



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

КАФЕДРА ИУ-7 «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
НА ТЕМУ:**

***«Моделирование процессов цифрового
документооборота с помощью сетей Петри»***

Студент ИУ7-85Б

Богаченко А. Е.

Руководитель

Строганов Ю. В.

Нормоконтроллер

2024 г.

РЕФЕРАТ

Расчетно-пояснительная записка 43 с., 24 рис., 1 табл., 20 ист.

КЛЮЧЕВЫЕ СЛОВА

сети Петри, цифровой документооборот, квантовые шахматы, квантовое моделирование

В работе представлен анализ применения стохастических временных сетей Петри для моделирования процессов цифрового документооборота.

Рассмотрены существующие системы цифрового документооборота. Рассмотрена классификация сетей Петри, составлена и проанализирована модель процесса утверждения темы для выпускной квалификационной работы. Составлен вывод на основе полученных результатов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	9
1 Аналитическая часть	10
1.1 Электронный документ	10
1.2 Постановка задачи	10
1.3 Моделирование с помощью сетей Петри	12
1.3.1 Описание поведения системы	12
1.3.2 Раскрашенная сеть Петри	15
1.3.3 Раскрашенная сеть Петри с временным механизмом	15
1.3.4 Стохастические сети Петри	15
1.3.5 Анализ стохастической сети Петри	17
1.4 Квантовое моделирование	17
1.4.1 Фазовое пространство	18
1.4.2 Квантовые шахматы	19
1.4.3 Стандартный ход	20
1.4.4 Второй шаг при суперпозиции	21
1.4.5 Вывод	21
2 Конструкторская часть	22
2.1 Проектирование системы	22
2.2 Проектирование алгоритма с использованием sTPN	22
2.3 Проектирование алгоритма с использованием квантового моделирования	23
2.4 Схема алгоритма расчёта вероятностей	24
3 Технологическая часть	25
3.1 Реализация алгоритма моделирования с помощью sTPN	25
3.2 Сопоставление BPMN 2.0 и сетей Петри	25
3.3 Реализация алгоритма квантового моделирования	25
3.3.1 Выбор и обоснование средств разработки	25
3.3.2 Листинги кода	26
4 Исследовательская часть	29
4.1 Моделирование с помощью sTPN	29
4.2 Квантовое моделирование	33
4.3 Вывод	37
ЗАКЛЮЧЕНИЕ	39
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	40
ПРИЛОЖЕНИЕ А	42

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

- 1) СЭД (система электронного документооборота) – программное обеспечение для работы с электронными документами на всех стадиях их жизненного цикла: создание, редактирование, хранение;
- 2) ЭДО (электронный документооборот) – передача электронных документов по информационно-телекоммуникационным сетям или обработка в информационных системах.

ВВЕДЕНИЕ

Развитие области электронного документооборота (ЭДО) является одной из приоритетных и актуальных задач цифровизации государственных и частных секторов, об этом свидетельствует ежегодно увеличивающийся на 15% объём электронного документооборота[1].

Системы электронного документооборота (СЭД) должны отвечать высоким стандартам в области безопасности, эффективности и надёжности, где критическим показателем является своевременное выполнение процессов[1]. В связи с этим основное внимание в области ЭДО уделяется контролю процессов управления, нежели управлению потоком данных[1]. СЭД должна регулярно проходить проверки на соответствие стандартам и требованиям проектирования соответствующих систем[1].

Целью данной работы является анализ существующих методов моделирования процессов цифрового документооборота, включая сети Петри и квантовое моделирование, а также разработка системы, способной проводить данное моделирование. Для достижения поставленной цели необходимо решить следующие задачи:

- проанализировать и проклассифицировать методы моделирования процессов цифрового документооборота с использованием различных сетей Петри;
- проанализировать и проклассифицировать методы квантового моделирования процессов цифрового документооборота;
- сформулировать общие требования к разрабатываемой системе;
- разработать алгоритмы, позволяющие промоделировать процессы цифрового документооборота;
- промоделировать процессы цифрового документооборота с использованием разработанного программного обеспечения;
- проанализировать полученные результаты.

1 Аналитическая часть

1.1 Электронный документ

Электронный документ – это документ, информация которого представлена в электронной форме. Юридическая значимость такого документа может быть получена с помощью электронной подписи, визуальное отображение подписи в документе не требуется[1]. Для электронного документа характерно следующее:

- аутентичность – свойство электронного документа, гарантирующее, что электронный документ идентичен заявленному;
- достоверность – свойство электронного документа, при котором содержание является полным и точным представлением подтверждаемых операций, деятельности или фактов и которому можно доверять в последующих операциях или в последующей деятельности;
- целостность – состояние электронного документа, в который после его создания не вносились никакие изменения;
- пригодность для использования – свойство электронного документа, позволяющее его локализовать и воспроизвести в любой момент времени.

1.2 Постановка задачи

Объём административных процедур, требующих документального подтверждения, ежегодно увеличивается примерно на 15%[1]. Бумажный документооборот обладает следующими критическими проблемами:

- практически для каждой административной процедуры имеется несколько шаблонов документов, что в дальнейшем затрудняет классификацию, хранение и поиск необходимого документа;
- в зависимости от типа документа срок его хранения может составлять от нескольких минут до 75 лет[1]. В то время как документооборот постоянно увеличивается, стоимость организации и ведения архива увеличивается примерно на 30% ежегодно. Процедуры поиска и классификации документа могут занимать от нескольких часов до нескольких дней;
- важным аспектом является защита конфиденциальных данных, поскольку физические архивы более подвержены риску компрометации критической информации и персональных данных из-за человеческого фактора;
- в зависимости от количества контрагентов, необходимых для проведения процедуры, и их территориального расположения время, затрачиваемое

на оформление процессов бумажного документооборота, может быть существенно увеличено. Ежегодная доля документов, в которых участвуют межрегиональные или межгосударственные ведомства, составляет 25%. Наглядный пример такого бизнес-процесса представлен на рис. 1. Отказ одного из контрагентов имеет решающее значение при межгосударственном бумажном оформлении, поскольку перезапускает весь процесс.

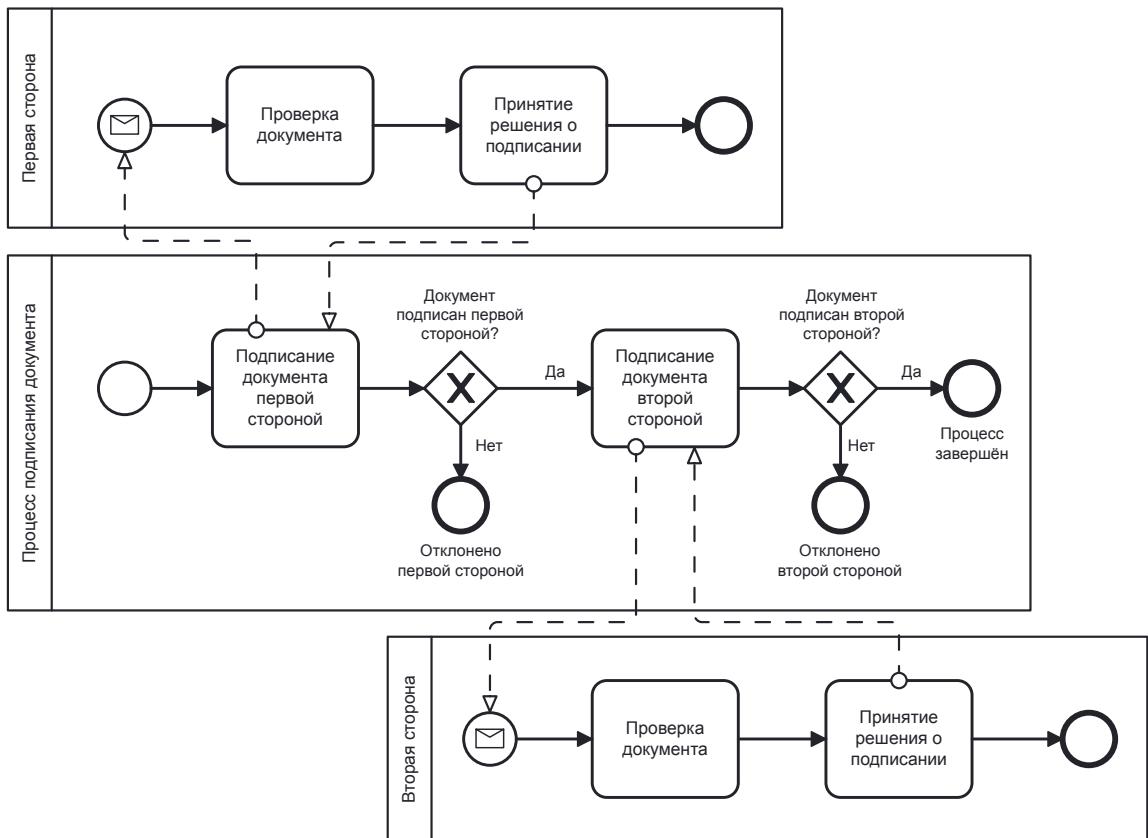


Рисунок 1 – Модель бизнес-процесса подписания документа

Данные недостатки приводят к развитию сферы цифрового документооборота[1], как одной из основных приоритетных задач цифровизации государственного и частного сектора.

Для обработки большого объёма операций с электронными документами между многочисленными, в том числе межгосударственными контрагентами, были разработаны системы электронного документооборота (СЭД). СЭД должна соответствовать современным стандартам безопасности, эффективности и надёжности[1], где ключевым требованием является своевременное выполнение процессов и отказоустойчивость системы. Современные СЭД ориентированы на управление процессами, а не на управление потоками данных[1], поскольку абстрактная структура документа практически не влияет ни на одну из

критических функций[1], но переходы состояний и их порядок сильно влияют на каждый аспект такой системы.

С точки зрения управления цифровыми документами, управление временем включает в себя следующее: планирование рабочего процесса, оценку продолжительности выполнения, выполнения процесса при соблюдении всех заданных временных ограничений[2]. Поэтому моделирование таких рабочих процессов документооборота с ограничениями по времени для поиска оптимального обхода состояний является сложной задачей[1].

1.3 Моделирование с помощью сетей Петри

Сеть Петри представляет собой модель двудольного графа[1], состоящую из двух классов узлов, мест и переходов. Места соединяются переходами с помощью дуг и могут содержать фишку, маркировка (текущее состояние) задаётся количеством фишек.

Переходы – это действия, которые могут происходить при изменении состояния системы и могут срабатывать только в том случае, если они включены (должны быть выполнены все предварительные условия). Когда переход срабатывает, он переносит токены из своих входных мест в свои выходные места. Количество передаваемых токенов зависит от мощности каждой дуги. Математически сеть Петри определяется как кортеж:

$$N = (P, T, I, O, M_0), \quad (1)$$

где $P = \{p_1, p_2, \dots, p_m\}$ множество мест, $T = \{t_1, t_2, \dots, t_n\}$ множество переходов, $I \subset P \times T$ множество входящих дуг, $O \subset T \times P$ множество исходящих дуг, $M_0 = (m_{01}, m_{02}, \dots, m_{0m})$ начальная разметка.

1.3.1 Описание поведения системы

В качестве математического инструмента сеть Петри можно использовать для создания различных алгебраических уравнений и математических моделей, описывающих поведение системы.

Рассмотрим четыре наиболее распространённых поведения системы[1]:

- 1) последовательное выполнение;
- 2) параллельное выполнение;
- 3) конкуренция;
- 4) синхронизация.

Последовательное выполнение в терминах сетей Петри можно смоделировать, как показано на рисунке 2. Каждый переход срабатывает один за другим.

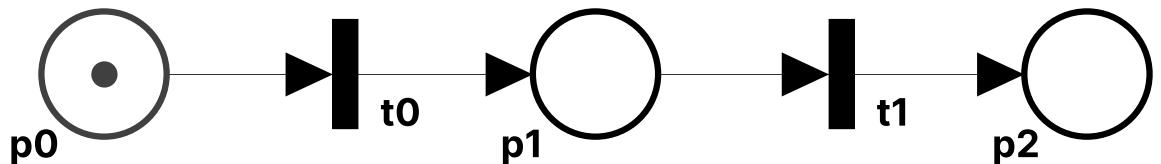


Рисунок 2 – Модель последовательного выполнения

Параллельное выполнение можно смоделировать, как показано на рисунке 3. Имеется три параллельных перехода, каждый из которых после получения фишки включается и может сработать.

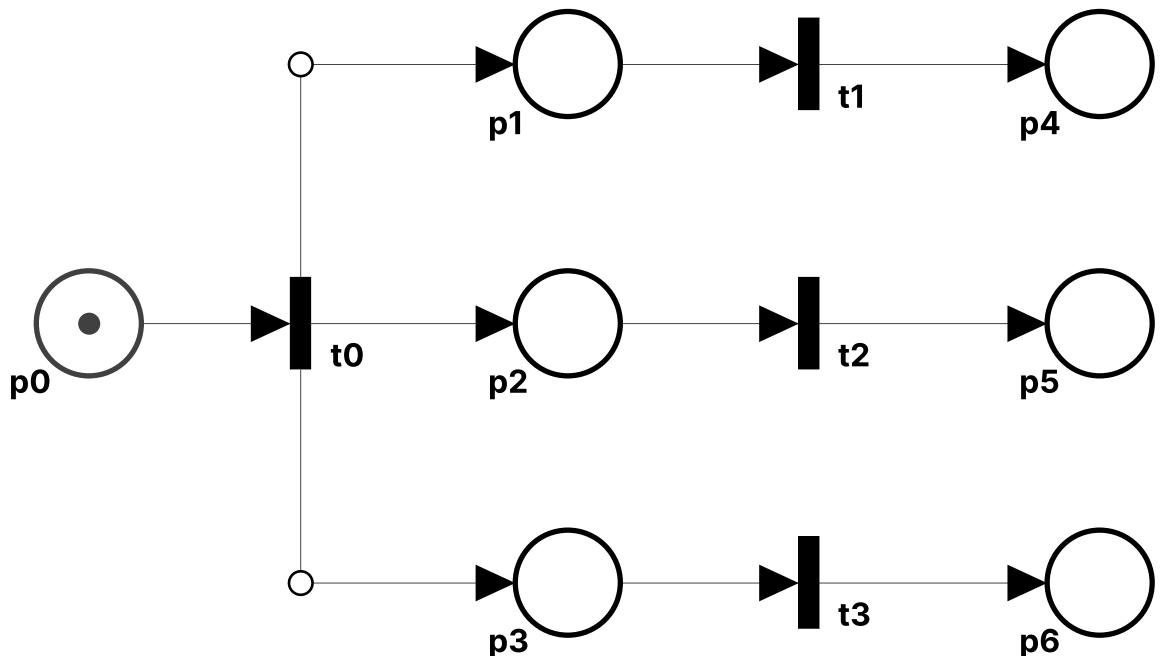


Рисунок 3 – Модель параллельного выполнения

Конкурентное поведение, также известное как принятие решений, можно смоделировать, как показано на рис. 4. Все три перехода активны и могут сработать, но сработает только один, оставив двух других перехода в отключённом состоянии. Если переход связан со временем срабатывания, будет выбран переход с минимальным временем, необходимым для срабатывания.

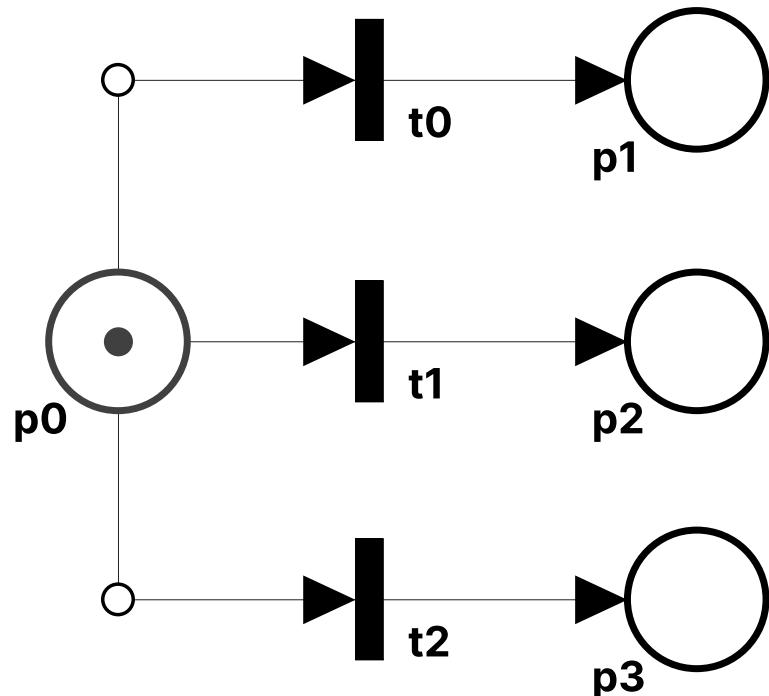


Рисунок 4 – Модель конкурентного поведения

Синхронизацию можно смоделировать как показано на рис. 5, последний переход t_2 будет активирован только в том случае, если в соединённых позициях p_2 и p_3 есть фишки.

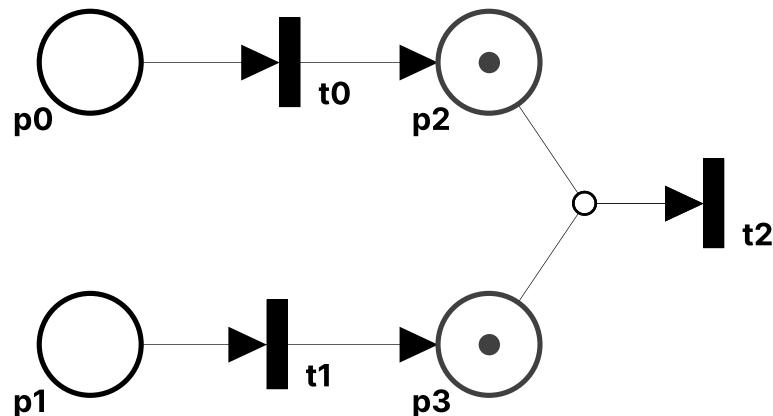


Рисунок 5 – Модель синхронизации системы

1.3.2 Раскрашенная сеть Петри

Раскрашенная сеть Петри – это кортеж, состоящий из восьми элементов:

$$CPN = \langle P, I, T, G, A, E, Z, C \rangle . \quad (2)$$

1. P – конечное множество позиций. С каждой позицией может быть связана определённая маркировка, которая учитывает наличие в данной позиции различных типов ресурсов. Маркировка позиции $p \in P$ представляет собой мульти множество, например, следующего вида: $m(p) = (1'r, 2'b, 1'g)$. Здесь r, b, g – переменные указанных цветовых типов, определяющие различные виды ресурсов, а цифры, стоящие перед апострофами, – количество фишек соответствующего типа в позиции $p \in P$.

2. $I(p)$ – функция инициализации сети CPN.
3. – конечное множество переходов.
4. G – блокировочная функция, описывающая дополнительные условия, которые должны быть выполнены для срабатывания перехода $t \in T$.
5. A – конечное множество ветвей, связывающих между собой позиции и переходы.
6. $E(a)$ – функция, задающая выражения на ветвях $a \in A$.
7. Z – конечное множество непустых типов, называемое цветами.
8. $C(p)$ – цветовая функция, определяющая множество типов цветов, разрешённых для каждой позиции.

1.3.3 Раскрашенная сеть Петри с временным механизмом

Существует ряд задач моделирования, в которых необходимо учитывать не только последовательность событий, но и время их наступления, а также продолжительность. Для этой цели предусмотрено расширение возможностей раскрашенных сетей Петри путём введения временного механизма – Timed CPN (tCPN):

$$tCPN = \langle P, I, T, G, A, E, Z, C, \tau, \Delta t \rangle . \quad (3)$$

В модель системы вводятся часы, показывающие глобальное время τ .

1.3.4 Стохастические сети Петри

Стохастическая временная сеть Петри – Stochastic Timed CPN (sTPN) расширена за счёт переходов по времени срабатывания[3], как детерминированных, так и стохастических[1]. Этот тип сетей широко используется[4], так

как большинство реальных процессов лучше описываются вероятностными моделями. Математически *sTPN* определена как кортеж:

$$sTPN = (P, T, I, O, M_0, \Lambda). \quad (4)$$

1. (P, T, I, O, M_0) сеть Петри (1).
2. $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_3)$, где λ_i случайное распределение времени срабатывания перехода.

Случайное распределение времени можно обозначить[8] как вероятность времени срабатывания:

$$\lambda_i(x) = P\{t_i < x\}. \quad (5)$$

Переходы с нулевым временем срабатывания называются немедленными переходами и подходят, когда продолжительность события можно игнорировать. Запуск перехода представляет собой атомарную операцию: токены удаляются из его входных мест и помещаются в его выходные места с помощью одной неделимой операции, в отличие от синхронизированных по времени моделей сетей Петри, где переходы разбиваются на три фазы.

Задержка срабатывания связана с каждым переходом и определяет количество времени, которое должно пройти, прежде чем переход сможет сработать. Эта задержка срабатывания является случайной величиной с отрицательной экспоненциальной функцией плотности вероятности (PDF). Параметром PDF, связанным с переходом t_i , является скорость перехода $\lambda_i(M_j)$, которая может зависеть от маркировки. Средняя задержка перехода t_i при маркировке M_j равен:

$$[\lambda_i(M_j)]^{-1}. \quad (6)$$

Всякий раз, когда изменение маркировки включает переход, который ранее не был включён с момента его последнего срабатывания, этот переход выбирает экземпляр задержки срабатывания из связанного отрицательного экспоненциального распределения PDF и устанавливает таймер на значение экземпляра выборочной задержки. Пока переход включён, таймер уменьшается с постоянной скоростью. Если переход отключён из-за срабатывания конфликтующего перехода, таймер останавливается и декремент возобновляется, когда

переход снова включается. Когда таймер достигает нулевого значения, срабатывает переход.

1.3.5 Анализ стохастической сети Петри

Рассмотрим два основных метода анализа моделей sTPN:

- 1) прямой переходный анализ;
- 2) обратный переходный анализ.

Прямой переходный анализ[5] оценивает дерево, состоящее из мест, где дуги соединены с переходами и вероятностями их срабатывания. Места — это классы стохастических состояний[1], включающие маркировку, функцию плотности вероятности (PDF) таймеров и матрицу их границ различий (DBM). Для заданного времени T , перечисление продолжается до тех пор, пока дерево не покроет переходные срабатывания sTPN к моменту времени T с вероятностью большей, чем $1 - \epsilon$, где $\epsilon > 0$ это ошибка.

Обратный переходный анализ, в то время как стандартный анализ переходных процессов перечисляет одно очень большое дерево событий, регенеративный анализ избегает перечисления повторяющихся поддеревьев, основанных на одной и той же точке регенерации (где все общие таймеры сбрасываются или включаются на детерминированное время). Шаг по времени используется для выбора равноотстоящих точек времени, в которых оцениваются переходные вероятности (непосредственно или путём решения уравнений восстановления Маркова).

1.4 Квантовое моделирование

Квантовые явления остаются малоизученными и непонятными для большинства людей, в основном потому, что квантовая механика явно не проявляется на уровне обыденного опыта. Доступ к пониманию механизмов квантовых систем часто требует специализированных лабораторий и математических знаний, которые не являются частью общего образования. Однако для интуитивного понимания квантовой механики глубокие математические знания могут быть необязательны. С развитием квантовых компьютеров возникает вопрос о возможности их использования для квантовых игр[6], которые могут стать полезным инструментом для демонстрации квантовых явлений[7]. Такие игры могут быть адаптированы для демонстрации различных аспектов квантового поведения, что особенно важно при ограниченных ресурсах. Алгоритмы искусственного интеллекта, разработанные для этих игр, могут использовать

возможности квантовых вычислений для анализа ходов[8]. Развитие таких алгоритмов может привести к улучшению квантовых технологий[9].

1.4.1 Фазовое пространство

Фазовое пространство (ФП) в классической механике и статистической физике[10], многомерное пространство всех обобщённых координат q_1 и обобщённых импульсов $p_i, i = 1, 2, \dots, N$, механической системы с N степенями свободы. ФП имеет размерность $2N$ и может быть описано с помощью ортогональной системы координат с $2N$ осями соответственно числу обобщённых координат и импульсов. Состояние системы изображается точкой с координатами $q_i, p_i, \dots, q_N, p_N$, а изменение состояния системы во времени – движением точки вдоль линии, называемой фазовой траекторией. Для ФП можно ввести понятие фазового объёма и понятия геометрии для множественных измерений. Фазовое пространство – основное для классической статистической механики, изучающей функции распределения системы многих частиц[11]. Методы ФП успешно применяются также в теории нелинейных колебаний. Пример отображения двумерного фазового пространства представлен на рисунке 6.

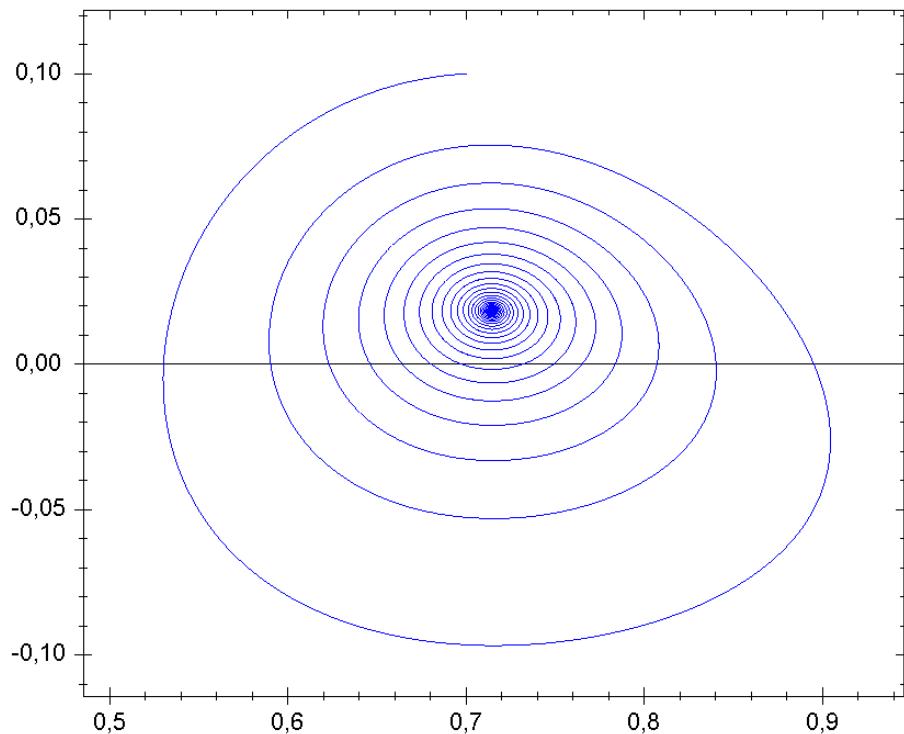


Рисунок 6 – Двумерное фазовое пространство

1.4.2 Квантовые шахматы

Квантовые шахматы расширяют классический вариант использованием фазового пространства и принципа суперпозиции. Пример реализации партии квантовых шахмат[12] представлен на рисунке 7.

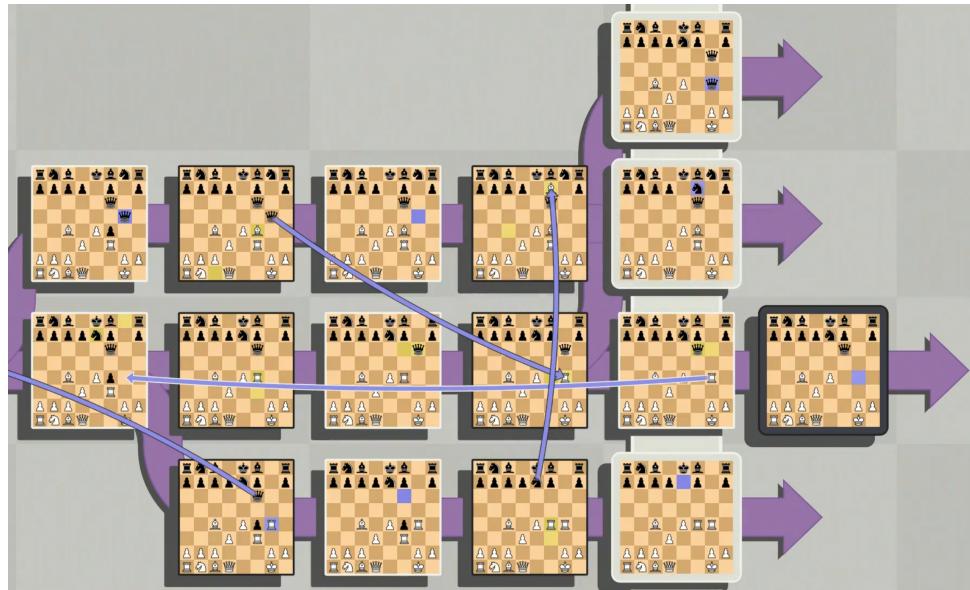


Рисунок 7 – Игровое поле

Классические шахматы имеют устоявшиеся алгебраические обозначения[13], для квантовых шахмат справедливо следующее:

- неквантовое перемещение: (*источник*)(*цель*);
- квантовое разделение: (*источник*) \wedge (*цель1*)(*цель2*);
- квантовое слияние: (*источник1*)(*источник2*) \wedge (*цель*);
- превращение пешки: (*источник*)(*цель*)(*фигура*)

Все неквантовые ходы включают в себя один источник и одну цель, рокировка тоже может полностью быть описана как источником король и цель короля. Добавление дополнительных символов в строку хода для получения более подробного описания, как это принято в стандартных шахматах, является сложной задачей, учитывая природу суперпозиции. Ещё одно соображение заключается в том, что в квантовых шахматах используются измерения. Расширенная нотация с учётом измерения выглядит как: (*ход*).(*измерение*). Тип измерения определяется по типу хода, который соответствует строке хода, действующей на текущее состояние игры.

Для описания состояния доски используется гибридное классическое квантовое представление состояний размером 64 кубита, где 0 это «свободное» со-

стояние, а 1 – «занятое»:

$$|\psi_\beta\rangle = \sum_i A_i |q_o^{(i)}, \dots q_{63}^{(i)}\rangle, q_j \in \{0, 1\}. \quad (7)$$

Это представление состояния похоже на представление «бит-доски», используемое в классических шахматах[13], где наше состояние аналогично суперпозиции «бит-доски» «все фигуры». Поверх этой суперпозиции занятости храниться классическая информация о типе для каждого квадрата. Эта информация состоит из одного вектора размерностью в 64 элемента, который описывает, какая фигура (если таковая имеется) занимает данную клетку.

$$\vec{v} = \{v_0, \dots v_{63}\}, v_i \in \{0, P, N, B, R, Q, K, p, n, b, r, q, k\}. \quad (8)$$

Значения v_i соответствуют стандартным значениям *FEN* для шахматных фигур[14]: строчные буквы обозначают чёрные фигуры, белые фигуры – заглавные, а 0 – пустой. Клетка занята, или частично занята фигурой, если существует ненулевая вероятность найти эту фигуру в этом квадрате.

Для каждого хода можно посчитать его вероятность:

$$P_m(\vec{v}, F) = \bigwedge_i C_i(m, \vec{v}, F). \quad (9)$$

где $C_{m,i}(\vec{v}, F)$ – множество ограничений описывающих текущее состояние для возможности совершения хода m .

1.4.3 Стандартный ход

Стандартный ход (англ. Standard Slide (SS)) – это эквивалент стандартного шахматного хода для слонов, ладей и ферзей в квантовых шахматах. Эти фигуры перемещаются по траектории, поэтому необходимо учитывать занятость клеток между источником и целью, для этого необходима вспомогательная переменная, отражающая «путь». Уравнение возможности стандартного перехода имеет следующий вид:

$$P_{SS} = (v_s \in \{B, R, Q\}) \wedge valid(t, s, v_s) \wedge ((v_t = 0) \vee (v_t = v_s)). \quad (10)$$

где *valid* – процедура проверки возможности хода, исходя из формулы вероятности хода.

1.4.4 Второй шаг при суперпозиции

В стандартных шахматах разрешён шаг второй пешки при первом ходе любой пешки. Если путь не заблокирован, пешка может продвинуться на две клетки вперёд. Уравнение возможности такого хода при наличии суперпозиции:

$$P_{PTS} = (v_s = P) \wedge (v_t \neq 0) \wedge (v_t \neq P) \wedge \text{two_step}(t, s). \quad (11)$$

Пример хода показан на рисунке 8.

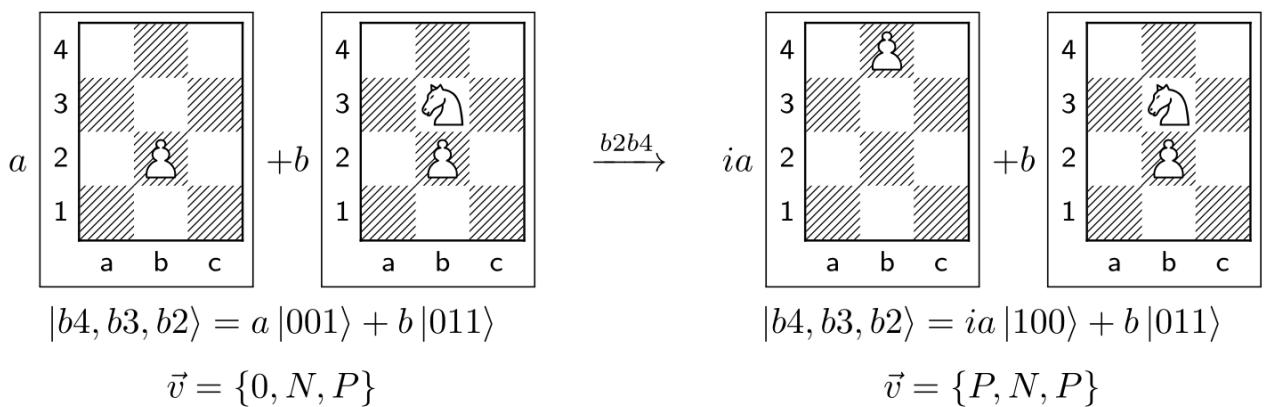


Рисунок 8 – Ход пешки с учётом суперпозиции

1.4.5 Вывод

Используя метод квантовых шахмат можно просчитать вероятность нахождения документа в каждом состоянии, по принципу суперпозиции документ одновременно находится во всех состояниях сразу и ни в одном. По аналогии с шахматами может быть получен путь документа.

2 Конструкторская часть

2.1 Проектирование системы

Верхнеуровневая диаграмма представлена на рисунке 9. На вход система получает данные о процессах документа и на выходе формирует результат моделирования исходя из статусной модели документа.

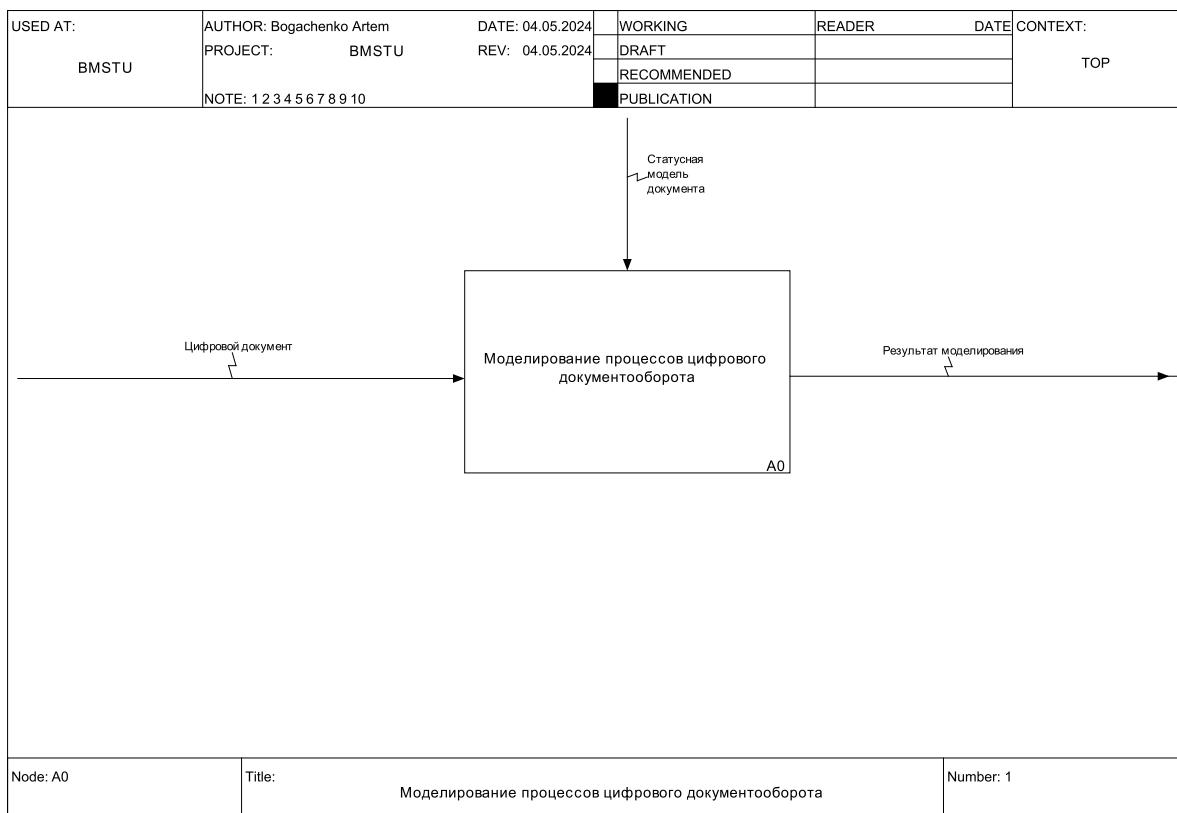


Рисунок 9 – Верхнеуровневая диаграмма приложения

Результатом моделирования в данном случае являются временные границы прохождения документа по статусной модели и вероятность данного события.

2.2 Проектирование алгоритма с использованием sTPN

Для моделирование с помощью стохастической сети Петри необходимо программное обеспечение, позволяющее построить сеть Петри в соответствие со спецификой процессов определённого документа. Данное программное обеспечение должно предоставлять возможность введения стохастического перехода между местами в сети, определяющими статусы документа в соответствии со статусной моделью. Для проведения моделирование необходима

возможно проведение прямого и обратного переходного анализа сети. Для интерпретации полученных результатов необходимо визуализация.

2.3 Проектирование алгоритма с использованием квантового моделирования

Аналогично концепции квантовых шахмат, размерность игрового поля при моделировании процессов цифрового документооборота будет зависеть от количества статусов, предусмотренных статусной моделью документа, от 1 до N . В качестве фигуры будет документ. Ход совершается перемещением документа из одного статуса в другой. Формула вероятности хода для шахмат 9 справедлива и для подсчёта вероятности перехода документа из одного статуса в другой. Для расчёта «пути» документа D , остаётся неизменной формула стандартного хода 10, с единственным убранным ограничением на фигуры $v_s \in \{D\}$, ввиду того, что единственno-возможной фигурой в данном случае является сам документ и только он:

$$P_D = (v_s \in \{D\}) \wedge valid(t, s, v_s) \wedge ((v_t = 0) \vee (v_t = v_s)). \quad (12)$$

Путь документа представлен на рисунке 10.

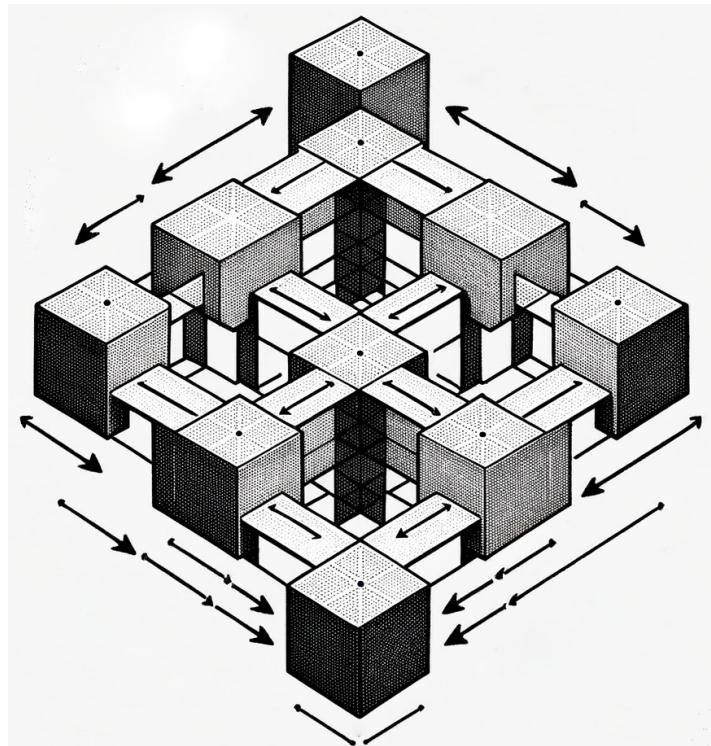


Рисунок 10 – Визуализация пути документа

2.4 Схема алгоритма расчёта вероятностей

Схема алгоритма подсчёта вероятностей представлена на рисунке 11.

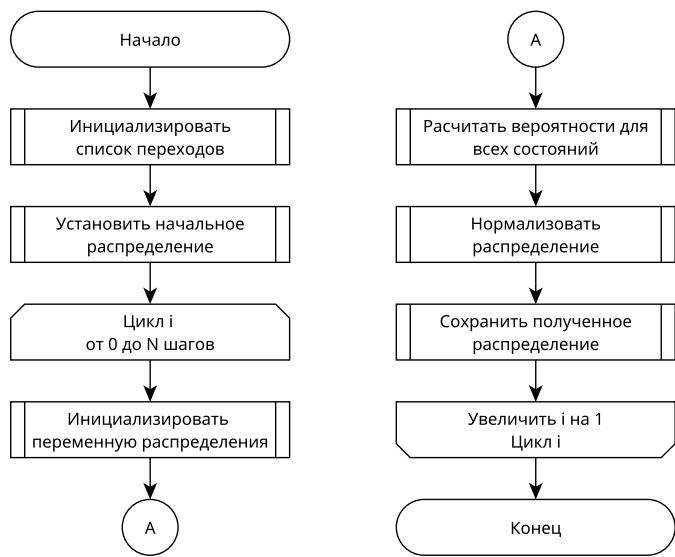


Рисунок 11 – Схема алгоритма подсчёта вероятностей

3 Технологическая часть

3.1 Реализация алгоритма моделирования с помощью sTPN

Для построения сети Петри и дальнейшей обработки результатов моделирования было выбрано программное обеспечение OrisTool[15]. Данное ПО обладает всеми необходимыми для решения поставленной задачи функциями:

- возможность построения стохастической сети Петри;
- возможность проведения моделирования сети Петри;
- наличие прямого и обратного переходного анализа;
- визуализация полученных данных.

3.2 Сопоставление BPMN 2.0 и сетей Петри

Для корректного построения модели необходимо установить связь между элементами BPMN 2.0 с сетями Петри. Сопоставление представлено на рисунке 12.

BPMN Object	Petri-net Module	BPMN Object	Petri-net Module
Start s	$P_s \xrightarrow{t_s} P(s, y)$	End e	$P(x, e) \xrightarrow{t_e} P_e$
Message E	$P(x, E1) \xrightarrow{E1} P(E1, y)$	Task T	$P(x, T1) \xrightarrow{T1} P(T1, y)$
Fork F1	$P(x, F1) \xrightarrow{t_{F1}} P(F1, y_1) \parallel P(F1, y_2)$	Join J1	$P(x_1, J1) \parallel P(x_2, J1) \xrightarrow{t_{J1}} P(J1, y)$
(Data-based) Decision D1	$P(x, D1) \xrightarrow{t(D1, y_1)} P(D1, y_1) \parallel P(D1, y_2)$	Merge M1	$P(x_1, M1) \parallel P(x_2, M1) \xrightarrow{t(M1, x_1)} P(M1, y) \parallel P(x_1, M1) \xrightarrow{t(M1, x_2)} P(M1, y)$

Рисунок 12 – Сопоставление BPMN 2.0 и сетей Петри

3.3 Реализация алгоритма квантового моделирования

3.3.1 Выбор и обоснование средств разработки

В качестве языка программирования был выбран логический язык SWI-Prolog[16] – инструмент для работы с логическим программированием, предназначенный для поиска заданного решения. Для решения поставленной задачи данный язык программирования обладает следующими достоинствами:

- декларативный подход: в отличие от императивных языков программирования, SWI-Prolog позволяет описывать задачи и решения в декларативном виде.

- тивной форме – что должно быть сделано, а не как именно это нужно выполнить;
- управление перебором: эффективное управление процессом перебора возможных решений (англ. *backtracking*) позволяет оптимально находить все возможные решения для заданной проблемы, автоматически пробуя различные варианты и отсекая неверные пути;
 - встроенные предикаты: SWI-Prolog предлагает обширную библиотеку встроенных предикатов для различных математических операций, что упрощает реализацию многих стандартных математических функций и операций;
 - работа с символическими выражениями: возможность обрабатывать и вычислять символические выражения, позволяет отказаться от полного числового ввода, что полезно для математических формул, требующих символического вывода, а не только численного решения;
 - интеграция с другими языками и системами: данный языка возможно интегрировать с другими языками программирования, такими как Python или C, что позволяет использовать SWI-Prolog для логических и символьических вычислений в рамках более крупных систем;
 - отладка и разработка: SWI-Prolog предлагает различные инструменты для отладки и трассировки программ, что помогает в разработке сложных алгоритмов и облегчает поиск и исправление ошибок;
 - библиотека для работы в числовых доменах[17].

3.3.2 Листинги кода

Реализация программы для подсчёта вероятности прохождения документа из начального статуса в финальный и поиска данного статусного пути, без учёта базы знаний, представлена в листинге 1.

Листинг 1 – Реализация поиска пути и вероятности

```
1 transition_probability(EFT, LFT, Probability) :-  
2     Probability is (LFT - EFT) / LFT.  
3  
4 find_path(Start, End, Path, Probability, Steps) :-  
5     travel(Start, End, [Start], 1, 0, RevPath, Probability, Steps),  
6     reverse(RevPath, Path).  
7  
8 travel(Start, End, Visited, CurrentProb, CurrentSteps, [End|Visited],  
9         FinalProb, FinalSteps) :-  
10    transition(Start, End, EFT, LFT),  
11    transition_probability(EFT, LFT, StepProb),  
12    FinalProb is CurrentProb * StepProb,  
13    FinalSteps is CurrentSteps + 1.  
14 travel(Start, End, Visited, CurrentProb, CurrentSteps, Path, FinalProb,  
15         FinalSteps) :-  
16    transition(Start, Mid, EFT, LFT),  
17    Mid \= End,  
18    \+member(Mid, Visited),  
19    transition_probability(EFT, LFT, StepProb),  
20    NewProb is CurrentProb * StepProb,  
21    NewSteps is CurrentSteps + 1,  
22    travel(Mid, End, [Mid|Visited], NewProb, NewSteps, Path, FinalProb,  
23         FinalSteps).
```

Реализация программы для нахождения вероятностного распределения представлена в листинге 2.

Листинг 2 – Реализация построения вероятностного распределения

```
1  initial_distribution(["status1"-p1, "status2"-p2]).  
2  
3  update_distribution(Dist, NewDist) :-  
4      findall(S-NewProb, (  
5          member(S-_, Dist),  
6          findall(P, (  
7              member(S1-P1, Dist),  
8              transition(S1, S, EFT, LFT),  
9              P is (EFT / LFT) * P1  
10             ), Probs),  
11             sum_list(Probs, TotalProb),  
12             NewProb is TotalProb  
13             ), TempDist),  
14             normalize_distribution(TempDist, NewDist).  
15  
16  normalize_distribution(Dist, Normalized) :-  
17      sum_list(Dist, Total, 0),  
18      maplist(normalize_prob(Total), Dist, Normalized).  
19  
20  normalize_prob(Total, S-P, S-NormP) :- NormP is P / Total.  
21  
22  sum_list([], Total, Total).  
23  sum_list([_-P|T], Total, Acc) :- NewAcc is Acc + P, sum_list(T, Total,  
24      NewAcc).  
25  
26  perform_steps(Dist, 0, [Dist]).  
27  perform_steps(Dist, N, [Dist|Rest]) :-  
28      N > 0,  
29      update_distribution(Dist, NewDist),  
30      M is N - 1,  
31      perform_steps(NewDist, M, Rest).  
32  
33  print_numeric_results([]).  
34  print_numeric_results([H|T]) :-  
35      findall(Prob, member(_-Prob, H), Probs),  
36      writeln(Probs),  
37      print_numeric_results(T).  
38  
39  run :-  
40      initial_distribution(Initial),  
41      perform_steps(Initial, 20, Results),  
42      print_numeric_results(Results).  
43  ?- run.
```

4 Исследовательская часть

В данном разделе будет исследован процесс утверждения темы. Модель бизнес-процесса представлена на рисунке 13.

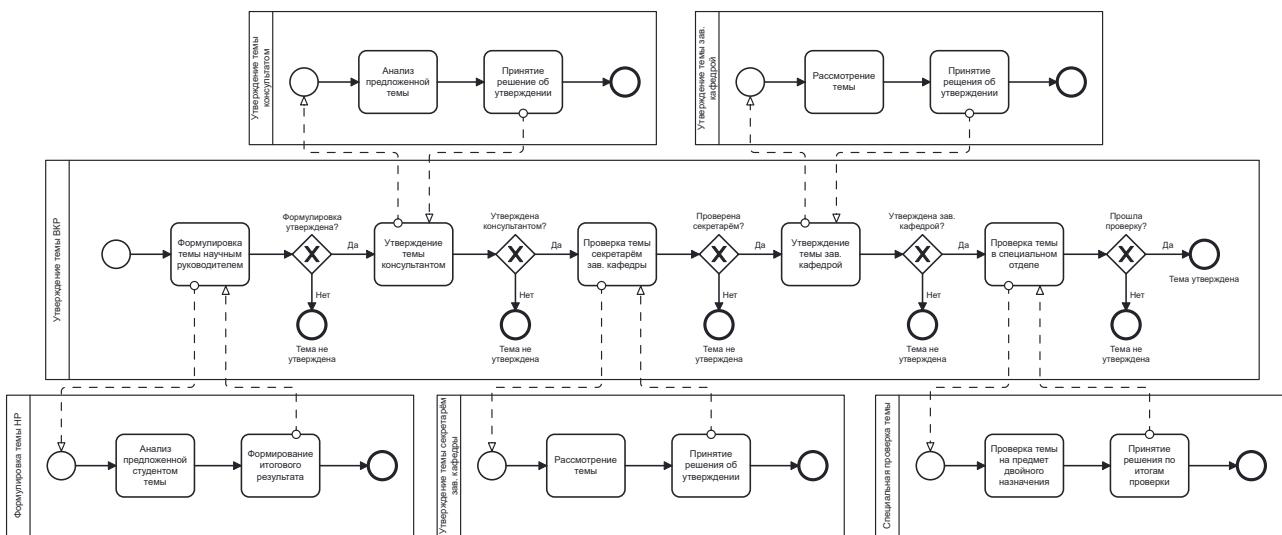


Рисунок 13 – Модель сети Петри

4.1 Моделирование с помощью sTPN

Построим соответствующую процессу стохастическую модель сети Петри для дальнейшего моделирования процесса. Итоговая модель сети Петри представлена на рисунке 14.

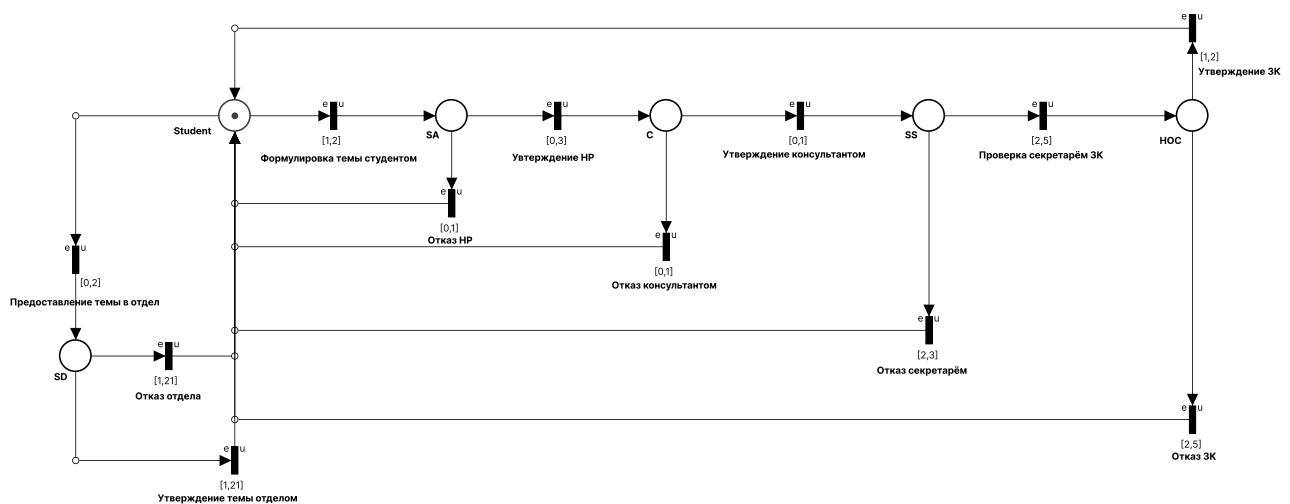
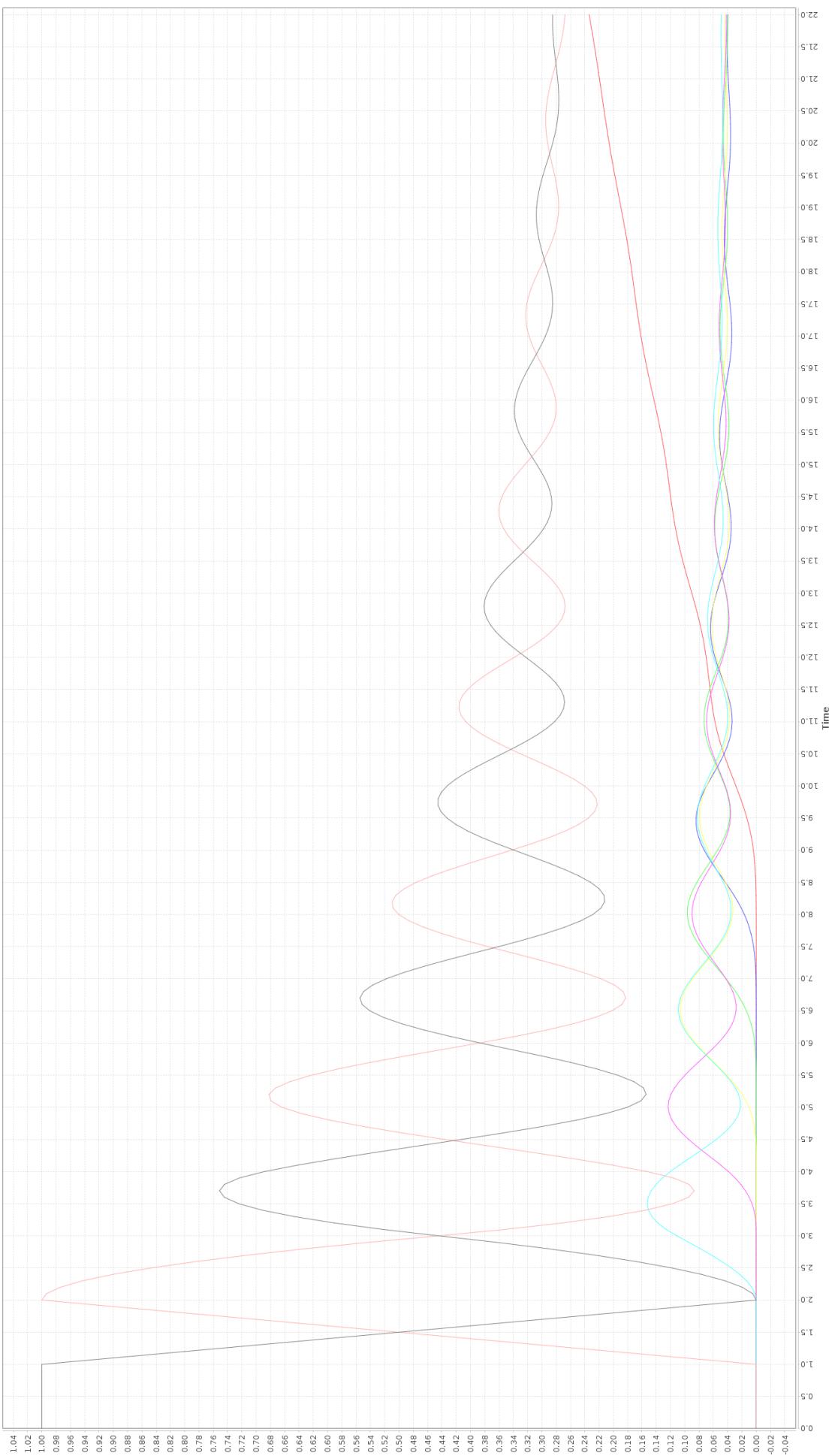


Рисунок 14 – Модель сети Петри

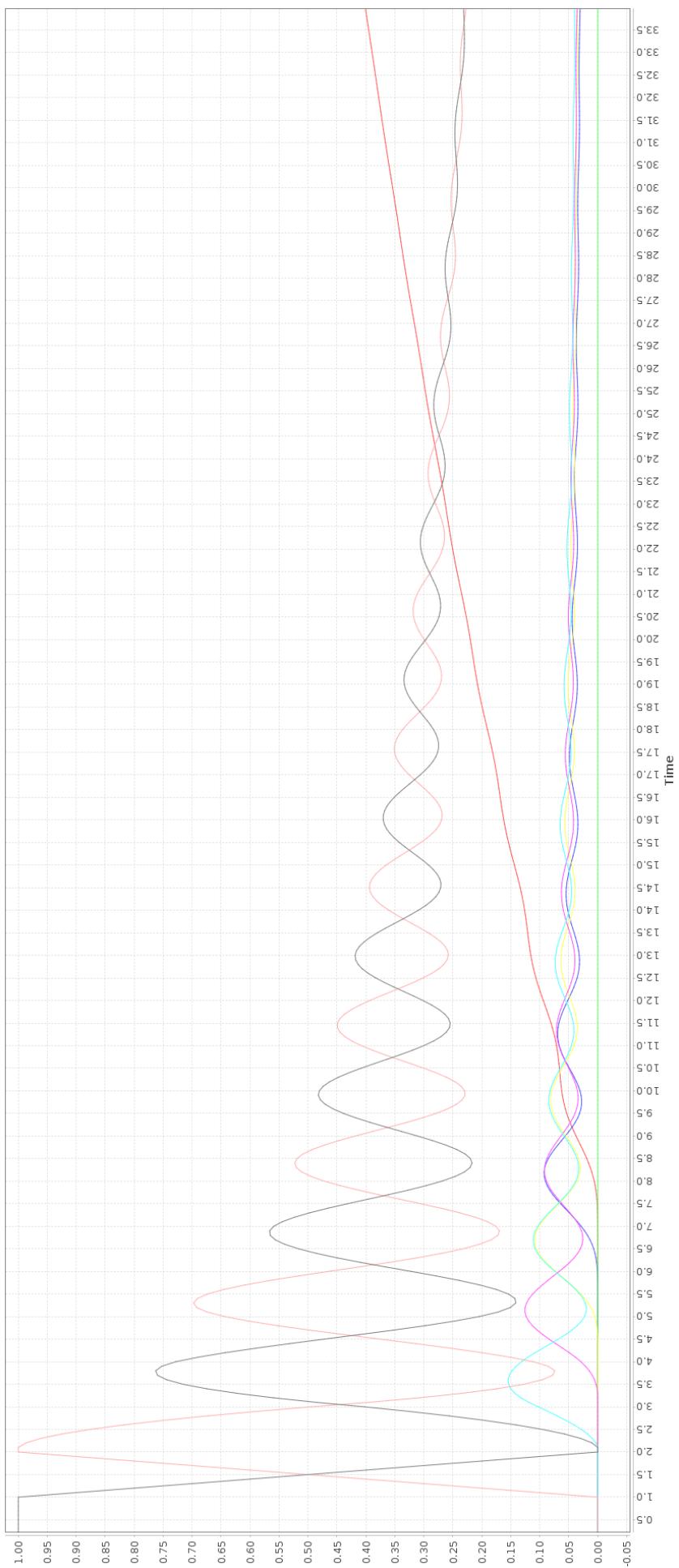
Прямой анализ переходных процессов[18] на временном отрезке в 22 временных единицы показал сходимость модели к 20 временным единицам. Результат прямого анализа переходных процессов представлен на рисунке 15.

Рисунок 15 – Результат прямого анализа переходных процессов сети Петри



Регенеративный анализ переходных процессов[19] на временном отрезке в 34 временные единицы с использованием переменной, отвечающей за полное выполнение цикла модели – Reward variable[20] подтвердил результаты прямого анализа. Результат регенеративного анализа переходных процессов представлен на рисунке 16.

Рисунок 16 – Результат регенеративного анализа переходных процессов



Результаты заключительного регенеративного анализа для установления отношения вероятности достижения успешного исхода событий от времени представлены на рисунке 17.

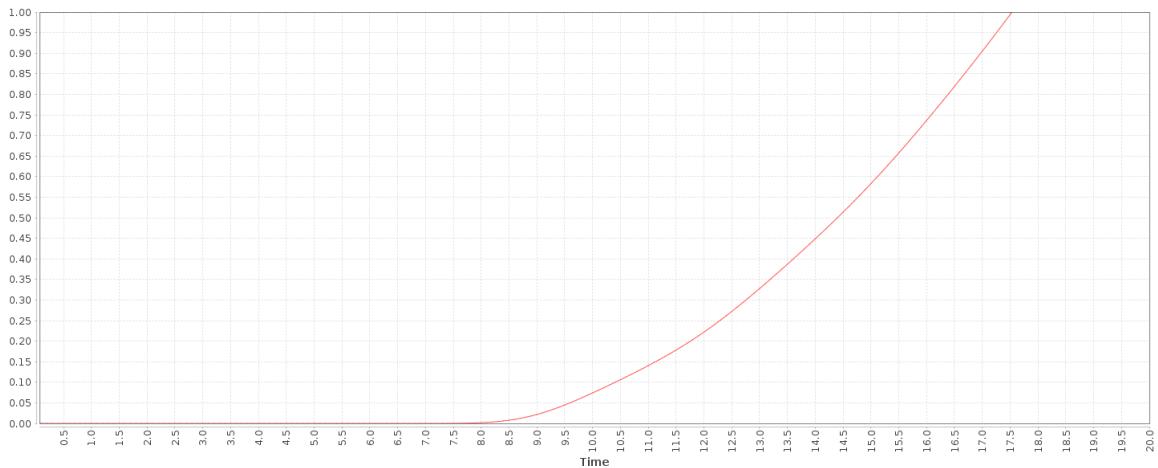


Рисунок 17 – Результат регенеративного анализа сети Петри с переменной

Исходя из полученных данных прямого и обратного анализов максимальное время на выполнение процесса может занять до 18 временных единиц. Совокупное время моделирования составило 72 часа.

4.2 Квантовое моделирование

Для расчёта необходимо заполнить базу знаний. В качестве стохастических переходов использовано аналогичное сетям Петри нормальное распределение с параметрами EFT (Expected Finish Time) и LFT (Latest Finish Time). База знаний представлена на рисунке 18.

```
transition("начало", "утверждение - НР", 1, 2).
transition("утверждение - НР", "начало", 1, 2).
transition("утверждение - НР", "утверждение - К", 1, 4).
transition("утверждение - К", "начало", 1, 2).
transition("утверждение - К", "проверка секретарём", 1, 2).
transition("проверка секретарём", "начало", 1, 5).
transition("проверка секретарём", "утверждение - ЗК", 1, 2).
transition("утверждение - ЗК", "начало", 1, 8).
transition("утверждение - ЗК", "рассмотрение в отделе", 1, 2).
transition("рассмотрение в отделе", "начало", 1, 2).
transition("рассмотрение в отделе", "утверждён", 1, 2).
```

Рисунок 18 – База знаний

Результат работы программы представлен на рисунке 19.

```
(root💀NebuchadnezzaR) - [~/bmstu/bmstu-diploma/diploma/src]
# swipl kvant4.pl
Welcome to SWI-Prolog (threaded, 64 bits, version 9.0.4)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- find_path("начало", "утверждён", Path, Prob, Steps).
Path = ["начало", "утверждение - НР", "утверждение - К", "проверка секретарём", "утверждение - ЗК", "рассмотрение в отделе", "утверждён"],
Prob = 0.0234375,
Steps = 6 .
```

Рисунок 19 – Результат работы программы

Вероятность успешно пройти всю цепочку составляет 2.3%. Распределение вероятностей представлено в таблице 1 в приложении А. Визуализация полученный данных представлена на рисунке 20. Совокупное время моделирование составило 2 миллисекунды.

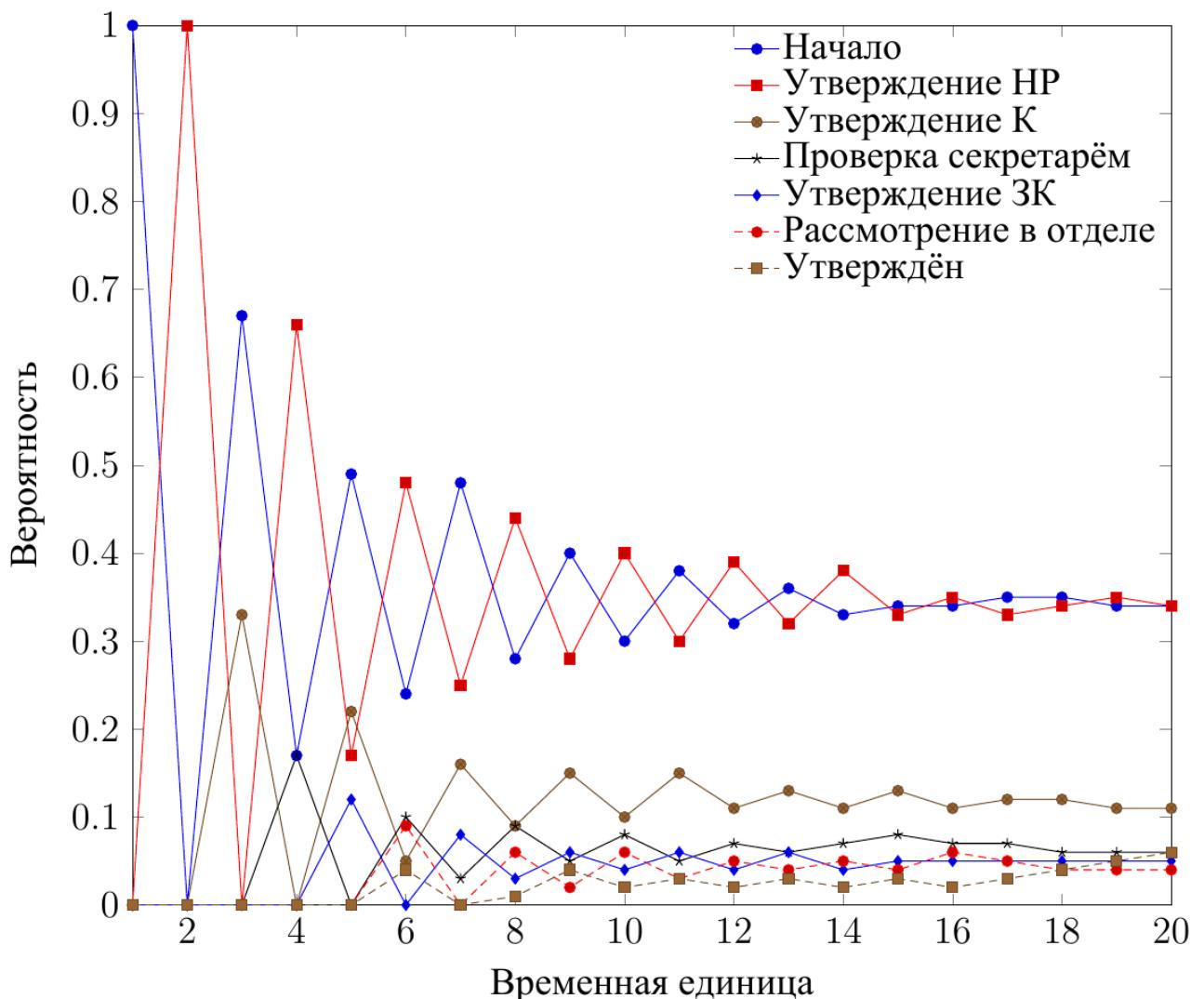


Рисунок 20 – График распределения вероятностей квантового моделирования

Возьмём другой произвольный пример. Соответствующая модель представлена на рисунке 21.

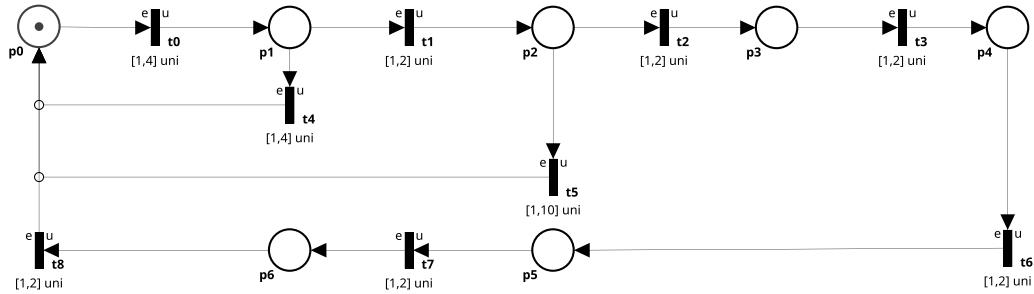


Рисунок 21 – Модель сети Петри

Соответствующая база знаний представлена на рисунке 22.

```
transition("p0", "p1", 1, 4).
transition("p1", "p0", 1, 4).
transition("p1", "p2", 1, 2).
transition("p2", "p0", 1, 10).
transition("p2", "p3", 1, 2).
transition("p4", "p5", 1, 2).
transition("p5", "p6", 1, 2).
transition("p6", "p0", 1, 2).
```

Рисунок 22 – База знаний

Регенеративный анализ сети Петри представлен на рисунке 23.

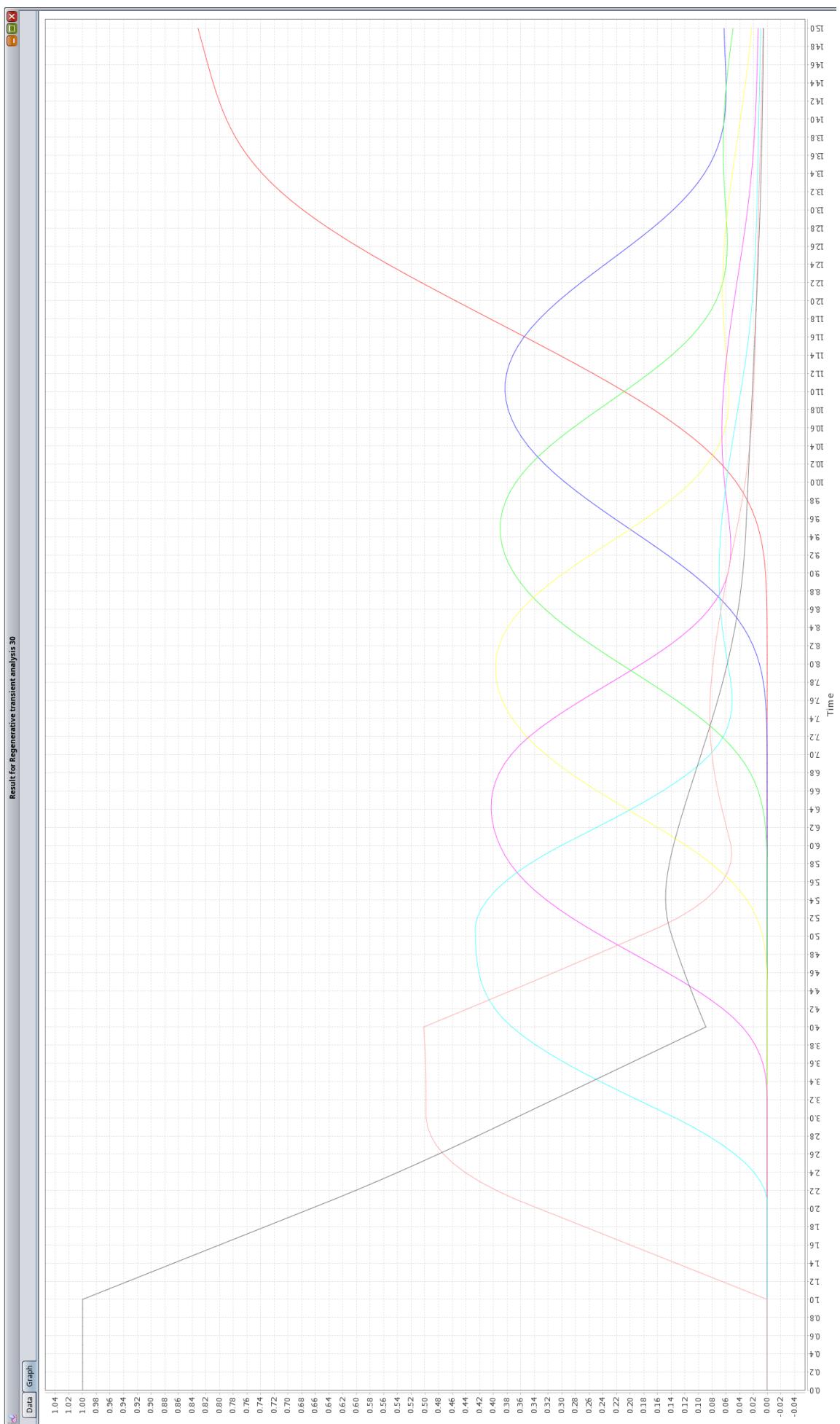


Рисунок 23 – Результат регенеративного анализа сети Петри

Результат моделирования методом квантовых шахмат представлен на рисунке 24.

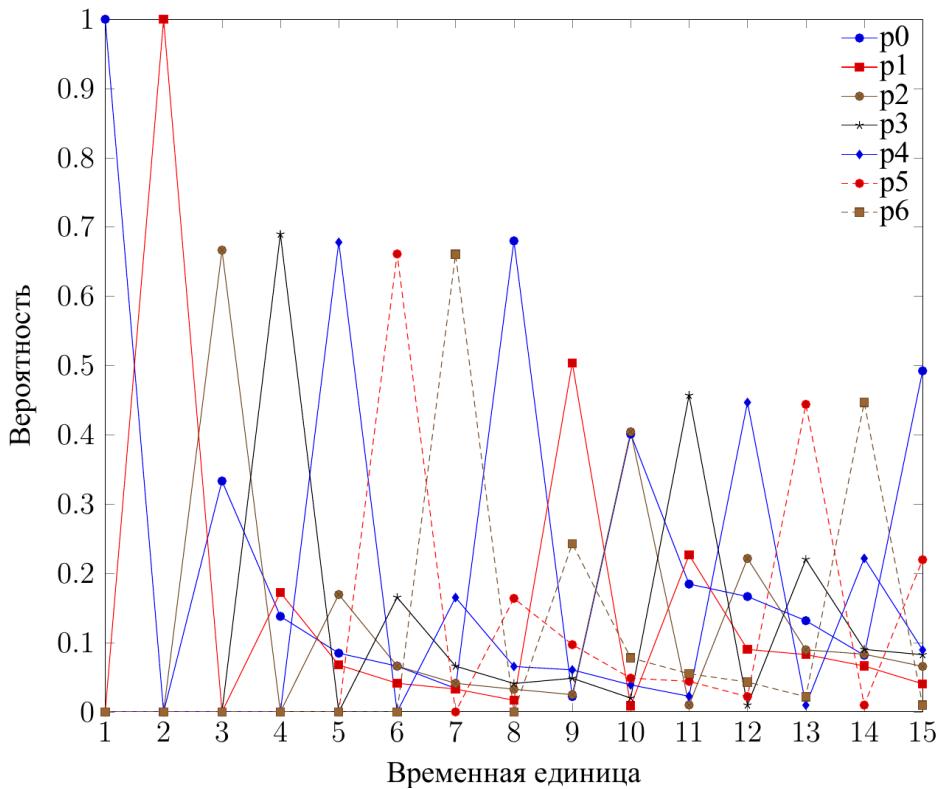


Рисунок 24 – Распределение вероятностей моделирования методом квантовых шахмат

Исходя из результатов, моделирование методом сетей Петри более точно отражает распределение вероятностей. Методу квантовых шахмат нужно на 20% больше шагов, чтобы сравняться с сетями Петри, что компенсируется более быстрым расчётом.

4.3 Вывод

Анализ данных показал, что максимальное время выполнения процесса может достигать 18 временных единиц, при этом вероятность успешного прохождения всей цепочки составляет 2.3%.

Квантовое моделирование продемонстрировало практически идентичные результаты распределения вероятностей по сравнению с моделированием с помощью сетей Петри. Однако, квантовое моделирование позволяет получить эти результаты в многократно быстрее, несмотря на невозможность отслеживания пути до конкретного статуса.

Результаты квантового моделирования также показали, что вероятность нахождения в финальном статусе возрастает с каждой итерацией, что подтвер-

ждается регенеративным анализом сетей Петри.

ЗАКЛЮЧЕНИЕ

Проведённое моделирование показало, что максимальное время выполнения процессов по утверждению темы может составить до 18 временных единиц. Полученные данные позволяют предложить меры по оптимизации текущих процессов, такие как объединение промежуточных состояний или разбиение длительных состояний на более короткие этапы. Системы ЭДО могут быть использованы для сокращения времени на переходы между состояниями процесса.

Квантовое моделирование позволило значительно быстрее подсчитать вероятность прохождения документа в финальное состояние и распределение вероятностей по времени, по сравнению с сетями Петри. Этот метод позволяет предварительно оценить сложность статусной модели документа. Моделирование с помощью сетей Петри дополнительно помогает установить временные границы для конкретных статусов, что является полезным для детальной оптимизации процессов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Stroganov Y. V., Bogachenko A. E. Modeling Digital Document Flow Processes with Stochastic Timed Petri Nets // 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE). 2024. C. 1–5.
2. Constantinescu C., Trivedi K. Dependability modeling of real-time systems using stochastic reward nets // Microelectronics Reliability. 1995. Т. 35, № 6. C. 903–914.
3. Bause F., Kritzinger P. Stochastic Petri Nets - An Introduction to the Theory. 2013. 11. с. 131.
4. Marsan M. A. Stochastic Petri nets: An elementary introduction // Advances in Petri Nets 1989 / под ред. Grzegorz Rozenberg. Springer Berlin Heidelberg, 1990. C. 1–29.
5. Transient analysis of non-Markovian models using stochastic state classes / András H., Marco P., Lorenzo R. [и др.] // Performance Evaluation. 2012. Т. 69, № 7. С. 315–335.
6. Piispanen L., Pfaffhauser M., Wootton J. [и др.]. Defining Quantum Games [Электронный ресурс]. Режим доступа: <https://doi.org/10.48550/arXiv.2206.00089> (дата обращения: 18 июня 2024 г.).
7. Discrete quantum theories / Andrew J. H., Gerardo O., Amr S. [и др.] // Journal of Physics A: Mathematical and Theoretical. 2014. Т. 47, № 11. с. 115–305.
8. Gatti L. N., Lacalle J. A model of discrete quantum computation // Quantum Information Processing. 2018. Т. 17, № 8. с. 192.
9. Altafini C., Ticozzi F.. Modeling and Control of Quantum Systems: An Introduction // IEEE Transactions on Automatic Control. 2012. Т. 57, № 8. С. 1898–1917.
10. Terence T. Phase Space [Электронный ресурс]. Режим доступа: https://www.math.ucla.edu/~tao/preprints/phase_space.pdf (дата обращения: 18 июня 2024 г.).

11. Konyak V. V. Classical and quantum discrete dynamical systems // Physics of Particles and Nuclei. 2013. Jan. T. 44, № 1. C. 47–91.
12. 5D Chess With Multiverse Time Travel [Электронный ресурс]. Режим доступа: https://store.steampowered.com/app/1349230/5D_Chess_With_Multiverse_Time_Travel/ (дата обращения: 18 июня 2024 г.).
13. FIDE laws of chess [Электронный ресурс]. Режим доступа: <https://www.fide.com/FIDE/handbook/LawsOfChess.pdf> (дата обращения: 18 июня 2024 г.).
14. Forsyth-Edwards Notation (FEN) [Электронный ресурс]. Режим доступа: <https://www.chess.com/terms/fen-chess> (дата обращения: 18 июня 2024 г.).
15. Oris Tool [Электронный ресурс]. Режим доступа: <https://www.oris-tool.org/> (дата обращения: 18 июня 2024 г.).
16. SWI-Prolog [Электронный ресурс]. Режим доступа: <https://www.swi-prolog.org/> (дата обращения: 18 июня 2024 г.).
17. SWI-Prolog manual [Электронный ресурс]. Режим доступа: <https://www.swi-prolog.org/man/clpf.html> (дата обращения: 18 июня 2024 г.).
18. Choi H. K. Vidyadhar G. Trivedi K. S. Transient analysis of deterministic and stochastic Petri nets // Application and Theory of Petri Nets 1993. Springer, 1993. C. 166–185.
19. German R., Logothetis D., Trivedi K. S. Transient analysis of Markov regenerative stochastic Petri nets: a comparison of approaches // Proceedings 6th International Workshop on Petri Nets and Performance Models. 1995. C. 103–112.
20. Miklos T., Andras P. Performance analysis of Markov regenerative reward models // Performance Evaluation. 1996. T. 27-28. C. 1–18.

ПРИЛОЖЕНИЕ А

Таблица 1 – Распределение вероятностей

Статус \ Шаг	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Начало	1.00	0.00	0.67	0.17	0.49	0.24	0.48	0.28	0.40	0.30	0.38	0.32	0.36	0.33	0.34	0.34	0.35	0.35	0.34	
Утверждение - НР	0.00	1.00	0.00	0.66	0.17	0.48	0.25	0.44	0.28	0.40	0.30	0.39	0.32	0.38	0.33	0.35	0.34	0.34	0.35	
Утверждение - К	0.00	0.00	0.33	0.00	0.22	0.05	0.16	0.09	0.15	0.10	0.15	0.11	0.13	0.11	0.13	0.11	0.12	0.12	0.11	
Проверка секретарём	0.00	0.00	0.17	0.00	0.10	0.03	0.09	0.05	0.08	0.05	0.07	0.06	0.07	0.08	0.07	0.07	0.07	0.06	0.06	
Утверждение - ЗК	0.00	0.00	0.00	0.00	0.12	0.00	0.08	0.03	0.06	0.04	0.06	0.04	0.06	0.04	0.05	0.05	0.05	0.05	0.05	
Рассмотрение в отделе	0.00	0.00	0.00	0.00	0.09	0.00	0.06	0.02	0.06	0.03	0.05	0.04	0.05	0.04	0.06	0.05	0.04	0.04	0.04	
Утверждён	0.00	0.00	0.00	0.00	0.04	0.00	0.01	0.04	0.02	0.03	0.02	0.03	0.02	0.03	0.02	0.03	0.04	0.05	0.06	

ПРИЛОЖЕНИЕ Б