}$. Сфокусировав внимание на двух уровнях инфляции мы следуем [1], где были предложены два наиболее естественных варианта – оптимальный уровень из (2.14) и согласованного по времени из (2.16)

$$\pi \in \left\{ L = \tilde{\pi}, H = \tilde{\pi} + \frac{\alpha\lambda}{2} \right\} \ni \omega_t^*. \quad (2.17)$$

Мы можем, учитывая (2.13)-(2.15) и поделив на $\left(\frac{\alpha\lambda}{2}\right)$, без потери общности вывести соответствующие выигрыши, представленные в таблице ниже. Так же вне зависимости от λ и α справедливы следующие ограничения для данной игры в дополнение к изначальным (2.12)

$$c > a = 0 > d > b, c = -d = -\frac{b}{2}, q > v, q \geq z > x \quad (2.18)$$

$$c = 1 > a = 0 > d = -1 > b = -2, q = z = 0 > v = x = -1, \quad (2.19)$$

		Общественность	
		L	H
Правительство	L	0,0	-1, -1
	H	$\frac{1}{2}, -1$	$-\frac{1}{2}, 0$

Таблица 2.1

Стандартная пошаговая игра имеет уникальное равновесие по Нэшу (H, H) . Однако, оно неэффективно, так как является Парето доминированным. Это означает, что существует такой исход игры, который улучшит состояние одного, но при этом не ухудшит его для других игроков. В данном случае это «не Нэшовский» исход (L, L) . У правительства возникает соблазн создать неожиданную инфляцию, чтобы повысить производительность и снизить уровень безработицы. Так как общественность рациональна, то будет

ожидать высокую инфляцию – оба игрока будут в проигрыше.

2.3 Устойчивость равновесных стратегий в модели Барро-Гордона на временных шкалах

Рассмотрим игровую интерпретацию модели Барро-Гордона на временных шкалах.

Все допущения выдвинутые в стандартной игре остаются. Расширим их:

- игра начинается одновременным ходом,
- заранее известно неизменное количество ходов $r^g \in \mathbb{N}$ и $r^p \in \mathbb{N}$,
- игра заканчивается через T периодов, где T - наименьшее общее кратное для r^g и r^p ,
- игроки рациональны, обладают равноценными знаниями и полной информацией о структуре игры, матрице выигрышей и всех предыдущих ходах.

Другими словами определяется три временных шкалы: правительства, общественности и самой игры:

$$T_g = \{0, r^g, 2r^g, \dots, T\}, T_p = \{0, r^p, 2r^p, \dots, T\}, T = T_g \cup T_p \quad (2.20)$$

Главным преимуществом использования однородных временных шкал является экономическая интерпретация: r^g и r^p представляют собой степень возможности изменений политики относительно инфляции и степень реагирования для внесения изменений в заработные платы.

Асинхронная игра на временных шкалах будет как правило иметь несколько равновесий по Нэшу, среди которых мы выберем лучшую в зависимости от под-игры.

Теорема 1. *Рассмотрим общую несогласованную по времени игру на однородных временных шкалах, для которой выполняются (2.18) и (2.20). Тогда все SNPE игры будут SNPE Рамсея, если и только если*

$$r^g > \bar{r}^g(R) = \begin{cases} \frac{c-d}{a-d}r^p = \frac{a-b}{a-d}r^p, & \text{если } R = 0 \\ \frac{(1+R)(c-d)}{a-d}r^p = \frac{a-b+R(c-d)}{a-d}r^p, & \text{если } R \in (0; \bar{R}) \\ \frac{c-d-(1-R)(a-b)}{a-d}r^p = \frac{(a-b)}{a-d}Rr^p, & \text{если } R \in (\bar{R}; 1) \end{cases} \quad (2.21)$$

где $\bar{R} = \frac{q-v}{z-x+q-v}$. В несогласованной игре, где справедливо (2.18), (2.21) преобразуется в

$$\frac{r^g}{r^p} \in \left(\frac{3}{2}, 2\right) \cup \left(\frac{5}{2}, \infty\right) \quad (2.22)$$

Доказательство: смотреть Либиха и Штелиха [2].

Можно утверждать, что действия игроков могут не всегда быть детерминистическими и/или что частота их ходов различается во времени или подчиняется некоторому случайному процессу. Разнородные временные шкалы позволяют изучать подобные случаи. Для большей эффективности нормализуем горизонт планирования как $T = r^g$. Как и ранее игроки g и p ходят одновременно в первом периоде и во всех $T = r^g$ периодах, в промежутках между которыми общественность так же способна реагировать на ход правительства. Для удобства сравнения оставим все предположения предыдущего параграфа неизменными.

Рассмотрим произвольную временную шкалу \mathbb{T} и произвольную невозрастающую функцию реакции общественности $f : \mathbb{T} \rightarrow [0,1]$. В данной работе рассматриваются два представляющих интерес особых случая. В первом, при разнородной (атомистической) общественности, функция реакции может быть интерпретирована как часть общественности, которая уже имела возможность совершить ход. Во втором, при вероятностных ходах общественности, функция реакции может быть интерпретирована как кумулятивная функция распределения её ходов.

Чтобы удостовериться в том, что все SPNE являются SPNE Рамсея, достаточно показать, что в первом ходе оптимальной стратегией g является L вне зависимости от хода общественности в первом периоде. Получим два соответствующих условия:

$$ar^g > c \int_0^{r^g} 1 - f(t) \Delta t + d \int_0^{r^g} f(t) \Delta t, \quad (2.23)$$

$$dr^g > b \int_0^{r^g} 1 - f(t) \Delta t + a \int_0^{r^g} f(t) \Delta t. \quad (2.24)$$

Теорема 2. *Рассмотрим общую несогласованную по времени игру, в которой выполняется (2.18) и $f : \mathbb{T} \rightarrow [0,1]$ – невозрастающая функция реакции. Тогда все SPNE игры являются SPNE Рамсея тогда и только тогда, когда выполняется неравенство*

$$\int_0^{r^g} f(t) \Delta t > \frac{r^g}{2}. \quad (2.25)$$

Для лучшей наглядности теоремы 2 и лучшего понимания приведём несколько следствий и рассмотрим пример. Как уже было указано ранее, мы концентрируемся на двух случаях: разнородной общественности и вероятностных моделях.

2.3.1 Разнородная общественность. Во-первых, предположим, что существует N различных общественных групп (профсоюзов) p_1, \dots, p_N с соответствующими r^{p_1}, \dots, r^{p_N} и размерами $s_1, \dots, s_N \in [0, 1]$, которые удовлетворяют естественному предположению $\sum_{i=1}^N N s_i = 1$. Чтобы свести наше внимание к первому ходу правительства, предположим что r^g является кратным для r^{p_i} для всех $i = 1, \dots, N$, то есть

$$\frac{r^g}{r^{p_i}} \in \mathbb{N}.$$

Для более асинхронных случаев см. [2]. Для данного особого случая теореме 2 можно переписать следующим образом.

Следствие 1. *Рассмотрим общую несогласованную по времени игру, в которой выполняется (2.18) и общественность состоит из N различных групп. Тогда все SPNE игры являются SPNE Рамсея тогда и только тогда, когда выполняется неравенство*

$$\sum_{i=1}^N s_i (r^g - r^{p_i}) \geq \frac{r^g}{2}. \quad (2.26)$$

2.3.2 Вероятностная модель. Вернёмся к случаю с унифицированной общественностью, чтобы рассмотреть вероятностные ходы отдельно от влияния эффектов разнородной общественности. В зависимости от результатов переговоров общественность будет реагировать на первый ход правительства в какой-то момент времени из интервала $[a, b]$, $0 < a < b < r^g$, с равномерно распределённым вероятностным законом. Теорема 2 принимает вид:

Следствие 2. *Рассмотрим общую несогласованную по времени игру, в которой выполняется (2.18) и общественность делает второй ход в какой-то момент времени из интервала $[a, b]$ с равномерно распределённым вероятностным законом. Тогда все SPNE игры являются SPNE Рамсея тогда и только тогда, когда выполняется неравенство*

$$\frac{b^2 - a^2}{2} \geq b - \frac{r^g}{2}. \quad (2.27)$$

Очевидно, что можно рассматривать произвольные комбинации описанных выше подходов, то есть разнородных игроков с вероятностными ходами, вероятность которых может быть описана некоторыми функциями распределения. Приведём пример.

Пример 1. Рассмотрим экономику с двумя профсоюзами, каждый из которых состоит из s_1 и s_2 рабочих соответственно, так что выполняется $s_1 + s_2 = 1$. Примем в качестве одного периода квартал (раз в квартал выпускаются сводки по макроэкономике). Предположим, что $r^g = 5$ и что время реакции профсоюзов составляют как минимум 1 и 3 квартала, но как максимум – 1,5 и 3,5 квартала соответственно. Для простоты предположим, что решение, принятое за полтора квартала подчиняется равномерному закону распределения. Тогда получим:

$$\mathbb{T} := 0 \cup [1, 1.5] \cup r^g,$$

$$\mathbb{T} := 0 \cup [3, 3.5] \cup r^g.$$

Функция реакции тогда имеет вид $f : \mathbb{T}_1 \cup \mathbb{T}_2 \rightarrow [0, 1]$:

$$f(x) \begin{cases} 0 & \text{if } x = 0, \\ 2s_1(x - 1) & \text{if } x \in [1, 1.5], \\ s_1 + 2s_2(x - 3) & \text{if } x \in [3, 3.5], \\ 1 & \text{if } x = 5. \end{cases}$$

Интегрируя по $\mathbb{T}_1 \cup \mathbb{T}_2$ получим

$$\int_0^5 f(x) \Delta x = \frac{15}{4}s_1 + \frac{7}{4}s_2.$$

Предположение (2.25) теоремы 2 удовлетворяется (и следовательно единственное равновесие является равновесием Рамсея) тогда и только тогда, когда выполняется

$$s_1 \geq \frac{3}{8}, \text{ или, что эквивалентно, } s_2 \leq \frac{5}{8}.$$

Интуитивно понятно, что профсоюз, принимающий решения быстрее, должен быть достаточно велик для того, чтобы общая реакция общественности была достаточно быстрой, чтобы препятствовать увеличению инфляции правительством.

2.4 Классическая модель профсоюз—монополист

Данная модель является моделью отношений профсоюз—фирма по Леонтьеву [3], в которой профсоюз задаёт уровень заработной платы, после чего фирма выбирает желаемое количество наёмных работников (уровень найма). Предположим что фирма оперирует в условиях конкурентного рынка с рыночной ценой P .

Порядок игры следующий:

Первый этап. Профсоюз устанавливает уровень заработной платы W .

Второй этап. Фирма выбирает уровень найма E .

Выигрыши составляют:

Функция полезности профсоюза имеет вид

$$U = U(W, E), \quad \frac{\partial U}{\partial W} > 0; \quad \frac{\partial U}{\partial E} > 0.$$

Например $U = \lambda W E$, где $\lambda \in (0; 1)$

Полезность для фирмы измеряется как прибыль

$$\Pi = PY(\bar{K}, E) - WE,$$

где цена P дана, а капитал \bar{K} фиксирован. Отсюда мы можем переписать

$$\Pi(W, E) = R(E) - WE,$$

где R – доход.

Решение методом обратной индукции:

Второй этап. Фирма максимизирует свою прибыль по E при заданном уровне заработной платы. Условие первого порядка примет вид

$$W = R'(E) = MPE.$$

Разрешая относительно E получим кривую спроса: $E = g(W)$.

Первый этап. Решается

$$\max_W U(W, E) = U(W, g(W)).$$

Решение представлено на графике 2.2. Графически можно показать, что исход игры неэффективен. Точка X является точкой равновесия обратной ин-

дукции модели монополии профсоюза. Точки в закрашенной области Парето-предпочтительнее X . AB является кривой контракта, состоящей из точек, в которых изо-профиты и кривые безразличия профсоюза имеют общий тангенс [4].

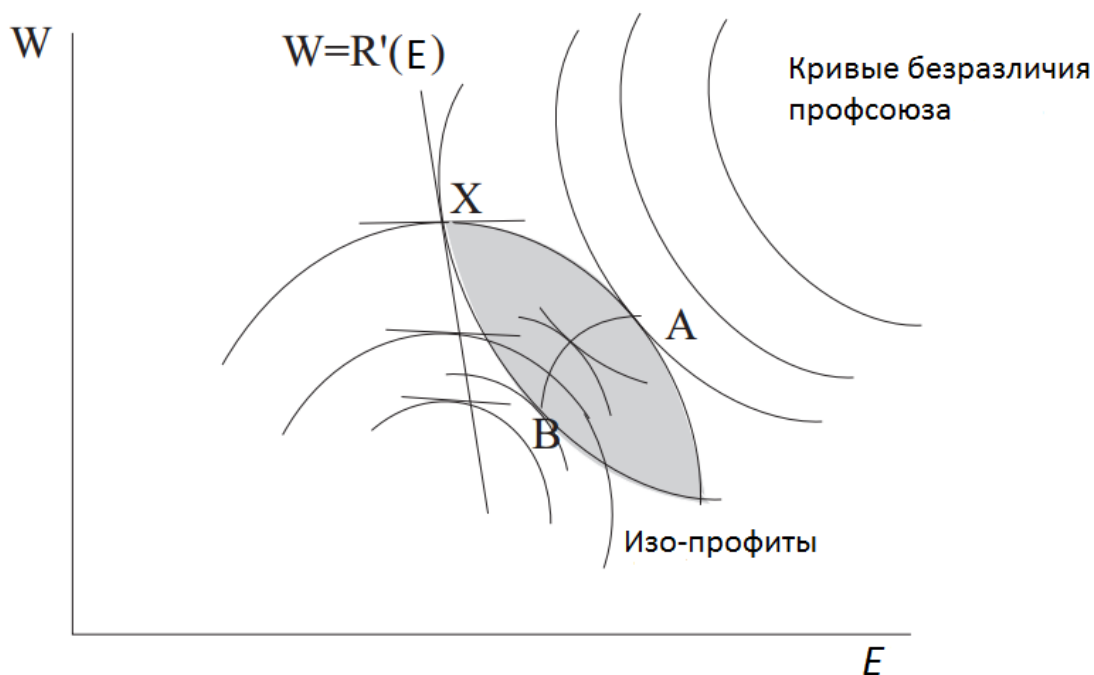


Рис 2.2

2.5 Модель профсоюз—монополист на временных шкалах

В игре, как и в непрерывной модели присутствует два игрока: профсоюз P и фирма F , чьими рычагами влияния на игру являются W - зарплата рабочего и E - количество нанятых рабочих соответственно. Каждый игрок выбирает из следующих стратегий: низким L и высоким H уровнем повышения. В общем виде игра может быть задана следующей матрицей выигрышей:

		Профсоюз	
		L	H
Фирма	L	a, q	b, v
	H	c, x	d, z

Функция полезности профсоюза $U_t(W, E) = \lambda W E$, где $\lambda \in (0; 1)$:

$$\frac{\partial U}{\partial W} > 0; \quad \frac{\partial U}{\partial E} > 0 \quad U(0, E) = U(W, 0) = U(0, 0) = 0,$$

Функция полезности фирмы $\Pi_t(W, E) = cP(\bar{K}, E) - WE$:

$$P(\bar{K}, E) = A\bar{K}^\alpha E^\beta,$$

где A – коэффициент нейтрального технического прогресса, α и β – коэффициенты эластичности валового внутреннего продукта по капитальным и трудовым затратам.

Для профсоюза соотношения функции полезности для всех стратегий будет следующим:

$$U(L, L) < U(L, H) \approx U(H, L) < U(H, H), \quad (2.28)$$

где соотношение $U(L, H) \approx U(H, L)$ неопределенное, так как это чисто "политический" момент, что для профсоюза более выгодно: большее количество людей, получающих меньшую зарплату или меньшее количество людей, получающих большую зарплату.

Для нахождения соотношений между функциями полезности для фирмы построим график 2.3.

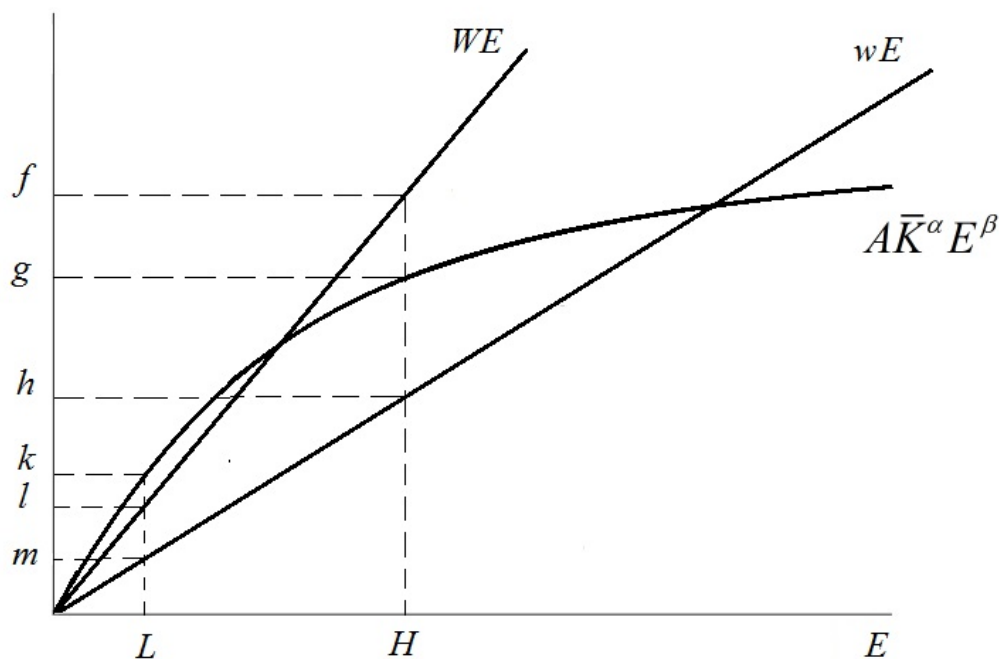


Рис 2.3

где w - низкий уровень зарплат по стратегии L , а W - высокий уровень зарплат по стратегии H . Принимая во внимание, что все коэффициенты кон-

станты за исключением количества нанятых и уровня зарплат, легко вывести следующее:

$$\Pi(H, H) = g - f < \Pi(H, L) = k - l < \Pi(L, L) = k - m < \Pi(L, H) = g - h. \quad (2.29)$$

Следовательно матрица выигрышей будет следующей:

		Профсоюз	
		L	H
Фирма	L	a, q	b, v
	H	c, x	d, z

Таблица 2.2

где $q < x \approx v < z, 0 > d < b < a < c$

РАЗДЕЛ 3 КОМПЬЮТЕРНАЯ ИМИТАЦИЯ МАКРОЭКОНОМИЧЕСКИХ МОДЕЛЕЙ

3.1 Анализ устойчивости стратегий в модели Барро-Гордона

Положим, что матрица выигрышей двух игроков соответствует 2.1.

		Общественность	
		L	H
Правительство	L	0,0	-1, -1
	H	$\frac{1}{2}, -1$	$-\frac{1}{2}, 0$

Таблица 3.1

$r^g = 7$ - количество ходов совершаемое правительством за игру

$r^p = 4$ - количество ходов совершаемое обществом за игру

Проверим выполнение условия теоремы 2.21 для матрицы выигрышей 3.1.

$$r^g > \bar{r}^g(R) = \begin{cases} 2r^p = 2r^p, & \text{если } R = 0 \\ 2(1+R)r^p = 2(1+R)r^p, & \text{если } R \in \left(0; \frac{1}{2}\right) \\ 2Rr^p = 2Rr^p, & \text{если } R \in \left(\frac{1}{2}; 1\right) \end{cases}$$

$$r^g > \bar{r}^g(R) = 0$$

$$4 > 0$$

Следовательно $\frac{r^g}{r^p} \in \left(\frac{3}{2}; 2\right)$ выбраны правильно и все совершенные равновесия по под-играм должны быть совершенными равновесиями по под-играм Рамсея.

– Рассмотрим случай, когда правительство скорее склонно ввести высокий уровень инфляции, а общественность предполагая, что правительство пойдет на этот шаг с высокой долей вероятности поднимет зарплаты:

$q^g = [0.5; 0.9; 0.7; 0.5; 0.9; 0.73; 0.8]$ - соответствующая скалярная функция

$q^p = [0.8; 0.9; 0.8; 1]$ - соответствующая скалярная функция

С помощью программы проведем эмуляцию 100 игр с такими исходными данными.

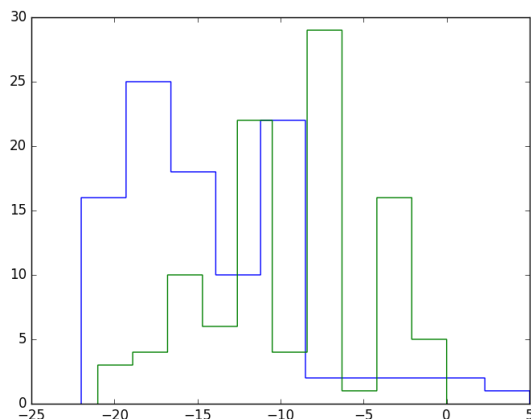


Рис 3.1 – Гистограмма выигрышей, синий - правительство, зеленый - общественность

	Правительство	Общественность
Среднее	-14.26	-9.46
Стандартное отклонение	5.34	4.66
Ассиметрия	1.10	-0.041
Эксцесс	1.37	-0.37

Согласно полученным результатам легко видеть, что подобный набор стратегий невыгоден обеим сторонам, к тому же правительству из-за более "мягкого" подхода "более невыгодно".

– Рассмотрим случай, когда правительство на последнем шаге может отступить от оптимальной для обоих игроков стратегии (L, L) .

Для этого положим скалярные функции:

$$q^g = [0; 0; 0; 0; 0; 0; 0.3]$$

$$q^p = [0; 0; 0; 0]$$

	Правительство	Общественность
Среднее	0.39	-0.78
Стандартное отклонение	0.78	1.56
Ассиметрия	1.6	-1.11
Эксцесс	0.58	0.58

Данная стратегия является выигрышной для правительства, если общественность не ожидает подобного хода под конец действия срока правительства.

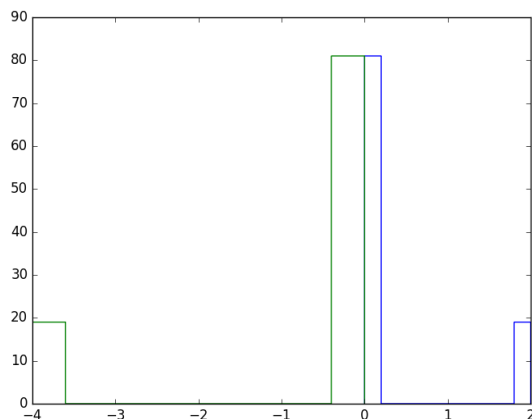


Рис 3.2 – Гистограмма выигрышей, синий - правительство, зеленый - общественность

– Рассмотрим случай, когда общественность доверяет правительству и почти уверенно, что оно не может поднять уровень инфляции к окончанию срока своего правления.

Для этого положим скалярные функции:

$$q^g = [0; 0; 0; 0; 0; 0; 0.3]$$

$$q^p = [0; 0; 0; 0.15]$$

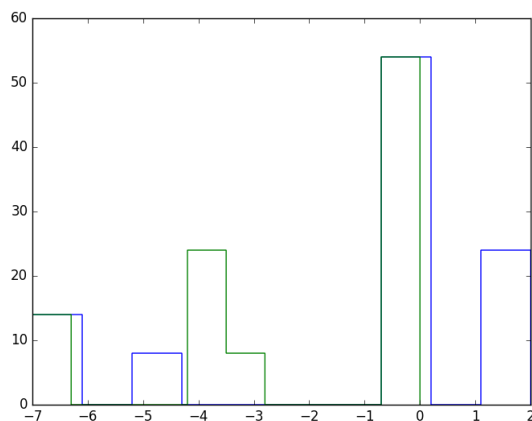


Рис 3.3 – Гистограмма выигрышей, синий - правительство, зеленый - общественность

Легко видеть, что данная стратегия однозначно ухудшает позицию правительства, но так же не выгодна общественности.

– Рассмотрим случай, когда общественность не доверяет правительству и почти уверенно, что оно повысит уровень инфляции к окончанию срока своего правления.

	Правительство	Общество
Среднее	-0.9	-2.18
Стандартное отклонение	3	2.58
Ассиметрия	-1.16	-0.68
Эксцесс	-0.06	-0.95

Для этого положим скалярные функции:

$$q^g = [0; 0; 0; 0; 0; 0; 0.3]$$

$$q^p = [0; 0; 0; 0.8]$$

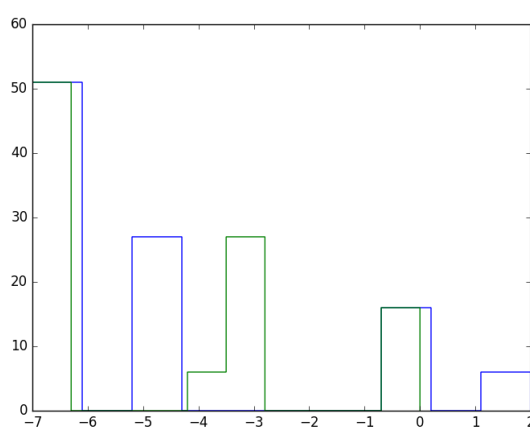


Рис 3.4 – Гистограмма выигрышей, синий - правительство, зеленый - общество

	Правительство	Общество
Среднее	-4.8	-4.62
Стандартное отклонение	2.99	2.65
Ассиметрия	1.18	0.54
Эксцесс	-0.12	-1.15

Данная стратегия является равносильно невыгодна для обоих игроков.

В связи с полученными результатами можно сделать вывод, что если правительство имеет больше шагов на временных шкалах, то ей имеет смысл отклониться от оптимальной по Парето стратегии L, L и повысить уровень инфляции под конец срока своего правления, так как даже если общество будет ожидать такого хода, то с целью минимизировать свои потери не будет повышать индексирование заработной платы заранее. Естественно

если правительство захочет заручиться поддержкой общественности на следующих выборах, то такой ход с её стороны будет сродни "политическому самоубийству".

3.2 Анализ устойчивости стратегий в модели профсоюз—монополист

Положим матрицу выигрышей в соответствии с 2.2 следующим образом:

		Профсоюз	
		L	H
Фирма	L	3, 1	2, 3.9
	H	7, 4	-3, 7

Для нахождения равновесия по Нэшу посчитаем следующее:

– Фирма

$$L : 3\alpha + 2(1 - \alpha) = \alpha + 2$$

$$H : 7\alpha - 3(1 - \alpha) = 10\alpha - 3$$

$$10\alpha - 3 = \alpha + 2$$

:

$$\alpha = \frac{5}{9}$$

– Профсоюз

$$L : \beta + 4(1 - \beta) = -3\beta + 4$$

$$H : 3.9\beta + 7(1 - \beta) = -3.1\beta + 7$$

$$0.1\beta = 3$$

$$\beta = 30$$

Следовательно равновесием по Нэшу будет стратегия L, H .

В каждом столбце матрицы фирмы найдем максимальный элемент. Затем в каждой строке матрицы профсоюза выберем наибольший элемент. Платежная матрица фирмы:

3	1	2	3.9
7	4	-3	7

Оптимальной по Парето будет стратегия (H, L) .

$r^f = 4$ - количество ходов совершаемое фирмой за игру

$r^p = 3$ - количество ходов совершаемое профсоюзом за игру

– Рассмотрим случай, когда оба игрока стараются следовать равновесной стратегии по Нэшу:

$q^f = [0.9; 0.9; 0.9; 0.9]$ - соответствующая скалярная функция фирмы

$q^p = [0.1; 0.1; 0.1]$ - соответствующая скалярная функция профсоюза

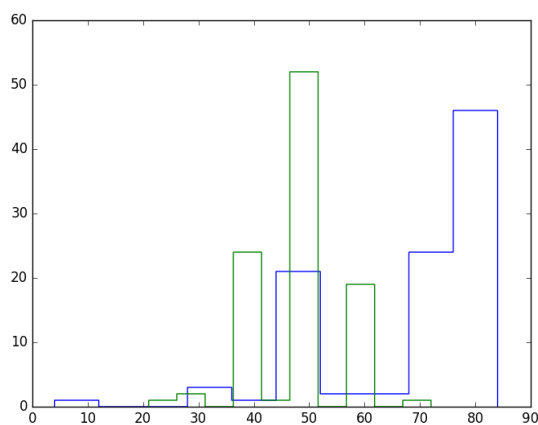


Рис 3.5 – Гистограмма выигрышей, синий - фирма, зеленый - профсоюз

	Фирма	Профсоюз
Среднее	69.02	47.84
Стандартное отклонение	18.10	8.19
Ассиметрия	-1.04	0.017
Эксцесс	0.25	0.69

– Рассмотрим случай, когда профсоюз пытается увеличивать зарплату, на своём последнем шаге, а фирма для выхода из нерентабельного положения сокращает количество сотрудников.

$q^f = [0.9; 0.9; 0.9; 0.1]$ - соответствующая скалярная функция фирмы

$q^p = [0.1; 0.1; 0.9]$ - соответствующая скалярная функция профсоюза

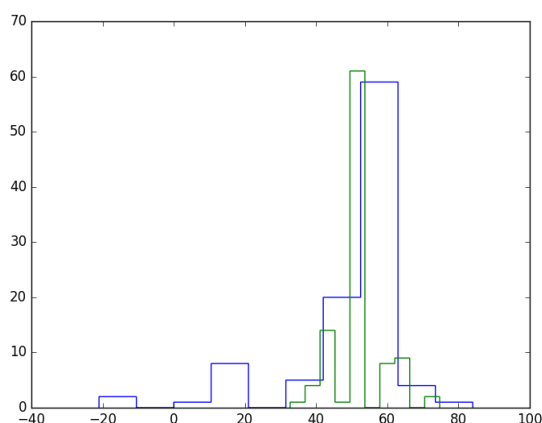


Рис 3.6 – Гистограмма выигрышей, синий - фирма, зеленый - профсоюз

	Фирма	Профсоюз
Среднее	50.48	51.1
Стандартное отклонение	17.18	7.17
Ассиметрия	-2.02	0.57
Эксцесс	5.21	1.44

такая стратегия незначительно улучшит позиции профсоюза, но заметно пошатнет позицию фирмы.

– Рассмотрим случай, когда профсоюз увеличит зарплату во второй раз и вернется к оптимальной стратегии на третий период:

$q^f = [0.9; 0.9; 0.1; 0.9]$ - соответствующая скалярная функция фирмы

$q^p = [0.1; 0.9; 0.1]$ - соответствующая скалярная функция профсоюза

	Фирма	Профсоюз
Среднее	43.06	50.64
Стандартное отклонение	14.47	7.38
Ассиметрия	-1.06	-0.29
Эксцесс	1.30	0.09

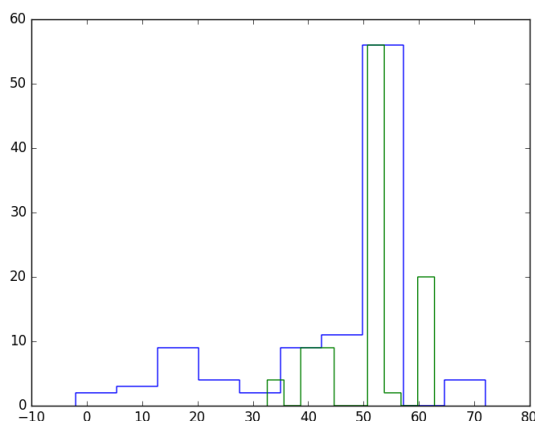


Рис 3.7 – Гистограмма выигрышей, синий - фирма, зеленый - профсоюз

Снова таки разовая акция поднятия зарплат даёт профсоюзу незначительный выигрыш, но фирма при этом проигрывает относительно много больше, чем выигрывает профсоюз.

– Рассмотрим случай, когда профсоюз поднимает на втором своём ходе зарплату, но не опускает его вплоть до конца игры:

$q^f = [0.9; 0.9; 0.1; 0.2]$ - соответствующая скалярная функция фирмы

$q^p = [0.1; 0.9; 0.8]$ - соответствующая скалярная функция профсоюза

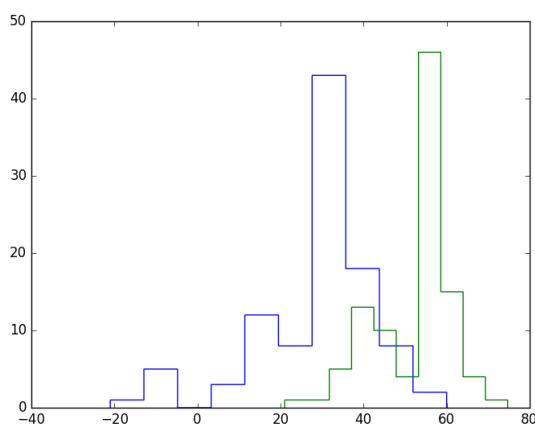


Рис 3.8 – Гистограмма выигрышей, синий - фирма, зеленый - профсоюз

Фирма снова теряет в относительных деньгах, в то время как профсоюз только незначительно увеличивает своё положение. Разумно с точки зрения фирмы сделать предложение профсоюзу не менять уровень зарплат с оптимального, а взамен отплачивать разным видом бонусов. Тогда функция

	Фирма	Профсоюз
Среднее	30.37	51.37
Стандартное отклонение	13.64	8.99
Ассиметрия	-1.36	-0.53
Экссесс	2.74	0.8

полезности изменится для обоих игроков на равную величину. Фирме стоит подобрать эту величину бонусов так, чтобы остаток между её оптимальным выигрышем после вычитания бонусов был чем-то средним между собственными выигрышами, когда профсоюз устанавливает высокий уровень зарплат и низкий.

ВЫВОДЫ

В данной работе была рассмотрена игровая интерпретация модели Барро-Гордона на временных шкалах и построена модель "профсоюз—монополист". Разработано программное обеспечение для множественной имитации игры на временных шкалах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Choi S., Smith B. “Inflation, Financial Markets, and Capital Formation / Smith B. Choi, S., J.H. Boyd // Federal Reserve Bank of St. Louis. — 1996. — Vol. 14. — Pp. 41–58.
- [2] J. Libich, P. Stehlic. Incorporating Rigidity in the Timing Structure of Macroeconomic Games / P. Stehlic J. Libich // Economic Modelling. — 2007. — Apr. — 29 P.
- [3] Leontief, W. Input-Output Economics / W. Leontief. — 2-nd edition. — 200 Madison Avenue, New York : Oxford University Press, Inc., 1946. — 436 P.
- [4] Toi, L. Game Theory / L. Toi. — Shandong : School of economics, Shandong university, 2012. — 67 P.

ДОДАТОК А

Информация о программе

В рамках дипломной работы была разработана программа, которая эмулирует биматричную игру на временных шкалах и доказывает эмпирически изложенные в работе результаты.

Разработка программного продукта велась на языке Python 3.4. Программа позволяет пользователю задать r^g , r^p и соответствующие скалярные функции поведения игрока в определенный момент времени. Значениями данной функции будет вероятность выбора игроком стратегии H в данный период времени. Так же пользователь может задать количество запусков игры с входящими данными для подсчета статистики. В качестве интерфейса входящих данных был выбран txt файл.

Пояснения к коду

`util.fromFileToMap(filepath)`

Функция парсит входящие данных

`nash.calculate_nash(government_payoffs, public_payoffs)`

Функция рассчитывает равновесие по Нэшу для стандартной игры.

`nash.time_scales_game(request, government_payoffs, public_payoffs))`

Имитация игры на временных шкалах, результатом который есть вектор выигрышей игроков за время игры

`util.create_all_stats(array, player_name, time)`

Рассчитывает статистические параметры, строит гистограмму для выигрышей одного игрока.

Листинг

entryPoint.py:

```
from datetime import datetime
import nash
import util
```

```
ClientInput = util.fromFileToMap(util.parse().filePath)
```

```
GovernmentPayoffs = ['Government',
                      ('L', 'L', 3),
                      ('L', 'H', 5),
                      ('H', 'L', 2),
                      ('H', 'H', -1)]
```

```
PublicPayoffs = ['Public',
                  ('L', 'L', 0),
                  ('L', 'H', 2),
                  ('H', 'L', 3),
```

```

('H', 'H', 7)]

nash.calculate_nash(GovernmentPayoffs, PublicPayoffs)

time = datetime.timestamp(datetime.now())
stop = int(util.parse().iterations)
mat = [[0] * stop for i in range(2)]
for i in range(0, stop):
    game = nash.time_scales_game(ClientInput, GovernmentPayoffs, PublicPayoffs)
    for j in range(0, 2):
        mat[j][i] = game[j]

util.create_all_stats(mat[0], 'Government', time)
util.create_all_stats(mat[1], 'Public', time)

print("␣")

```

util.py

```

import argparse
import os
import re

import matplotlib.pyplot as plt
import scipy.stats.stats as st
from numpy import std

def fromFileToMap(filePath):
    newDict = {}
    with open(filePath) as f:
        for line in f:
            line = re.split("#", line)
            splitLine = line[0].split()
            newDict[splitLine[0]] = ",".join(splitLine[1:])
    return newDict

def parse():
    parser = argparse.ArgumentParser()
    parser.add_argument('-f', '--filePath', default="files/income.txt")
    parser.add_argument('-i', '--iterations', default=1)
    return parser.parse_args()

def lcm(a, b):
    m = a * b
    while a != 0 and b != 0:
        if a > b:
            a %= b
        else:
            b %= a
    return m // (a + b)

```



```

def add(x, y):
    return list(map(lambda a, b: a + b, x, y))

def make_histogramm(array, player_name, path):
    plt.hist(array, histtype='step')
    file_name = str(player_name) + ".png"
    try:
        os.mkdir(path, mode=0o777)
    except OSError:
        pass
    plt.savefig(path + file_name, format='png')

def make_all_calculations(array, player_name, file):
    file.write(str(player_name + "s_average=" + str(float(sum(array)) / len(array)) + "\n"))
    file.write(str(player_name + "s_standard_deviation=" + str(std(array)) + "\n"))
    file.write(str(player_name + "s_asymmetry=" + str(st.skew(array, bias=False)) + "\n"))
    file.write(str(player_name + "s_excess=" + str(st.kurtosis(array, bias=False)) + "\n"))

def create_all_stats(array, player_name, time):
    path = "results/game" + str(time) + "/"
    make_histogramm(array, player_name, path)

    file = open(path + "allEquations.txt", "a")
    make_all_calculations(array, player_name, file)

```

nash.py

```

import math
import random

import globalConstants
import util

class Player:
    def __init__(self, name, order, strategy_space, payoffs, choice, suboptimal, strategies,
                 state, game_play):
        self.name = name
        self.order = order
        self.strategy_space = strategy_space
        self.payoffs = payoffs
        self.choice = choice
        self.suboptimal = suboptimal
        self.strategies = strategies
        self.state = state
        self.game_play = game_play

    def process_game(self, G):
        for i in range(0, len(G)):
            X = G[i]

```

```

        if X[0] == self.name:
            for j in range(1, len(X)):
                Branch = X[j]
                Alternative = list(Branch)
                del Alternative[len(Aalternative) - 1]
                self.strategy_space = self.strategy_space + [tuple(Aalternative)]
                self.payoffs = self.payoffs + [Branch[len(Branch) - 1]]

def evaluate(self):
    X = []
    for i in range(0, len(self.strategy_space)):
        Alternative1 = self.strategy_space[i]
        for j in range(0, len(self.strategy_space)):
            Alternative2 = self.strategy_space[j]
            if Alternative1 != Alternative2:
                if len(Aalternative1) == len(Aalternative2):
                    Compare = 0
                    for k in range(0, len(Aalternative1) - 1):
                        if Alternative1[k] == Alternative2[k]:
                            Compare = Compare + 0
                        else:
                            Compare = Compare + 1
                    if Compare == 0:
                        PayoffCompare = [self.payoffs[i], self.payoffs[j]]
                        M = max(PayoffCompare)
                        if self.payoffs[i] == M:
                            self.choice = Alternative1
                            X = X + [self.choice]
                        else:
                            self.suboptimal = self.suboptimal + [Alternative1]
                    if self.payoffs[j] == M:
                        self.choice = Alternative2
                        X = X + [self.choice]
                    else:
                        self.suboptimal = self.suboptimal + [Alternative2]

    X = set(X)
    self.suboptimal = set(self.suboptimal)
    self.strategies = list(X - self.suboptimal)
    print("\nStrategies_selected_by_" + self.name + ":")
    print(self.strategies)
    for l in range(0, len(self.strategies)):
        strategy = self.strategies[l]
        for m in range(0, len(strategy)):
            0 = self.order[m]
            self.state[0] = strategy[m]
            self.game_play = self.game_play + [tuple(self.state)]

class Game:
    def __init__(self, players, structure, optimal):
        self.players = players
        self.structure = structure
        self.optimal = optimal

```

```

def nash(self, GP):
    Y = set(GP[0])
    for i in range(0, len(GP)):
        X = set(GP[i])
        Y = Y & X
    self.optimal = list(Y)
    if len(self.optimal) != 0:
        print("\nThe pure strategies Nash equilibrium are:")
        for k in range(0, len(self.optimal)):
            print(self.optimal[k])
    else:
        print("\nThis game has no pure strategies Nash equilibrium!")

class TimeScaleGame(Game):
    def __init__(self, players, structure, optimal, government_period, public_period,
                 government_strategy,
                 public_strategy):
        super().__init__(players, structure, optimal)
        self.government_period = government_period
        self.public_period = public_period
        self.government_strategy = government_strategy
        self.public_strategy = public_strategy

    def play(self):
        fullTime = util.lcm(self.government_period, self.public_period)
        self.government_strategy = self.getStrategyForAllPeriod(self.government_strategy,
                                                                self.government_period,
                                                                fullTime)

        self.public_strategy = self.getStrategyForAllPeriod(self.public_strategy, self.
                                                                public_period, fullTime)
        self.payoff = [0, 0]
        for x in range(0, fullTime):
            self.payoff = util.add(self.getPayoff(x), self.payoff)
        return self.payoff

    def getPayoff(self, x):
        g_strat = self.government_strategy[x]
        p_strat = self.public_strategy[x]
        summ = []
        for i in range(0, len(self.structure)):
            X = self.structure[i]
            for d in range(0, len(self.players)):
                if X[0] == self.players[d]:
                    for j in range(1, len(X)):
                        Branch = X[j]
                        Alternative = list(Branch)
                        del Alternative[len(Alternative) - 1]
                        if Alternative == [g_strat, p_strat]:
                            summ.append(list(Branch)[len(Branch) - 1])
        return summ

    def getStrategyForAllPeriod(self, strategy, period, fullTime):
        new_strategy = []

```

```

changePeriod = fullTime / period
tmp = -1
for x in range(0, fullTime):
    index = math.floor(x / changePeriod)
    if index != tmp:
        probability = float(strategy[index])
        rand = random.randrange(0, 100) / 100
        if probability > rand:
            new_strategy.append('H')
        else:
            new_strategy.append('L')
        tmp = index
    else:
        new_strategy.append(new_strategy[x - 1])
return new_strategy

def calculate_nash(government_payoffs, public_payoffs):
    # game(players,structure,plays,optimal)
    game = Game(('Government', 'Public'), [government_payoffs, public_payoffs], None)
    # player(name,order,strategySpace,payoffs,choice,suboptimal,strategies,state,gameplay):
    player1 = Player('Government', (1, 0), [], [], None, [], None, [0, 0], [])
    player2 = Player('Public', (0, 1), [], [], None, [], None, [0, 0], [])
    players = [player1, player2]
    for i in range(0, len(players)):
        players[i].process_game(game.structure)
        players[i].evaluate()
    GP = []
    for i in range(0, len(players)):
        X = players[i].game_play
        GP = GP + [X]
    game.nash(GP)

def time_scales_game(request, government_payoffs,
                     public_payoffs):
    government_period = request.get(globalConstants.GOVERNMENT_TIME_PERIOD)
    public_period = request.get(globalConstants.PUBLIC_TIME_PERIOD)
    government_strategy = str(request.get(globalConstants.GOVERNMENT_SCALAR_STRATEGY)).split(
        ',')
    public_strategy = str(request.get(globalConstants.PUBLIC_SCALAR_STRATEGY)).split(',')

    game = TimeScaleGame(('Government', 'Public'), [government_payoffs, public_payoffs],
                        None, int(government_period),
                        int(public_period), government_strategy, public_strategy)
    return game.play()

```