Вестник ВГУ. Серия: Экономика и управление. 2025. № 4. С. 20–33.
Proceedings of Voronezh State University. Series: Economics and Management. (4), 20–33.

Вестник Воронежского государственного университета
Серия: Экономика и управление

Математические и инструментальные методы экономики

Научная статья
УДК 330:51-8
DOI: https://doi.org/10.17308/econ.2025.4/13355
JEL: C73

«Индонезийский покер». Полное решение игры из шоу «Тейбл тайм»

Е. М. Крохалев[1], В. С. Попов[2], А. В. Савватеев[3, 4, 5, 6]✉

[1] ООО «СофтКом», Волковский пр., 32, лит. А, 192102, Санкт-Петербург, Российская Федерация
[2] Воронежский государственный университет, Университетская пл., 1,
394018, Воронеж, Российская Федерация
[3] Адыгейский государственный университет, ул. Первомайская, 208,
385000, Майкоп, Российская Федерация
[4] Московский физико-технический институт, ул. Керченская, 1А, к. 1,
117303, Москва, Российская Федерация
[5] Университет Иннополис, ул. Университетская, 1, 420500, Иннополис, Российская Федерация
[6] Центральный экономико-математический институт Российской
академии наук, Нахимовский пр., 47, 117418, Москва, Российская Федерация

Предмет и мотивация исследования. В статье дан стратегический разбор игры «Индонезийский покер» из популярного интернет-шоу «Тейбл тайм». Игроки переворачивают фишки с номерами от 1 до 12, в зависимости от выпавших значений на двух кубиках можно переворачивать либо карточку, номер которой равен сумме выпавших цифр, либо пару карточек с номерами выпавших на кубиках цифр. Мотивация такого рода разборов состоит в идее привить широким массам населения подлинный интерес к научным исследованиям, используя изначально развлекательный контекст, что до сих пор не было представлено в научной литературе.
Цель. Нахождение стратегии, минимизирующей среднее время до завершения игры. Отметим, что это не означает победу даже в среднем, ибо против найденной стратегии закрытия карточек даже в игре всего двух игроков оптимальным ответом может быть какая-то субоптимальная стратегия, что совершенно контринтуитивно. Этот парадокс обсуждается в работе; к сожалению, найти равновесную стратегию, гарантирующую победу в среднем против любой другой, авторам пока не удалось в силу значительной (дважды экспоненциальной) сложности множества всех мыслимых стратегий в этой игре.
Дизайн исследования. В работе формулируется несколько теорем, значительно сокращающих сложность компьютерного перебора всех мыслимых стратегий. Затем вычисляется оптимальная стратегия, которая оказывается чрезвычайно запутанной в применении. Возникает вопрос о существовании стратегии, алгоритмически гораздо более простой, но не сильно уступающей оптимальной.
Результаты. В работе дается всестороннее описание оптимальной стратегии. Кроме того, демонстрируется малозаметная тонкость, заключающаяся в стратегическом пропуске хода в ряде позиций при определенных результатах бросков. Далее угадывается очень простая в использовании субоптимальная стратегия, проигрывающая оптимальной буквально на флажке. Как оптимальная, так и субоптимальная простая стратегии значительно улучшают шансы на победу в игре против «болвана», ходящего всегда случайным образом.

Ключевые слова: теория игр, динамическое программирование, оптимальная стратегия.

Для цитирования: Крохалев, Е. М., Попов, В. С., & Савватеев, А. В. (2025). «Индонезийский покер». Полное решение игры из шоу «Тейбл тайм». *Вестник Воронежского государственного университета. Серия: Экономика и управление*, (4), 20–33. DOI: https://doi.org/10.17308/econ.2025.4/13355

Proceedings of Voronezh State University
Series: Economics and Management

Mathematical and Quantitative Methods

Original article
UDC 330:51-8
DOI: https://doi.org/10.17308/econ.2025.4/13355
JEL: C73

"Indonesian Poker": complete game solution from the "Table Time" show

E. M. Krokhalev[1], V. S. Popov[2], A. V. Savvateev[3, 4, 5, 6]✉

[1] SoftCom LLC, Saint Petersburg, Russian Federation

[2] Voronezh State University, 1 Universitetskaya Sq., 394018, Voronezh, Russian Federation

[3] Adyghe State University, 208 Pervomayskaya St., 385000, Maykop, Republic of Adygea, Russian Federation

[4] Moscow Institute of Physics and Technology, 1A Kerchenskaya St., Bldg. 1, 117303, Moscow, Russian Federation

[5] Innopolis University, 1 Universitetskaya St., 420500, Innopolis, Russian Federation

[6] Central Economics and Mathematics Institute, Russian Academy of Sciences, 47 Nakhimovsky Ave., 117418, Moscow, Russian Federation

Subject and motivation. This article provides a strategic analysis of the "Indonesian Poker" game from the popular online show "Table Time". Players flip numbered chips (1–12) based on outcomes from two dice; they may flip either the chip corresponding to the sum of the dice or the pair of chips matching the individual die values. The motivation for such analyses lies in cultivating genuine public interest in scientific research through initially entertaining context – an approach not yet documented in academic literature.
Purpose. To identify a strategy minimizing the average time to game completion. Notably, this does not guarantee victory even on average, as against the found chip-flipping strategy, even in a two-player game, the optimal response might be a suboptimal strategy – a highly counterintuitive outcome. This paradox is discussed in the paper; unfortunately, the authors have yet to find an equilibrium strategy guaranteeing average victory against any other due to the substantial (doubly exponential) complexity of all conceivable strategies in this game.
Research Design. The paper formulates several theorems that significantly reduce the computational complexity of exhaustive strategy search. The optimal strategy is then computed, proving extremely intricate in practice. This raises the question of whether an algorithmically simpler strategy exists that performs nearly as well as the optimal one.
Results. The paper provides a comprehensive description of the optimal strategy. Additionally, it reveals a subtle strategic nuance: deliberately skipping turns in certain positions based on specific dice outcomes. Then we proposed a very simple-to-use suboptimal strategy, that is losing to the optimal strategy by the narrowest margin. Both the optimal and simple suboptimal strategies substantially improve winning chances against a "naïve" player who always moves randomly.

Key words: game theory, dynamic programming, optimal strategy.

For citation: Krokhalev, E. M., Popov, V. S., & Savvateev, A. V. (2025). "Indonesian Poker": complete game solution from the "Table Time" show. *Proceedings of Voronezh State University. Series: Economics and Management*, (4), 20–33. DOI: https://doi.org/10.17308/econ.2025.4/13355

Введение

Теория игр прочно заняла центральное место в современной экономической науке. Классические модели стратегического взаимодействия, начиная с фундаментальной работы Von Neumann & Morgenstern (1944), сосредоточены на анализе формализованных ситуаций принятия решений, допускающих строгий математический разбор и вычисление равновесий в духе равновесия по Нэшу и его аналогов и обобщений.

При этом вопрос осмысленности модели является краеугольным камнем любого экономического исследования, проводимого с применением математических методов, будь то оптимизация или теория игр. Большой процент даже опубликованных работ остается незамеченным для широкой экономической публики в силу фактической оторванности построенной модели от изначальной проблемной ситуации. Здесь хотелось бы пройти между Сциллой и Харибдой, не усложняя модель излишними подробностями, но и не игнорируя очевидно существенные соображения. Сделать это получается не у многих. Соответствующие модели входят в сокровищницу книг по экономике, нередко добиваются признания Нобелевским комитетом и т. д.

В какой-то момент был сформулирован «критерий качества» исследования по математической экономике. Он состоит в том, что исследование либо идеально описывает конкретную проблемную ситуацию, возможно, пожертвовав изящностью, либо содержит далеко идущие обобщения и математические результаты высокого теоретического уровня, достойные с чисто математических позиций.

Мы предлагаем добавить в этот критерий третий вариант: исследование нашумевших игр, которые появляются в интернете. В настоящее время значительный пласт таких игр, возникающих спонтанно в медиасреде и приобретающих массовую популярность, продолжает оставаться за пределами академического дискурса. Старая гвардия на них смотрит скептически и «свысока», не понимая смысла разбирать этот ширпотреб.

Между тем мы полагаем, что именно такие игры позволяют наблюдать живую экономическую логику поведения «в дикой природе» – вне лабораторных условий и строгих экспериментальных рамок. Формальный анализ таких игр открывает путь к пониманию того, как реальные люди выбирают стратегии в условиях неопределенности, ограниченной информации и соревновательной динамики. Чему-то похожему посвящена ежегодная конференция Fun with algoritms[1], однако там редко встречается именно теоретико-игровой анализ интернет-игр.

Настоящее исследование делает шаг к заполнению существующего научного пробела. Мы переносим обсуждение игры «Индонезийский покер» (появившейся в шоу «Тейбл Тайм»[2]) из плоскости фольклора медиапространства в область количественного экономического анализа. На открытых лекциях один из авторов статьи, А. В. Савватеев, неоднократно демонстрировал разборы интернет-игр как примеры применения теории игр в популярной форме. Однако научные работы, в которых подобные игры подвергаются столь же подробному теоретико-игровому анализу, пока встречаются крайне редко. Большинство обсуждений остается в формате популярного рассказа, а не строгого формального исследования.

В этом смысле настоящая работа является одной из первых, показывающих, что даже неформальная интернет-игра способна стать объектом строгого математического исследования.

Интерес к подобным играм возникает естественным образом: правила просты, исходы подчинены случайности, а разница между интуитивной и оптимальной стратегией нередко оказывается существенной. Именно поэтому важен строгий научный анализ, позволяющий превратить массовый интерес в демонстрацию того, как работают современные методы теории игр и оптимального поведения.

Цель данной статьи состоит в том, чтобы провести полный теоретический разбор «Индонезийского покера», определить оптимальную стратегию игры, описать алгоритмические методы ее вычисления и проверить гипотезу о том, что следование оптимальной стратегии существенно увеличивает шансы на победу.

Мы не только строим стратегию, минимизирующую математическое ожидание числа ходов, но и сравниваем ее с альтернативными

[1] URL: https://fun2026.limos.fr/

[2] Тейбл Тайм. 3 сезон (2023). Импроком. URL: https://youtu.be/LWGIRs5P26U (момент начала обсуждения игры: 5:00:58)

стратегиями по вероятности выигрыша, тем самым приближая анализ к классическому понятию равновесия. В работе определяется полная оптимальная стратегия игры и строится простая в применении стратегия, не сильно уступающая оптимальной. Кроме того, в статье проведен всесторонний теоретический анализ игры и предложены алгоритмы для подсчета основных характеристик стратегий, посчитаны математические ожидания для всех основных стратегий, а также получены вероятности победы/поражения/ничьей для любых их пар.

Полученные результаты показывают, что даже в игре, кажущейся на первый взгляд «несерьезной», скрыты нетривиальные стратегические эффекты, а ее изучение способствует развитию методов оптимизации и стохастического моделирования, лежащих в основе современной экономической теории.

Обзор литературы

Игра «Индонезийский покер» органично вписывается в обширный ландшафт теоретико-игровых исследований. Ее механизмы наследуют и комбинируют идеи из нескольких классических разделов теории игр: игр с полной информацией, последовательных игр с принятием решений в условиях неопределенности, а также комбинаторных игр.

Наиболее близкими аналогами «Индонезийского покера» являются классические игры с костями, такие как «Яцзы» (или же покер на костях). Интересен тот факт, что данная игра также была представлена в интернете – в одном из следующих выпусков шоу «Тейбл Тайм»[3].

Фундаментальную основу для анализа подобных игр заложили Von Neumann & Morgenstern (1944) в своей основополагающей работе, где была представлена концепция игр с полной информацией. Пусть «Индонезийский покер» и содержит элемент случайности от бросков кубиков, но последовательность состояний игры (перевернутых фишек) известна всем игрокам, что позволяет отнести ее к классу игр с полной информацией и стохастическими ходами. Исследованию таких игр посвящены работы, подобные Shapley (1953), где изучается существование и свойства равновесий в стохастических играх.

Ключевым аспектом «Индонезийского покера» является механизм переворачивания фишек, что позволяет игроку переходить из текущей позиции в различные другие. Этот механизм напрямую отсылает нас к теории комбинаторных игр, в частности к игре «Ним» и ее многочисленным вариациям (Bouton, 1901). По данному классу игр существует множество фундаментальных работ, выходящих и по сей день (Sprague, 1935; Grundy, 1939; Burke et al., 2024). Однако следует заметить, что в отличие от канонического «Нима» «Индонезийский покер» имеет стохастический элемент, что выводит нас за рамки разработанной теории.

Еще одной важной теоретической параллелью являются игры с управлением ресурсами. Игрок на каждом ходу сталкивается с дилеммой: перевернуть либо две «дешевые» фишки, либо одну «дорогую». Это напоминает задачи о выборе наилучшего объекта, например задачу разборчивой невесты (Ferguson, 1989). Работы Bertsekas (2005) и Puterman (1994) демонстрируют, как в подобных динамических системах формируются оптимальные стратегии, часто имеющие пороговый характер, что можно отнести в том числе и к нашей игре.

Теперь, когда мы продемонстрировали обширные связи нашей работы с классическими сюжетами теории игр, хочется отметить, что в ней ведутся активные исследования и по сей день, в том числе разбираются игры из интернета. Иллюстрацией сказанного является международная конференция Fun with Algorithms[4]. На этой конференции представлены как фундаментальные исследования, анализирующие целый пласт типовых игр (Burke et al., 2022; 2024; Bagan et al., 2024), так и разборы отдельных настольных и интернет-игр, таких как «Морской бой» (Crombez et al., 2020), Wordle (Subercaseaux & Lokshtanov, 2022), Magic: the Gathering (Churchill et al., 2019), Matching Match (Iburi & Uehara, 2024), Swish (Dailly et al., 2024), а также разборы шахмат и подобных им игр (в том числе режим Solo Chess с сайта chess.com) (Rin & Schipper, 2024; Ambrona, 2022; Aravind et al., 2022). Подобные конференции и статьи показывают, что важен не источник игры для исследования, а математические идеи и принципы, заложенные в ней.

[3] Тейбл Тайм. 5 сезон (2025). Импроком. URL: https://youtu.be/7MZZWt25HUY

[4] URL: https://fun2026.limos.fr/

Несмотря на богатый арсенал существующих теоретико-игровых моделей, игра «Индонезийский покер» до сих пор не получила систематического научного осмысления. Во-первых, отсутствует формальная модель игры. Существующие исследования фокусируются либо на чисто комбинаторных аспектах (Bouton, 1901; Sprague, 1935; Grundy, 1939; Burke et al., 2024), либо на чисто вероятностных (Ferguson, 1989; Bertsekas, 2005; Puterman, 1994), не затрагивая гибридной природы данной игры.

Настоящая статья восполняет этот пробел, предлагая строгое определение игры как стохастического процесса с управлением на ориентированном графе позиций.

Во-вторых, отсутствует полное или даже частичное решение игры. До настоящей работы не были исследованы базовые вопросы: какова оптимальная стратегия в игре? Насколько на вероятность победы влияют умения игрока, важна ли удача? Как определить стратегию, обеспечивающую более чем 50 % вероятности выигрыша против любой другой (ее существование гарантируется теоремой Нэша)?

На бóльшую часть из этих вопросов получены ответы в настоящей статье.

Оптимальная стратегия. Математическое ожидание времени игры

Авторов в настоящей статье интересует стратегия, оптимальная в среднем, т. е. план действий, минимизирующий среднее время (в ходах) до полного переворота всех фишек. На первый взгляд, в этой игре речь в принципе не идет об игровом взаимодействии, а лишь о везении и умении выбрать правильный ход. Однако в некоторых случаях теоретически может оказаться стратегически выгодным применять иную стратегию, которая в ответ на используемые оппонентами стратегии приведет к победе с бóльшей вероятностью, несмотря на то, что будет по среднему времени уступать какой-то иной стратегии. Данное весьма тонкое соображение погружает нас в теоретико-игровой контекст, с ходу делая задачу на порядок сложнее (ибо придется искать для любого числа игроков стратегии, оптимальные в ответ друг на друга – иными словами, искать равновесие по Нэшу в игре выбора стратегий поведения; кроме того, появляются динамические аспекты, так как на каждом шаге в игре видна реализация всех случайностей в ходах произошедших до сих пор). Даже поиск симметричного равновесия, т. е. стратегии, оптимальной в ответ на саму себя, нам видится за пределами наших интеллектуальных возможностей. Мы оставляем эту задачу для будущих гениев и переходим к вычислению стратегии, оптимальной в среднем.

Назовем состоянием игры (S) множество уже перевернутых фишек. Для каждого состояния мы обозначим среднее время до завершения игры за $t[S]$. Здесь уже необходимо решить, допускаются ли стратегические пропуски ходов; вскоре мы увидим, что от этого будут зависеть и оптимальная стратегия, и среднее время протекания игры при ней. Задача с запретом на пропуски ходов алгоритмически почти тривиальна, поэтому мы сосредоточили наше внимание на варианте игры с возможностью стратегического пропуска хода. Впрочем, основная рекуррентная формула, связывающая среднее время до завершения игры в конкретном состоянии S с временами в состояниях, непосредственно следующих за S, выглядит для двух вариантов правил игры одинаково; разница лишь в том, трактуем ли мы множество пропусков хода как множество позиций, где сделать ход нельзя, либо как множество позиций, где игрок либо не может, либо не хочет сделать ход в соответствии с выбранной им стратегией игры.

Выведем эту формулу.

На каждом ходу возможно 36 различных комбинаций чисел на кубиках. Игрок реагирует на каждую комбинацию в соответствии со стратегией. Если он пропускает ход, то состояние не меняется, среднее время остается прежним. Если он совершает ход, то состояние меняется, и соответствующее ожидаемое время задается значением t для новой конфигурации при нашей стратегии в игре.

Рекуррентная формула выглядит следующим образом:

$$t\left[S\right]=1+\frac{1}{36}\sum_{a,b}t\left[S\left(S,a,b\right)\right], \qquad (1)$$

где $\delta\left(S,a,b\right)$ – выбранное действие: либо множество $\left\{a,b\right\}$, либо $\left\{a+b\right\}$, либо пустое множество. Последняя ситуация имеет место в том случае, если ход невозможен (для варианта игры с запретом на пропуски ходов). Та же самая ситуация для варианта игры с разрешением на пропуск хода имеет место тогда, когда ход либо невозможен, либо нежелателен.

Теорема 1. Если стратегия в конфигурации *S* предписывает ходить ровно в *n* случаях из 36 (неважно, в каком из двух вариантов игры), то

$$t\left[S\right]=\frac{36}{n}+\frac{1}{n}\sum_{\delta(S,a,b)\neq\varnothing}t\left[S\left(S,a,b\right)\right]. \qquad (2)$$

Доказательство начнем с общей формулы (1). Из 36 исходов кубиков ровно *n* соответствуют ненулевым ходам $\delta\left(S,a,b\right)=\varnothing$, а в остальных $36-n$ случаях конфигурация не меняется, и соответствующие слагаемые равны $t\left[S\right]$. Поэтому:

$$t\left[S\right]=1+\frac{1}{36}\left(\left(36-n\right)t\left[S\right]+\sum_{\delta(S,a,b)\neq\varnothing}t\left[S\left(S,a,b\right)\right]\right).$$

Домножая на 36 и перегруппировав члены, получаем:

$$nt\left[S\right]=36+\sum_{\delta(S,a,b)\neq\varnothing}t\left[S\left(S,a,b\right)\right].$$

Поделив на *n*, получаем формулу теоремы. Ч.т.д.

Теорема 1 сводит задачу к динамическому программированию (Аннабаева, 2023): двигаясь от простых состояний к более сложным, мы выбираем действия $\delta\left(S,a,b\right)$, минимизирующие ожидаемое время. Алгоритм вычисления стратегии приведен в приложении 1. Поговорим об этом алгоритме более подробно, возвращаясь к вопросу о стратегическом пропуске хода.

Пример. Пусть остаются фишки {1,6}. Если выпадает дубль (6,6) и игрок закрывает шестерку, то единственным оставшимся ходом будет выпадение (1,1), что в среднем занимает 36 бросков. Если же игрок сознательно пропустит ход, то он будет ждать выпадения одной из комбинаций (1,6), (6,1) или (1,1), что в среднем займет 12 ходов. В трети случаев (выпадение (1,1)) нужно будет дополнительно закрывать шестерку, но для этого подходят целых 6 комбинаций. В среднем на это уйдет еще 6 ходов. Таким образом, общее ожидаемое время составляет около 14 ходов вместо 36 – выигрыш весьма существенный.

Чтобы явным образом учесть в алгоритме стратегические пропуски ходов, мы ниже будем считать, что $\delta\left(S,a,b\right)$ обозначает ход во всех случаях, когда он по правилам возможен, и введем дополнительную функцию $\rho\left(S,a,b\right)\in\left\{0,1\right\}$, определяющую осознанный пропуск хода:

– $\rho\left(S,a,b\right)=1$, если при выпадении $\left(a,b\right)$ игрок делает ход $\delta\left(S,a,b\right)$;

– $\rho\left(S,a,b\right)=0$, если он сознательно пропускает.

Функция $\rho\left(S,a,b\right)$ определяется только для таких комбинаций $\left(S,a,b\right)$, в которых ход по правилам возможен, т. е. в которых по-новому определенная $\delta\left(S,\ a,\ b\right)\neq\varnothing$ (эта $\delta\left(S,\ a,\ b\right)$ соответствует игре с запретом на пропуски ходов). С учетом нововведенных обозначений формула среднего времени принимает следующий вид:

$$t\left[S\right]=\frac{36+\sum_{\delta(S,a,b)\neq\varnothing}\rho\left(S,a,b\right)t\left[S\left(S,a,b\right)\right]}{\sum_{\delta(S,a,b)\neq\varnothing}\rho\left(S,a,b\right)}. \qquad (3)$$

Очевиднейшим алгоритмом, позволяющим минимизировать данный функционал, будет полный перебор по всем возможным реализациям функции ρ. Их, как несложно видеть, 2^{36}. На практике нас интересуют только значения $\rho\left(S,a,b\right)$, для которых $\delta\left(S,a,b\right)\neq\varnothing$, что, конечно, уменьшает перебор, но всё же оставляет нас в классе экспоненциальных алгоритмов (Карпов, Трифонов, 2007).

Однако следующая абстрактная теорема из теории оптимального управления позволяет свести решение к полиномиальному времени.

Теорема 2 (о монотонности). Пусть заданы положительные числа $\left(a_1,\ldots,a_n\right)$. Тогда минимум функционала

$$f\left(p_1,\ldots,p_n\right)=\frac{K+\sum_{k=1}^{n}a_kp_k}{\sum_{k=1}^{n}p_k},\quad p_k\in\left[0,1\right] \qquad (4)$$

достигается на пороговой комбинации

$$p_k=\left[a_k\leq L\right]$$

для некоторого порога *L*.

Доказательство теоремы проведем в несколько этапов.

Лемма 1 (минимум при фиксированной сумме). Упорядочим числа $a_k: a_1\leq a_2\leq\ldots\leq a_n$.

Среди всех наборов с фиксированной суммой

$$P \quad \sum p$$

минимальное значение функционала достигается на «жадной» комбинации:

$$p_k=\begin{cases}1, k\leq P\\ \left\{P\right\}, P<k<P+1,\\ 0, k\geq P+1\end{cases}$$

где {*P*} – дробная часть числа *P*.

Доказательство Леммы 1. Пусть минимум достигается на каком-то наборе, отличном от указанного. Тогда найдем первое k, где $p_k < 1$. Так как сумма равна P, существует индекс $l > k$ с $p_l > 0$. Увеличим p_k на величину $d = \min(1 - p_k, p_l)$ и одновременно уменьшим p_l на d. Тогда значение функционала изменится на

$$d \cdot (a_k - a_l).$$

Так как $a_1 \ge a_k$, это изменение не увеличивает значение функционала. А значит, у нас либо получилось уменьшить целевое значение (противоречие), либо за конечное число шагов мы сумеем достичь комбинации, удовлетворяющей условию «жадности».

Введем полезную функцию g:

$$g(P) = \min_{\sum p_k = P} f(p_1, \ldots, p_k). \quad (5)$$

Лемма 2 (зависимость g от P). Для целого $P = m$:

$$g(m) = \frac{K + \sum_{k=1}^{m} a_k}{m}.$$

Для $P = m + x, \ \ x \in [0,1]$:

$$g(m+x) = \frac{K + \sum_{k=1}^{m} a_k + a_{m+1}x}{m+x} = a_{m+1} + \frac{A_m - m a_{m+1}}{m+x},$$

$$A_m = K + \sum_{k=1}^{m} a_k.$$

На каждом отрезке $[m, \ m+1]$ функция g монотонна: убывает, возрастает или постоянна в зависимости от знака $A_m - m a_{m+1}$.

Лемма 3 (о продолжении монотонного возрастания). Если функционал g возрастает на отрезке $[P, P+1]$, то он продолжит возрастать на всех следующих отрезках.

Действительно, для $[B, B+1]$

$$g(B) = \frac{A'}{B}, \quad A' \le A + a_{B+1}(B - P).$$

Тогда

$$A' - B a_{B+1} \le A + (B - P) a_{B+1} - B a_{B+1} =$$
$$= A - P a_{B+1} \le A - P a_{P+1} < 0.$$

А это и является критерием возрастания функции на $[B, B+1]$.

Следствие. Функция g либо чисто монотонная, либо унимодальна: сначала убывает, затем возрастает. Минимум достигается при некотором целом P. И тогда порог $L = a_P$. отвечает условию теоремы 1. При этом оптимальная комбинация соответствует первым P наименьшим a_k.

Из теоремы следует жадный алгоритм (Новиков, Поздняков, 2005): сортируем значения по возрастанию, перебираем префиксы и выбираем тот, где дробь минимальна. Подробный псевдокод алгоритма приведен в приложении 2.

Математические ожидания, полученные с помощью приведенных алгоритмов[5]:

1) для случайной стратегии ~ 52,33;

2) для оптимальной стратегии без пропусков ~ 48,31;

3) для оптимальной стратегии с пропусками ~ 46,56.

С помощью компьютерных вычислений нами получено полное описание идеального хода для любой ситуации в игре, однако это описание не подходит для практического применения ввиду очень большого объема информации для запоминания. Поэтому нами была предложена (угадана) простая стратегия, которая дает примерно схожее с оптимальным математическое ожидание. Стратегия состоит всего из двух пунктов:

1) если мы можем закрыть и сумму, и пару, то закрываем сумму, если она больше или равна 8, в противном случае закрываем пару;

2) если после хода из чисел от 1 до 6 останутся только 1 или только 2, пропускаем ход.

Во всех остальных случаях, когда можем закрыть ровно 1 вариант и правило 2 не выполняется, просто закрываем этот вариант.

При всей простоте стратегия дает весьма неплохой результат: среднее время ожидания закрытия всей доски для игрока, который ее придерживается, ≈46,91, что очень слабо отличается от оптимальной стратегии.

Вычисление вероятностей для пары стратегий

Мы уже нашли среднее время до выигрыша для всех интересующих нас стратегий, однако оно не дает полного представления, насколько одна стратегия лучше другой. Как часто игрок, закрывающий доску в среднем за 48 ходов, будет побеждать игрока, заканчивающего за 52 хода? На самом деле соотношение побед и поражений будет не 52 к 48, а заметно лучше. Поэтому дополнительная наша задача – выяснить, как найти эту вероятность.

Постановка задачи: пусть x_1 и x_2 – стратегии, и нам нужно определить, с какой вероятностью игрок, следующий стратегии x_1, закон-

[5] Исходный код для вычисления ожидаемых времен игры, а также полные таблицы для каждого из возможных состояний доступны по ссылке https://github.com/heni/GameShastun

чит раньше игрока, следующего стратегии x_2, и наоборот, а также вероятность ничьей.

Вероятность ничьей будем считать из следующих соображений: ничья случается, если оба игрока закончили игру за одинаковое число ходов (за 1 ход – что невозможно, за 2 хода – что тоже невозможно, … за 10 ходов – что уже происходит с положительной вероятностью, … за n ходов и так далее до бесконечности – здесь нет предела, теоретически любая стратегия в применении может реализоваться за сколь угодно большое время). Итоговая вероятность представляет собой сумму бесконечного ряда, где каждый член – произведение вероятностей закончить на k-м шаге для каждой из двух стратегий. Вероятность победы каждого участника мы найдем схожим образом: это сумма ряда, где члены представляют собой произведение вероятностей того, что первый закончит ровно за k шагов, а второй более чем за k шагов, по всем k.

Всё, что осталось научиться находить, – это распределение вероятности для произвольной стратегии, чтобы точно знать вероятность победить на k-м шаге.

Чтобы посчитать эту вероятность, сведем игру по стратегии к случайному процессу. Вектор состояния случайного процесса (4096-мерный) будет в позиции i содержать вероятность нахождения в соответствующей позиции игры. Для изначального положения этот вектор содержит все нули, кроме единицы в позиции 4096 (все фишки открыты).

Введем также матрицу стратегии – это квадратная таблица 4096 на 4096, в элементах i, j которой находятся вероятности перехода из позиции с номером i в позицию с номером j при следовании этой стратегии за один ход (таких переходов немного и матрица сильно разрежена). Можно заметить, что сумма каждой строки матрицы равна 1 – такие матрицы называются стохастическими (Альпин, Альпина, 2012).

Для нас полезным фактом является то, что данная матрица позволяет просто рассчитывать состояния процесса после k итераций.

Если s_0 – вектор состояния процесса в момент времени 0, s_k – вектор состояния процесса в момент времени k, M – стохастическая матрица процесса, то

$$s_k = s_0 M^k.$$

Для вычисления вероятностей победы/поражения нужно уметь вычислять значения вероятности закончить игру в ход с номером k при следовании определенной стратегии. Вполне очевидно, что если в ход с номером k вероятность оказаться в позиции 0 (конечной позиции) была p_1, а после этого хода стала p_2, значит, вероятность закончить игру ровно в этот ход – $p_2 - p_1$.

Вычислительный эксперимент показывает, что вектор состояния s достаточно быстро сходится к естественному стационарному положению (с вероятностью 1 мы находимся в состоянии 0 – все фишки закрыты).

На практике 1000 итераций уже достаточно, чтобы вероятность незавершения игры была меньше 10^{-13}, чего достаточно для точного вычисления 10 значащих цифр вероятностей победы/поражения. Результаты вычисления приведены в таблице.

В процессе исследования возможностей для ускорения вычислений помимо применения классических подходов к перемножению вектора на разреженную матрицу нам удалось обнаружить, что и сам вектор состояния в процессе вычислений с некоторого момента приобретает некоторое количество нулей, которые сохраняются при любом дальнейшем числе итераций. Это может послужить дополнительным источником оптимизации. Поэтому далее будут рассмотрены особенности изменения данного вектора.

Каждая позиция (j) (обозначающая вероятность нахождения в позиции под номером j) вектора s_k вычисляется по формуле:

$$p_j = \bar{p}_1 M_{1,j} + \bar{p}_2 M_{2,j} + \ldots + \bar{p}_{4096} M_{4096,j}.$$

Здесь слева – значение позиции (j) на следующей итерации вектора, а справа – значения на текущей итерации этого вектора.

Назовем позицию вектора (j) зависимой от позиции (i), если элемент матрицы $M_{i,i}$ ненулевой. Зависимость обозначает непосредственное влияние на значение позиции на следующем шаге.

Позиция (i) косвенно зависит от (n), если существует набор индексов $i, j, k, \ldots, m, n$ такой, что (i) зависит от (j), (j) зависит от (k),…, (m) зависит от (n).

Если же $M_{i,i} \neq 0$, тогда позиция (i) – самозависимая. Самозависимые позиции обладают следующим свойством – если они принимают ненулевое (даже положительное, так как это вероятность) значение, они уже не примут нулевое значение ни на какой итерации. Так

Таблица

Вероятности для пар стратегий

Стратегия	Оптимальная с пропусками	«Простая» стратегия	Оптимальная без пропусков	Случайный выбор
Оптимальная с пропусками	49,2382 %	49,7463 %	51,0983 %	55,5169 %
	1,5237 %	1,5014 %	1,4627 %	1,3748 %
	49,2382 %	48,7523 %	47,4389 %	43,1083 %
«Простая» стратегия		49,2596 %	50,6160 %	55,0196 %
		1,4808 %	1,4438 %	1,3600 %
		49,2596 %	47,9402 %	43,6204 %
Оптимальная без пропусков			49,2953 %	53,6568 %
			1,4093 %	1,3321 %
			49,2953 %	45,0111 %
Случайный выбор				49,3607 %
				1,2786 %
				49,3607 %

как их значение на следующей итерации – сумма его текущего положительного значения с положительным коэффициентом $M_{i,i}$ и еще нескольких неотрицательных слагаемых.

Позиции, которые, начиная с некоторой итерации, никогда не принимают нулевое значение, назовем необнуляемыми (они не обязательно должны быть самозависимые). Позиции, которые после некоторой итерации всегда будут равны нулю, – обнуляемыми. Стоит отметить, что промежуточных позиций, которые через сколь угодно много итераций могут принимать и нулевое и ненулевое значение, в векторе существовать не может, это будет показано позже.

Теперь же сформулируем несколько свойств на основе введенных обозначений.

Свойство 1. Позиция, которая косвенно зависит от необнуляемой, – необнуляемая.

Доказательство. Сначала докажем это свойство для прямой зависимости. Если позиция прямо зависит от необнуляемой, значит в сумме, определяющей ее значение на следующей итерации, всегда будет ненулевой элемент, а значит, она тоже необнуляемая. Если учесть, что косвенная зависимость – это конечная цепочка прямых зависимостей, свойство доказано.

Свойство 2. Несамозависимая позиция, не зависимая косвенно от необнуляемых, – обнуляемая.

Доказательство. Поскольку такая позиция зависит прямо только от обнуляемых позиций, существует итерация, на которой они все уже обратятся в 0. А так как ее значение определяется только суммой их значений и она дополнительно не зависит от себя, то она тоже обнулится.

Из этих двух свойств уже следует, что все позиции либо обнуляемые, либо необнуляемые. Чтобы показать это, мы будем постепенно выделять группы позиций, для которых мы точно знаем их обнуляемость. Если позиция косвенно зависит от необнуляемой, то она необнуляемая по свойству 1. Далее рассматриваем косвенно независимые от необнуляемой позиции. Если позиция не является самозависимой, она обнуляемая по свойству 2. Далее позиции еще и самозависимые. Если такая позиция хоть раз примет положительное значение, она останется положительной сколь угодно долго, значит, она необнуляемая. Если же она ни на какой итерации не примет положительное значение, она обнуляемая.

Понять, какие позиции вектора обнуляемые, а какие нет, можно только анализируя матрицу, поскольку она полностью определяет эти свойства. Однако мы предлагаем следующий простой способ, привлекающий совсем небольшие вычислительные затраты, учитывая приведенное ниже свойство.

Свойство 3. Максимальная длина цепочки зависимостей в косвенной зависимости – 12. Длина – число отношений зависимости, а не позиций в цепочке.

Доказательство. Вспоминаем, что позиция в нашем векторе представляет состояние игры. Вероятность оказаться в текущем состоянии игры может зависеть только от вероятности быть в этом же состоянии игры на прошлом ходу (самозависимость) и вероятностей находиться в состояниях с меньшим числом перевернутых фишек на прошлом ходу. Так как фишек у нас всего 12, значит, максимальная длина цепочки зависимостей тоже 12.

Теперь же переходим к способу поиска обнуляемых позиций. Мы предлагаем просто вычислить вектор s_{12} и посмотреть, какие позиции равны 0. Из свойств 1 и 3 мы можем понять, что в s_{12} все необнуляемые позиции нашего вектора успеют принять ненулевое значение. Все обнуляемые позиции, в свою очередь, гарантированно обнуляются через ход после того, как все позиции, от которых они зависят, обнулились (следствие свойства 2). После первого хода обнулится позиция с нулем перевернутых фишек, если она обнуляемая. После второго хода обнулятся все обнуляемые позиции с 1 перевернутой фишкой, так как они зависят максимум от позиции с 0 перевернутых фишек, и т. д. Следовательно, все обнуляемые элементы (кроме, пожалуй, позиции, обозначающей конец игры, но она очевидно необнуляемая из-за самозависимости) успеют обнулиться.

Проведенный после этого численный эксперимент приводит к интересным результатам: он показывает, что оптимальная стратегия весьма уникальна и разительно отличается от всех остальных по числу обнуляемых позиций: в то время как у всех остальных их число не превышает 10 % от общего числа, в оптимальной стратегии таких позиций 1682 из 4096, около 41 %! Помимо того, что оптимальную стратегию считать проще остальных, мы получаем очень красивый факт: оптимальный игрок, в отличие от всех остальных рассмотренных, уже через 12 ходов после начала игры гарантированно не окажется почти в половине всех возможных игровых ситуаций, так как тонко чувствует небольшие потери от перехода в «плохие» позиции.

На этом мы завершаем разбор игры, однако перед этим сделаем еще одно важное замечание, точнее, вернемся к уже озвученной в начале статьи сложной проблеме: когда мы говорили об оптимальной стратегии, мы говорили о стратегии с наименьшим средним ожиданием времени закрытия всех фишек. Однако оптимальна ли такая стратегия против любой другой стратегии по критерию вероятности победы в игре? Согласно общим теоремам существования равновесия Нэша и его аналогов в динамической постановке игры, должна существовать стратегия, не проигрывающая по вероятности победы всем остальным стратегиям (и, как следствие, наилучшая в ответ на применение копии себя же). Как ее найти?

Этот интереснейший и сложный вопрос мы оставляем для будущих исследований (или кто-то из читателей статьи его разрешит?).

Авторский вклад

Авторы внесли равный вклад в работу, представленную в статье.

Конфликт интересов

Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

Contribution of the authors

The authors contributed equally to this article.

Conflict of Interest

The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

Библиографический список

1. Альпин, Ю. А., & Альпина, В. С. (2012). О нормальной форме стохастической матрицы. *Ученые записки Казанского университета. Серия: Физико-математические науки*, (2), 60–72. [Alpin, Yu. A., & Alpina, V. S. (2012). On the normal form of a stochastic matrix. *Proceedings of Kazan University. Physics and Mathematics Series*, (2), 60–72. (In Russian).]

2. Аннабаева, Н. Р. (2023). Динамическое программирование. *Символ науки*, (10–2), 12–13. [Annabaeva, N. R. (2023). Dynamic programming. *Symbol of Science*, (10–2), 12–13. (In Russian).]

3. Карпов, Ю. Г., & Трифонов, П. В. (2007). Сложность алгоритмов и программ. *Компьютерные инструменты в образовании*, (6), 3–10. [Karpov, Yu. G., & Trifonov, P. V. (2007). Complexity of algorithms and programs. *Computer Tools in Education*, (6), 3–10. (In Russian).]

4. Новиков, Ф. А., & Поздняков, С. Н. (2005). Жадные алгоритмы. *Компьютерные инструменты в образовании*, (2), 49–58. [Novikov, F. A., & Pozdnyakov, S. N. (2005). Greedy Algorithms. *Computer Tools in Education*, (2), 49–58. (In Russian).]

5. Ambrona, M. (2022). An Efficient Algorithm for Chess Unwinnability. In: *11th International Conference on Fun with Algorithms (FUN 2022). Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

6. Aravind, N. R., Misra, N., & Mittal, H. (2022). Chess is hard even for a single player. In: *11th International Conference on Fun with Algorithms (FUN 2022). Leibniz International Proceedings in Informatics (LIPIcs).* Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

7. Bagan, G., Duchene, E., Galliot, F., Gledel, V., Mikalacki, M., Oijid, N., Parreau, A., & Stojakovic, M. (2024). Poset Positional Games. In: *12th International Conference on Fun with Algorithms (FUN 2024)*. *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

8. Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific.

9. Bouton, C. L. (1901). Nim, a game with a complete mathematical theory. *Annals of Mathematics, 3*(1/4), 35–39.

10. Burke, K. W., Ferland, M., & Teng, S. H. (2022). Quantum-Inspired Combinatorial Games: Algorithms and Complexity. In: *11th International Conference on Fun with Algorithms (FUN 2022)*. *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

11. Burke, K. W., Ferland, M., & Teng, S. H. (2024). Nimber-Preserving Reduction: Game Secrets and Homomorphic Sprague-Grundy Theorem. *Theoretical Computer Science*, (1005), 114636. DOI: 10.1016/j.tcs.2024.114636

12. Churchill, A., Biderman, S., & Herrick, A. (2019). Magic: the Gathering is Turing Complete. In: *FUN 2019 (Formal Underpinnings of Computing and Communication)*. Leibniz International Proceedings in Informatics (LIPIcs).

13. Crombez, L., da Fonseca, G. D., & Gerard, Y. (2020). Efficient Algorithms for Battleship. In: *10th International Conference on Fun with Algorithms (FUN 2020). Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

14. Dailly, A., Lafourcade P., & Marcadet, G. (2024). How did they design this game? Swish: complexity and unplayable positions. In: *12th International Conference on Fun with Algorithms (FUN 2024). Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

15. Ferguson, T. S. (1989). Who solved the secretary problem? *Statistical Science, 4*(3), 282–289. DOI: 10.1214/ss/1177012493

16. Grundy, P. M. (1939). Mathematics and games. *Eureka*, (2), 6–8.

17. Hearn, R. A., & Demaine, E. D. (2009). *Games, Puzzles, and Computation*. Wellesley, MA: A K Peters, Ltd.

18. Iburi, Y., & Uehara, R. (2024). Computational Complexity of Matching Match Puzzle. In: *12th International Conference on Fun with Algorithms (FUN 2024)*. *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

19. Nesterov, A., & Sveridov, O. (2025). *Smarties and smartypants: overconfidence in a TV competition for young talents*. St. Petersburg, Russia: Department of Economics and Game Theory Lab, Higher School of Economics.

20. Puterman, M. L. (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

21. Rin, B., & Schipper, A. (2024). Arimaa is PSPACE-hard. In: *12th International Conference on Fun with Algorithms (FUN 2024)*. *Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

22. Shapley, L. S. (1953). Stochastic Games. *Proceedings of the National Academy of Sciences of the United States of America*, (39), 1095–1100.

23. Sprague, R. (1935). Über mathematische Kampfspiele. *Tohoku Mathematical Journal*, (41), 438–444.

24. Subercaseaux, B., & Lokshtanov, D. (2022). Wordle is NP-hard. In: *11th International Conference on Fun with Algorithms (FUN 2022). Leibniz International Proceedings in Informatics (LIPIcs)*. Schloss Dagstuhl – Leibniz-Zentrum fur Informatik.

25. Von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.

Крохалев Евгений Михайлович, ООО «СофтКом», Санкт-Петербург, Российская Федерация

E-mail: eugene.krokhalev@gmail.com

ORCID ID: 0009-0008-2584-7212

Evgeniy M. Krokhalyov, SoftCom, Ltd, Saint-Petersburg, Russian Federation

E-mail: eugene.krokhalev@gmail.com

ORCID ID: 0009-0008-2584-7212

Попов Виталий Сергеевич, студент, Воронежский государственный университет, Воронеж, Российская Федерация
E-mail: vitaly.popov.res@gmail.com
ORCID ID: 0009-0005-5859-3990

Савватеев Алексей Владимирович, д-р физ.-мат. наук, член-корреспондент, Российская академия наук; профессор, Адыгейский государственный университет; профессор, Московский физико-технический институт; профессор, Университет Иннополис; главный научный сотрудник, Центральный экономико-математический институт Российской академии наук, Москва, Российская Федерация
E-mail: hibiny@mail.ru
ORCID ID: 0000-0002-6942-2282

Получена 08.10.2025
Получена в доработанном виде 28.10.2025
Одобрена 31.10.2025

Vitaliy S. Popov, Student, Voronezh State University, Voronezh, Russian Federation
E-mail: vitaly.popov.res@gmail.com
ORCID ID: 0009-0005-5859-3990

Aleksey V. Savvateev, Dr. Sci. (Phys.-Math.), Corresponding Member of the Russian Academy of Sciences; Professor, Adyghe State University; Professor, Moscow Institute of Physics and Technology; Professor, Innopolis University; Chief Research Fellow, Central Economics and Mathematics Institute of the Russian Academy of Sciences, Moscow, Russian Federation
E-mail: hibiny@mail.ru
ORCID ID: 0000-0002-6942-2282

Received 08.10.2025
Revisited 28.10.2025
Accepted 31.10.2025

Приложение 1

Псевдокод динамического программирования для вычисления $\delta(S,a,b)$

Задача: для каждого состояния S и исхода кубиков (a,b) выбрать действие $\delta(S,a,b)$, минимизирующее ожидаемое время до победы.

```
function compute_strategy():
     # всего 2^12 состояний (каждая из 12 фишек открыта/закрыта)
     for each state S in order of increasing |S|:
          if S = ∅:
               t[S] = 0
               continue
          # перебираем все 36 исходов кубиков
          for each (a, b) in dice_pairs:
               options = possible_moves(S, a, b) # {a+b}, {a,b}
          если они возможны

               best_value = ∞
               best_action = ∅

               # выбираем действие с минимальным t
               for move in options:
                    new_state = S \ move
                    candidate = t[new_state]
                    if candidate < best_value:
                         best_value = candidate
                         best_action = move
               δ[S, a, b] = best_action

          # вычисляем t[S] по теореме 1
          n = number of (a,b) with δ[S,a,b] ≠ ∅
          t[S] = (36 + sum(t[S \ δ[S,a,b]] over δ[S,a,b] ≠ ∅)) / n
     return δ, t
```

Приложение 2

Псевдокод жадного алгоритма для $\delta(S,a,b)$

```
function compute_rho(S, δ, t):
     candidates = [] # список (t_next, (a,b))

     for each (a, b) in dice_pairs:
          if δ(S,a,b) ≠ ∅:
               next_state = S \ δ(S,a,b)
               candidates.append((t[next_state], (a,b)))
     if candidates is empty:
          return map_all_pairs_to_zero()

     sort(candidates by t_next ascending)

     best_value = ∞
     best_m = 1
     prefix_sum = 0

     for m from 1 to length(candidates):
          prefix_sum += candidates[m].t_next
          candidate = (36 + prefix_sum) / m
          if candidate < best_value:
               best_value = candidate
               best_m = m
     rho = {}
     for i, (val, pair) in enumerate(candidates):
          rho[pair] = 1 if i < best_m else 0
     return rho
```