

## Математическое и компьютерное моделирование

Научная статья

Статья в открытом доступе

УДК 519.876.5

doi: 10.30987/2658-6436-2024-1-21-32

### РАЗРАБОТКА МОДЕЛИ ВР-СХЕМЫ НА ОСНОВЕ ТЕОРИИ АВТОМАТОВ И С ПРИМЕНЕНИЕМ МЕТОДОВ КВАНТОВОГО ПРОГРАММИРОВАНИЯ

Игорь Сергеевич Беляев<sup>1</sup>, Алексей Михайлович Петров<sup>2</sup>, Антон Николаевич Попов<sup>3</sup>

<sup>1,2</sup> Заполярный государственный университет им. Н.М. Федоровского, г. Норильск, Россия

<sup>3</sup> Государственный аграрный университет Северного Зауралья; Тюменский государственный университет, г. Тюмень, Россия

<sup>1</sup> belyaervis@norvuz.ru, <http://orcid.org/0000-0001-9149-4357>

<sup>2</sup> darker2012@yandex.ru, <http://orcid.org/0000-0002-3265-520X>

<sup>3</sup> 264241@mail.ru, <http://orcid.org/0000-0003-0774-411X>

**Аннотация.** *Описывается концепция нового вида абстрактных автоматов «ВР-схема» (Business Process), эффективно распределяющих управленческие команды между сущностями при их перманентном росте в парадигме сложно моделируемых систем. Исследована предлагаемая модель реализации ВР-схемы, представляющая собой искусственную нейронную сеть, функционирование которой основано на правилах нечеткой логики с применением алгоритма Такаги-Сугено, для вывода правил в форме функциональных зависимостей. Модель использует QPU при вычислениях поведения системы, что предполагает быстрый отклик на изменения сущностей в системе. Использованный с учетом выработанных правил нечеткой нейронной сети позволяет совершенствовать сценарии и прогностические модели поведения сложной системы.*

**Ключевые слова:** моделирование систем, система управления, искусственная нейронная сеть, квантовые технологии, абстрактные автоматы, ВР-схема

**Для цитирования:** Беляев И.С., Петров А.М., Попов А.Н. Разработка модели ВР-схемы на основе теории автоматов и с применением методов квантового программирования // Автоматизация и моделирование в проектировании и управлении. 2024. №1 (23). С.21-32. doi: 10.30987/2658-6436-2024-1-21-32.

Original article

Open Access Article

### DEVELOPMENT OF A BP-SCHEME MODEL BASED ON THE AUTOMATA THEORY AND USING QUANTUM PROGRAMMING METHODS

Igor S. Belyaev<sup>1</sup>, Aleksey M. Petrov<sup>2</sup>, Anton N. Popov<sup>3</sup>

<sup>1,2</sup> Fedorovsky Polar State University, Norilsk, Russia

<sup>3</sup> Northern Trans-Ural State Agricultural University; Tyumen State University, Tyumen, Russia

<sup>1</sup> belyaervis@norvuz.ru, <http://orcid.org/0000-0001-9149-4357>

<sup>2</sup> darker2012@yandex.ru, <http://orcid.org/0000-0002-3265-520X>

<sup>3</sup> 264241@mail.ru, <http://orcid.org/0000-0003-0774-411X>

**Abstract.** *The paper describes the concept of a new type of abstract automata “BP-scheme” (Business Process) that effectively distributes management commands between entities with their permanent growth in the paradigm of complexly modelled systems. The paper investigates the proposed model of the BP-scheme implementation, which is an artificial neural network, functioning based on fuzzy logic rules using the Takagi-Sugeno algorithm, for deriving rules in the form of functional dependencies. The model uses QPU (Quantum Processing Unit) in computing system behaviour, which implies a quick response to changes in the system entities. QPU used taking into account the developed rules of*

fuzzy neural network allows improving scenarios and predictive models of a complex system behaviour.

**Keywords:** system modelling, management system, artificial neural network, quantum technology, abstract automata, BP-scheme

**For citation:** Belyaev I.S., Petrov A.M., Popov A.N. Development of a BP-Scheme Model Based on the Automata Theory and Using Quantum Programming Methods. Automation and modeling in design and management, 2024, no. 1 (23). pp. 21-32. doi: 10.30987/2658-6436-2024-1-21-32.

## Введение

Технические системы идут по пути усложнения, всего за несколько десятилетий был осуществлен переход от моно- к мульти (много) агентности. В виртуальной среде в эффективном тандеме друг с другом «работают» искусственные нейронные сети, в физическом мире разрабатываются и реализовываются концепции взаимодействия между собою таких сложных систем, как технологические линии, цеха и заводы. Причем, уже на этапе разработки закладываются возможности быстрой «перенастройки» подобных сложных систем с одного вида продукции на другой. Подобные вещи невозможно было сделать без возможности предварительного моделирования сложных систем [1, 2].

## Методики моделирования систем

Моделирование систем, как научная область, в настоящее время обладает разнообразным методическим инструментарием, на рис. 1 показана его базовая классификация.



**Рис. 1. Классификация методик моделирования систем**  
**Fig. 1. Classification of system modeling techniques**

Фундаментальный математический аппарат данной научной области, на прикладном уровне представлен следующими абстрактными схемами:

1. *D-схемы*, которые представляют собой непрерывно-детерминированные модели, отражающие моделируемую систему в динамике (ее изменение, в зависимости от не статичного параметра, которым в большинстве случаев является время).

2. *F-схемы*, которые реализовываются в виде математической абстракции конечного автомата со множеством регулируемых или учитываемых внутренних состояний, а также входных и выходных сигналов, которые в большинстве случаев представляются в виде конечного множества.

3. *P-схемы*, которые реализовываются в виде вероятностных автоматов, через дискретно-стохастический подход, что позволяет симитировать условный преобразователь информации с возможностью запоминания и, как следствие, корректировки последующей своей работы на основе предыдущего функционирования.

4. *Q-схемы*, являющиеся имитацией очерёдности элементов в целом и систем массового обслуживания в частности. При этом схема учитывает вероятность появления случайного

«возмущения» в случайный момент времени, т.е. реализуется возможность учета стохастического характера процесса.

5. *N-схемы*, имитирующие трудноформализуемые системы, т.к. учитывают описания структуры и взаимодействия параллельных систем и процессов, с возможностью запоминания и обучения причинно-следственных связей в них.

6. *A-схемы*, которые являются всевозможными комбинациями вышеперечисленных схем.

Прикладная реализация данных схем приводит к усовершенствованию текущих технологических процессов и, как следствие, приводит к снижению логистических издержек, затрат ресурсов, повышению точности управления и иных сопутствующих явлений. Однако даже подобные подходы со временем станут неэффективными, поскольку представляемые производством задачи опережают потенциальные возможности абстрактной схемы.

Чаще всего подобные схемы оцениваются/сравниваются по критерию оптимальности затрат ресурсов на создание управленческих команд. Логично, что чем больше связей между сущностями, тем больше ресурсов затрачивается на создание подобных сигналов. Это в свою очередь означает необходимость создания технологии, которая способна осуществлять минимальное создание управленческих команд при увеличении сущностей.

В связи с этим предлагается концепция нового вида абстрактных автоматов «BP-схема» (Business Process), эффективно распределяющих управленческие команды между сущностями при их перманентном росте в парадигме сложно моделируемых систем.

### Концептуальные особенности нового вида абстрактных автоматов

*Постановка задачи.* Описание и принцип действия BP-схемы рассматриваются применительно к сложной системе, которая состоит из  $n$ -сущностей, функционал которых заключается в реагировании на поступающие триггеры в диапазоне  $[0 \dots N]$  и изменения состояния (режима работы) в зависимости от значения триггера.

Входные данные:

– Сложная система в двумерном пространстве, состоящая из связанных друг с другом  $n$ -сущностей.

–  $n$ -сущности, каждая из которых может пребывать в состояниях  $[0 \dots N]$ .

– Триггеры  $n$ -сущностей в диапазоне  $[0 \dots N]$ .

Требуется: разработать концептуальную модель BP-схемы; описать принцип действия BP-схемы в виде конечного автомата; сформулировать критерий оптимальности; провести модельные эксперименты и оценить эффективность предложенных решений, по сравнению с другими видами конечных автоматов.

*Описание моделей и алгоритмов.* В соответствии с постановкой задачи система рассматривается как сложная, а значит она состоит из множества взаимодействующих друг с другом сущностей. Взаимодействие сущностей приводит к появлению и приобретению системой новых свойств, которые отсутствуют на локальном инфраструктурном уровне. Это означает, что динамику развития подобной системы сложно моделировать ввиду перманентного изменения зависимостей/отношений, между сущностями и, как следствие, появления таких свойств, как нелинейность, эмерджентность, спонтанный порядок, адаптация и петли обратной связи. При визуализации подобных систем их изображают в виде сети, где узлы представляют компоненты и связи между ними. Также необходимо учитывать, что ряд сущностей может получать триггеры параллельно или одна сущность может получать одновременно несколько управленческих команд.

Для описания информационного потока триггеров и управленческих команд определим интервал времени, между их взаимодействиями с сущностями системы. Интервал времени определяется, как:

$$T_S = t_s - t_{s-1}, \quad (1)$$

где  $T_S$  – интервал времени бездействия системы, между взаимодействиями с сущностями;  $t_s$  – интервал времени, затраченный на взаимодействие с текущей сущностью в системе;

$t_{s-1}$  – интервал времени, затраченный на взаимодействие с предыдущей сущностью в системе.

Логично, что данная формула расчетов будет адекватна при рассмотрении сущностей, которые находятся на одном маршруте информационного потока системы, иными словами, когда сущности системы линейно взаимосвязаны друг с другом. Это, в свою очередь, приводит к суждению о том, что сложную систему, состоящую из различных сущностей, взаимосвязь между которыми может вероятностно измениться в каждый момент времени, можно подвергнуть процедуре аппроксимации и представить ее в следующем виде:

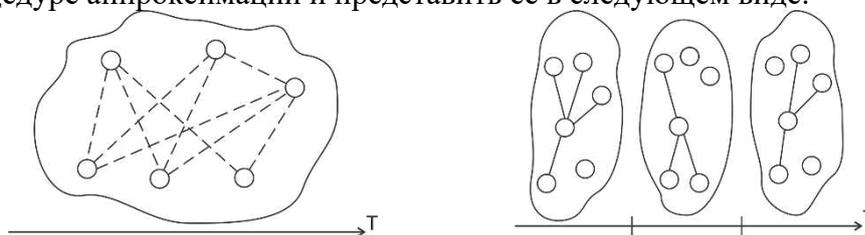


Рис. 2. Сложная система до аппроксимации (слева) и после (справа)  
 Fig. 2. Complex system before approximation (left) and after (right)

Подобное упрощение системы, а именно понимание какие сущности взаимодействуют друг с другом в каждый момент времени дает возможность расчета  $T_S$  для каждого графа, а следовательно, и возможность оценки интенсивности  $L_S$  по каждому его отрезку. Ввиду того, что информационные потоки нестационарные, их появления случайны, а рассматриваемые интервалы времени не равномерны, то процессы в системе могут быть охарактеризованы, как групповые потоки без последствия.

Учет вышеперечисленных моментов приводит к необходимости описать плотность распределения как:

$$\int_{-\infty}^{+\infty} f(T_S) dt = B(t). \quad (2)$$

И через нее выводим интенсивность как:

$$L_S = \frac{1}{B(t)}. \quad (3)$$

Временной интервал взаимодействия триггеров и управленческих команд с сущностью в системе определяется интервалом времени от момента начала взаимодействия до его завершения. При моделировании подобной системы необходимо учитывать приоритетность подобных информационных потоков для сущности. Количество подобных взаимодействий может быть неограниченно. Сущности не могут взаимодействовать с одним информационным потоком группой, но каждая сущность может создать на основе триггера или управленческой команды ряд других триггеров или управленческих команд на другие сущности, в этом случае сущность переходит в иную позицию. Всего у сущностей может быть три позиции:

- 1) исполнитель – когда сущность принимает триггеры или управленческие команды;
- 2) заказчик – когда сущность направляет триггеры или управленческие команды;
- 3) исполнитель/заказчик – комбинация из двух вышеперечисленных.

В итоге получается следующая схема:

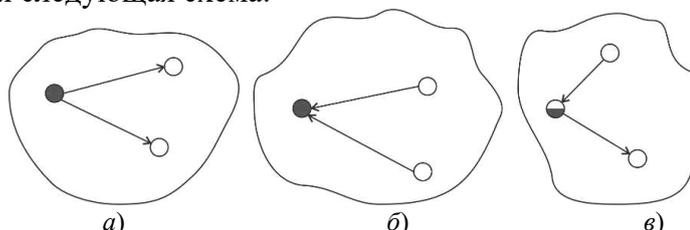


Рис. 3. Виды позиций сущностей в моделируемой системе:

*a* – сущность в позиции заказчик; *б* – сущность в позиции исполнитель; *в* – сущность в позиции исполнитель/заказчик

Fig. 3. Types of entity positions in the simulated system:

*a* – entity in the position customer; *b* – entity in the position of performer; *c* – entity in the position performer/customer

В данной системе информационный поток будет рассматриваться как однородный в связи с тем, что разный объем триггеров или управленческих команд, может быть обслужен

любой сущностью.

В связи со всем вышеизложенным, предложена следующая схема и временная диаграмма моделируемой системы (рис. 4).

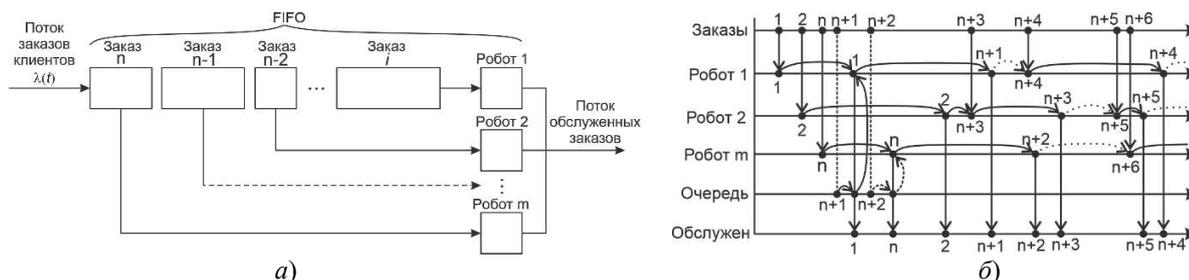


Рис. 4. Схема моделируемой системы (а) и временная диаграмма (б)  
 Fig. 4. Diagram of the simulated system (a) and the timing diagram (b)

В соответствии с временной диаграммой, представленной на рис. 4, приходящие на сущности триггеры или управленческие команды могут выполняться параллельно. В силу того, что триггеры или управленческие команды различаются по степени важности (ранжируются), то длительность взаимодействия с ними и их место в очереди для сущности может изменяться.

Таким образом, требуют решения следующие задачи:

- 1) разработать методику корректного ранжирования триггеров/управленческих команд;
- 2) определить алгоритм создания параллельных очередей триггеров/управленческих команд на одну сущность;
- 3) минимизировать время нахождения триггеров/управленческих команд в параллельных очередях на одну сущность;
- 4) создать процедуру определения позиции одной сущности относительно других.

Для оценки эффективности решения поставленных задач сформированы следующие критерии:

- 1) количество обрабатываемых одной сущностью триггеров/управленческих команд за определенный временной интервал;
- 2) максимальная длина очереди на одну сущность;
- 3) максимальное количество параллельных очередей на одну сущность;
- 4) равномерность распределения рангов триггеров/управленческих команд в очередях на одну сущность;
- 5) соотношение распределения состояний сущностей (исполнитель, заказчик, исполнитель/заказчик) внутри системы.

О пятом критерии поговорим подробнее, поскольку представляемая концепция нового вида автоматов «BP-схема», на прикладном уровне представляется авторами, как модель экономической системы взаимодействия друг с другом организаций-игроков, то необходимо обозначит следующее:

– При перенасыщении системы сущностями, прибывающими в состоянии «исполнитель» и «исполнитель/заказчик» система становится негативно нестабильной.

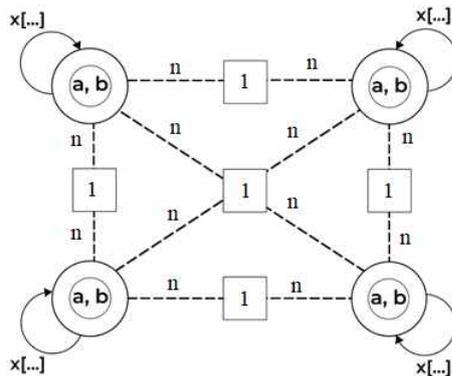
– При перенасыщении системы сущностями, прибывающими в состоянии «заказчик» система становится стабильно развивающейся.

В соответствии с вышеизложенными теоретическими положениями предлагается следующая формальная модель:

$$BP = \langle L_S, X, Y, Z, D, R, T \rangle, \quad (4)$$

где  $X = \{s_{1...3}, \tau\}$ ;  $s_{1...3}$  – одно из трех состояний, в котором пребывает сущность;  $\tau$  – приблизительная оценка времени пребывания сущности в данном состоянии;  $Y$  – объем триггеров/управленческих команд на сущности в дискретный момент времени  $t$ ;  $Z$  – длина очереди на одну сущность;  $D$  – количество параллельных очередей на одну сущность;  $R$  – правила назначения триггеров/управленческих команд на сущность;  $T$  – время работы сущности с триггерами/управленческими командами.

Опираясь на разработанную модель системы (1) мы можем представить ВР-схему в виде графа (рис. 5).



**Рис. 5. Граф автомата ВР-схемы:**

$a, b$  – внутренние состояния;  $x$  – условия перехода;  $n$  – триггеров/управленческих команд;  $1$  – оракул

**Fig. 5. The graph of the BP scheme automaton:**

$a, b$  – internal states;  $x$  – transition conditions;  $n$  – triggers/control commands;  $1$  – oracle

Если указанные выше элементы легенды графа классически, либо интуитивно понятны, «оракул» является уникальным для описания элементом. Естественно, необходимо объяснить данный инструмент, который является ключевым, для функционирования ВР-схемы.

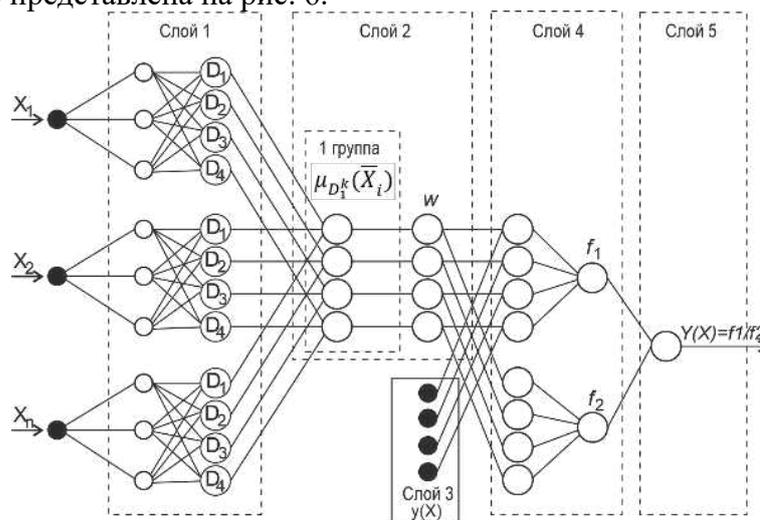
«Оракул» ВР-схемы представляет собой искусственную нейронную сеть, функционирование которой основано на правилах нечеткой логики. Приходящие на «оракул» информационные потоки в виде триггеров/управленческих команд могут быть выражены в виде цифрового сигнала и формировать динамическую базу данных для обучения нейронной сети и выработки уникальных правил для каждой ситуации.

Нечёткая нейронная сеть (ННС) «оракула» для аппроксимации входящих данных по информационному потоку использует алгоритм Такаги-Сугено для вывода правил в форме функциональных зависимостей:

$$R^{(k)}: \text{если } (x_1 \text{ есть } D_1^k \text{ И } \dots \text{ И } x_N \text{ есть } D_N^k), \text{ то } y = f^{(k)}(x_1 \dots x_n), \quad (5)$$

где  $R^{(k)}$  – условное нечеткое правило во множестве нечетких правил по признакам  $(k_1 \dots k_n)$ ;  $D_1^k$  – нечёткие множества во множестве нечетких правил;  $x_1 \dots x_n$  – входные переменные.

Структура ННС представлена на рис. 6.



**Рис. 6. Структура ННС «оракула»:**

$\mu_{D_1^k}$  – обобщенное линейризованное представление проявления признака нечёткого множества во множестве нечетких правил;  $y$  – выходное значение;  $f_{1..n}$  – аппроксимирующий элемент;  $w$  – весовой коэффициент

**Fig. 6. The structure of a neural network «oracle»:**

$\mu_{D_1^k}$  – generalized linearized representation of the manifestation of a fuzzy set feature in a set of fuzzy rules;  $y$  – output value;  $f_{1..n}$  – approximating element;  $w$  – weight coefficient

Функциональные особенности каждого слоя ННС следующие:

1. Распределение входных данных по потокам триггеров/управленческих команд, их преобразование в нечёткие переменные и параллельное сравнение с имеющимися данными.
2. Определение метрик истинности для правил, уже имеющихся в базе данных.
3. Обучение ННС, результатом которого является выражение входных данных в виде одной объединенной функции принадлежности.
4. Составление списка выработанных правил.
5. Переход от функции принадлежности к ее измеряемому значению.

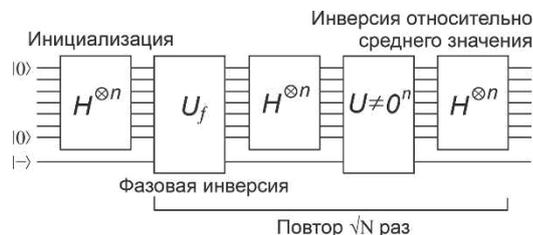
Итогом работы ННС является сценарий распределения потоков триггеров/управленческих команд на сущности, входящие в систему для поддержки корректного соотношения состояний сущности с целью сохранения системы в режиме стабильно развивающейся.

Как можно заключить, «оракул» является ключевым элементом ВР-схемы и описание алгоритма его работы, а также последующее моделирование является превалирующим для логики данной работы.

Поведение сложной системы, которую ВР-схема должна поддерживать в стабильно развивающемся состоянии функционирует в логике, что плотность потока триггеров/управленческих команд зависит от степени свободы сущности. Таким образом  $L_S$  является приоритетным элементом формальной модели и, как следствие, именно на него ориентируется «оракул» при составлении сценария, чтобы уменьшить плотность потока триггеров/управленческих команд на одну сущность и сократить время их выполнения.

Ввиду того, что триггеры/управленческие команды между сущностями должны распределяться в порядке параллельных очередей и не выстраиваться в одну последовательную очередь, то классические алгоритмы теории массового обслуживания в данном случае будут не эффективны. Поэтому для сохранения высоких метрик  $L_S$  логично будет использовать квантовый алгоритм Гровера, чаще всего использующийся для решения задачи перебора с целью поиска данных [3 – 8].

Обобщённая схема данного алгоритма представлена на рис. 7.



**Рис. 7. Обобщенная форма алгоритма Гровера для произвольного числа кубитов**  
**Fig. 7. The generalized form of Grover's algorithm for an arbitrary number of qubits**

При поступлении потока триггеров/«управленческих импульсов» в систему, «оракулу» необходимо осуществить расчет  $L_S$  и направить поток на ту сущность, где значение данного параметра будет минимальным. Предположим, что подобный поток будет состоять из  $k$  триггеров и  $n$  «управленческих импульсов».

В подобной постановке задачи оценка «оракулом» критерия  $L_S$  можно представить как:

$$L_S = \frac{1}{B(t)} \sum_{|0 \dots k\rangle_n} (t_{L_S} + t_p + t_s), \quad (6)$$

где  $t_{L_S}$  – время направления потока на сущность;  $t_p$  – длительность выполнения сущностью операции по триггеру;  $t_s$  – длительность выполнения сущностью операции по формированию управленческой команды.

Аппроксимируем сложное пространство сущностей до уровня отрезка-взаимодействия между двумя сущностями. Подробно рассмотрим одно взаимодействие между двумя сущностями с потоком триггеров/управленческих команд. Как можно понять существуют следующие сценарии:

- 1) одна сущность в позиции «заказчик», другая в позиции «исполнитель»;

- 2) одна сущность в позиции «заказчик/исполнитель», другая в позиции «исполнитель»;
- 3) обе сущности в позиции «заказчик/исполнитель».

Поэтому на уровне полноценной системы, которая состоит из подобных отрезков-взаимодействий, возникает задача нахождения оптимального пути для информационного потока триггеров/управленческих команд от одной сущности к другой через ряд произвольных сущностей, пока информационный поток не будет исполнен.

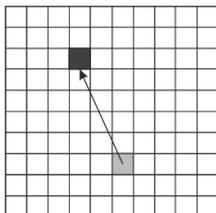
Логично, что подобную задачу можно рассмотреть, как поисковую задачу на графе, а именно:

$$G = (V, E), \quad (7)$$

где  $E$  – множество ребер;  $V$  – множество вершин.

Сложную систему взаимодействия сущностей, можно выразить в виде сетки и, как следствие, в виде графа, где его ребра – это сущности, а маршруты информационных потоков между ними – это вершины. Отсюда следует, что для рассмотрения каждого маршрута информационного потока необходимо найти некую последовательность ребер графа. Очевидно, что существует  $m$  маршрутов от стартовой до финальной сущности и чем дальше они будут находиться друг от друга, тем больше данных маршрутов может быть.

Рассмотрим процедуру поиска оптимального маршрута из  $m$  маршрутов с использованием инструментов и терминов комбинаторики. Систему и пролагаемый по ней маршрут информационного потока от стартовой сущности до финальной представлена на рис. 8.



**Рис. 8. Сложная система в виде пространства графа**  
*Fig. 8. Complex system represented as a graph space*

В этом случае стартовая позиция информационного потока (стартовая сущность, которая создаёт поток) и его конечная позиция (финальная сущность, которая поток выполняет) обязательны, а значит могут быть выражены, как  $m_s \in M$  и  $m_f \in M$  соответственно, где  $M$  – комбинация произвольно взятых элементов. Рассмотрев рис. 8, можно умозаключить, что наиболее оптимальным является тот маршрут, при прохождении по которому от  $m_s$  до  $m_f$  все элементы  $M$  задействованы только по одному разу.

Примем, что мощность  $E$  равна  $z$ , тогда выражение  $k + n = z!$  будет описывать общее количество всех возможных упорядоченных взаимодействий.

Учитывая, что маршрут от стартовой сущности к конечной по пространству графа может быть длительным, но не бесконечным (т.к. количество сущностей ограничено и маршрут не может проходить более одного раза по одной сущности), то для поиска наиболее короткого маршрута информационного потока триггеров/управленческих команд необходимо:

- 1) найти все возможные комбинации упорядоченных установок без повторений;
- 2) вычислить возможную длину информационного потока от стартовой до конечной сущности;
- 3) разработать инструмент выбора маршрута с наименьшим количеством сущностей в промежутки от стартовой до конечной.

Наименьший маршрут находится по формуле:

$$l_{\min sE} = \min \{l_{sE1}, l_{sE2} \dots l_{sEn}\}, \quad (8)$$

где  $sE$  – позиция стартовой сущности.

Отсюда следует, что с помощью данного выражения можно осуществить нахождение наименьшего маршрута, превратив выражение (8) в систему уравнений:

$$\begin{cases} l_{\min sE1} = \min\{l_{sE11}, l_{sE12} \dots l_{sE1n}\} \\ l_{\min sE2} = \min\{l_{sE21}, l_{sE22} \dots l_{sE2n}\} \\ \dots \\ l_{\min sEn} = \min\{l_{sEn1}, l_{sEn2} \dots l_{sEnn}\} \end{cases} \quad (9)$$

Затем выразить в матричном виде и учесть величину  $L_S$ , для каждого маршрута:

$$\begin{pmatrix} l_{sE11} & \dots & l_{sE1n} \\ \vdots & \ddots & \vdots \\ l_{sEn1} & \dots & l_{sEnn} \end{pmatrix} \begin{pmatrix} L_{s1} \\ \dots \\ L_{sn} \end{pmatrix} = \begin{pmatrix} M_{s1} \\ \dots \\ M_{sn} \end{pmatrix}. \quad (10)$$

Итоги расчета (10) покажут, что одна из величин больше других и, как следствие, имеет больший маршрут/сроки выполнения сущностей триггера/управленческой команды.

Алгоритм распределения информационного потока представляется в виде следующей последовательности:

1. Ожидание триггера/управленческой команды. Если действие получено, то осуществляется переход на следующий этап.
2. Сущностью выполняется, либо не выполняется условие перехода  $x$ , ввиду чего внутреннее состояние сущности меняется на  $a$  или  $b$ . Если действие осуществлено, то происходит переход на следующий этап.
3. Оракулом производится расчет  $M_{sn}$ .
4. Производится выбор наиболее короткого маршрута.
5. Происходит перерасчет режима системы. Если обнаружены отклонения от стабильно развивающегося режима, то «оракул» производит процедуру корректировки состояний сущностей.
6. Осуществляется вывод правил в форме функциональных зависимостей и информация по ним поступает на «оракула»
7. После выполнения триггера/управленческой команды неактивные сущности переходят в режим ожидания.

Во время работы «оракулом» используется квантовый алгоритм Гровера (для сохранения высоких метрик  $L_S$ ) [9 – 13]. Подобный алгоритм уже реализован в [14], что существенно упрощает реализацию ВР-схемы на квантовом компьютере и позволяет перейти к экспериментальным исследованиям.

## Результаты экспериментов

Для моделирования работы ВР-схемы разработана программа для QPU (Quantum Processor Unit). Схематичное изображение модели, выраженное с помощью QPU представлено на рис. 9.

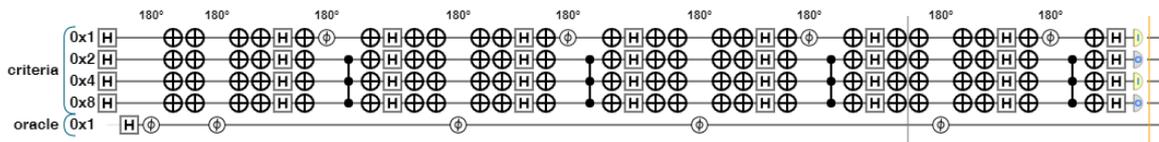


Рис. 9. ВР-схема, выраженная с помощью средств QPU  
Fig. 9. BP scheme expressed using QPU means

Моделировался процесс поиска (с помощью алгоритма Гровера) нахождения определённой (наиболее приоритетной) сущности, которая бы удовлетворяла вышеописанным критериям в неструктурированном пространстве сложной системы (неструктурированной базе данных).

Алгоритм работы данной модели выглядит следующим образом:

1. Инициализация кубитов: вначале появления триггера/управленческой команды, инициализируется пять кубитов, четыре из которых хранят информацию о критериях, по которым происходит поиск, а оставшийся – используется как «оракул» для фазового сдвига.

2. Переход кубитов в состояние «суперпозиции»: все кубиты критериев и оракула переводятся в суперпозицию с помощью преобразования Адамара. Кубит «оракула» также получает дополнительный фазовый сдвиг на 180 градусов ( $\pi$ -радиан), что меняет его фазу на противоположную.

3. Работа «оракула»: «оракул» производит изменение фазы состояния кубитов, отвечающих за критерии потока триггера/«управленческого импульса» с наивысшим приоритетом. При программировании данного алгоритма, оракул идентифицирует состояние  $|1101\rangle$  как наиболее приоритетное и прикладывает к нему фазовый сдвиг.

4. Работа оператора диффузии: данный оператор усиливает вероятность измерения состояния, фаза которого была инвертирована оракулом, и уменьшает вероятность всех остальных. Это достигается путем преобразования Адамара, применения операции NOT ко всем кубитам, применения фазового сдвига и повторения преобразования Адамара и операции NOT.

5. Выполнение алгоритма Гровера: поисковый алгоритм повторяет комбинацию применения оракула и оператора диффузии, а именно  $\sqrt{N}$ , где  $N$  это количество возможных состояний. Соответственно, с учётом примера в четыре кубита это будет  $\sqrt{2^4}$ .

6. Измерение: после выполнения необходимого числа итераций алгоритма, кубиты измеряются для получения конечного результата.

7. Вывод результата: измеренное значение показывает состояние с наивысшим приоритетом. Результат измерения выводится в консоль. В двоичном виде это будет представлять собой сочетание критериев, которое «оракул» определил, как приоритетное (рис. 10).

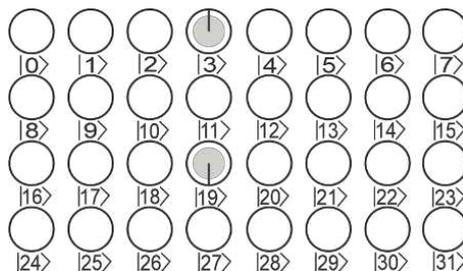


Рис. 10. Визуализация работы BP-схема, по определению оптимальной сущности  
 Fig. 10. Visualization of the BP-scheme operation, defining the optimal entity

На данной визуализации мы видим, что изменились состояния кубитов №3 и №19. Это означает, что данные кубиты находятся в равной суперпозиции (это видно по площади заполнения круга) состояний  $|00\rangle|00\rangle$  и  $|11\rangle|11\rangle$ . Положение стрелок на данных кубитах указывает сдвиг на  $0^0$  и  $180^0$  соответственно. В контексте алгоритма Гровера, стрелка вниз указывает на кубиты, чья фаза была инвертирована оракулом — это соответствует маркировке правильного ответа. В данном случае это означает, что кубиты №3 и №19 помечены, как соответствующие условию поиска. В результате выполнения алгоритма получено число «11», это означает, что после измерения кубиты критериев находились в состоянии, которое соответствует двоичному представлению числа 3 (в двоичной системе «11» равно десятичному числу 3). Это означает, что некий третий условный элемент списка/сценария, по которому должна перестроиться BP-схема, как имитационная модель сложной системы является приоритетным, согласно правилам ННС, на основе которых выведены критерии для «оракула».

### Заключение

Представленная в данной работе BP-схема может быть использована при имитационном моделировании сложных систем в области экономики или гибкого производства. Предлагаемое решение отличается использованием QPU в вычислениях поведения системы, а значит быстрым откликом на изменения состояния сущностей в системе. Помимо этого, использование поискового алгоритма Гровера с учетом выработанных правил ННС позволяет

совершенствовать сценарии и прогностические модели поведения сложной системы. В перспективе авторы научного коллектива планируют исследовать поведение «оракула» ВР-схем в сложных системах и определить дальнейшие способы его совершенствования.

#### Список источников:

1. Петров А.М., Попов А.Н. Разработка метода математического моделирования термодинамических процессов однофазных потоков наружных сетей теплоснабжения // *Строительство и техногенная безопасность*. – 2022. – № 26(78). – С. 59-63.
2. Петров А.М., Попов А.Н. Разработка интеллектуальной системы поддержки принятия решений по оценке состояния объектов системы теплоснабжения // *Автоматизация и информатизация ТЭК*. – 2023. – № 6(599). – С. 15-21.
3. Markov L., Saeedi M. Constant-optimized quantum circuits for modular multiplication and exponentiation. *Quantum Information and Computation* 12, 0361-0394, 2012.
4. Ambainis A. A nearly optimal discrete query quantum algorithm for evaluating NAND formulas, 2017.
5. Маслов И.О., Бачило А.О., Черкесова Л.В. Повышение быстродействия квантового алгоритма Гровера путем применения инверсии вокруг среднего // *Молодой исследователь Дона*. – 2019. – № 2(17). – С. 35-41.
6. Ulyanov S., Reshetnikov A., Tyatyushkina O. Modelling of Grover's quantum search algorithms: implementations of simple quantum simulators on classical computers // *Системный анализ в науке и образовании*. – 2020. – № 3. – С. 65-128.
7. A Fixed-Phase Quantum Search Algorithm with More Flexible Behavior / X Li et al. // *J. of Quantum Information Science*. – 2012. – No 2. – Pp. 28-34.
8. Johansson N., Larsson J.A. Quantum simulation logic, oracles, and the quantum advantage // *Entropy*. – 2019. – Vol. 21, No. 8.
9. Оценка возможностей классических компьютеров при реализации симуляторов квантовых алгоритмов / С. В. Ульянов, Н. В. Рябов, П. В. Зрелов и др. // *Программные продукты и системы*. – 2022. – № 4. – С. 618-630.
10. Quantum Simulators: Architectures and Opportunities / E. Altman, I. Siddiqi, K. R. Brown et al. // *PRX Quantum*. – 2021. – Vol. 2, No. 1. – P. 017003.
11. Quantum Computer Systems for Scientific Discovery / Y. Alexeev, D. Bacon, K. R. Brown et al. // *PRX Quantum*. – 2021. – Vol. 2, No. 1. – P. 017001.
12. Gyongyosi L., Imre S. Circuit Depth Reduction for Gate-Model Quantum Computers // *Scientific Reports*. – 2020. – Vol. 10, No. 1. – P. 11229.
13. A Modified Quantum Search Algorithm / H. Mehri-Dehnavi, H. Dashtianeh, H. Ya. Kuchaksaraei et al. // *International Journal of Theoretical Physics*. – 2018. – Vol. 57, No. 12. – P. 3668-3681.
14. Химено-Сеговиа М., Хэриган Н., Джонстон Э. Программирование квантовых компьютеров. Базовые алгоритмы и примеры кода. – СПб.: Питер, 2021. – 336 с.

#### References:

1. Petrov A.M., Popov A.N. Development of a Method for Mathematical Modelling of Thermodynamic Processes of Single-Phase Flows of External Heat Supply Networks. *Construction and Industrial Safety*. 2022;26(78):59-63.
2. Petrov A.M., Popov A.N. Development of a Decision-Making Intelligent Support System for Assessing the State of the Heat Supply System Facilities. *Automation and Informatization of the Fuel and Energy Complex*. 2023;6(599):15-21.
3. Markov L., Saeedi M. Constant-Optimized Quantum Circuits for Modular Multiplication and Expansion. *Quantum Information and Computation*. 2012;12:0361-0394.
4. Ambainis A. A Nearly Optimal Discrete Query Quantum Algorithm for Evaluating NAND Formulas; 2017.
5. Maslov I.O., Bachilo A.O., Cherkesova L.V. Speed Improvement of Quantum Grover's Algorithm Through the Use of Inversion About the Mean. *Young Don Researcher*. 2019;2(17):35-41.
6. Ulyanov S., Reshetnikov A., Tyatyushkina O. Modelling of Grover's Quantum Search Algorithms: Implementations of Simple Quantum Simulators on Classical Computers. *System Analysis in Science and Education*. 2020;3:65-128.
7. X Li, et al. A Fixed-Phase Quantum Search Algorithm With More Flexible Behaviour. *J. of Quantum Information Science*. 2012;2:28-34.
8. Johansson N., Larsson J.A. Quantum Simulation Logic, Oracles, and the Quantum Advantage. *Entropy*. 2019;21:8.
9. Ulyanov SV, Ryabov NV, Zrelov PV, et al. Evaluating the Capabilities of Classical Computers in Implementing Quantum Algorithm Simulators. *Software and Systems*. 2022;4:618-630.
10. Altman E, Siddiqi I, Brown KR, et al. Quantum Simulators: Architectures and Opportunities. *PRX Quantum*. 2021;2(1):017003.
11. Alexeev Y, Bacon D, Brown KR, et al. Quantum Computer Systems for Scientific Discovery. *PRX Quantum*. 2021;2(1):017001.
12. Gyongyosi L., Imre S. Circuit Depth Reduction for Gate-Model Quantum Computers. *Scientific Reports*. 2020;10(1):11229.
13. Mehri-Dehnavi H, Dashtianeh H, Kuchaksaraei HYa, et al. A Modified Quantum Search Algorithm. *International Journal of Theoretical Physics*. 2018;57(12):3668-3681.
14. Gimeno-Segovia M., Harrigan N., Johnston E. Programming Quantum Computers. Essential Algorithms and Code Samples. St. Petersburg: Peter; 2021.

### **Информация об авторах:**

#### **Беляев Игорь Сергеевич**

кандидат экономических наук, доцент, зав. кафедрой информационных систем и технологий «Заполярного государственного университета им. Н.М. Федоровского», <http://orcid.org/0000-0001-9149-4357>.

#### **Петров Алексей Михайлович**

Кандидат технических наук, доцент, зав. кафедрой электроэнергетики и автоматизации «Заполярного государственного университета им. Н.М. Федоровского», <http://orcid.org/0000-0002-3265-520X>.

#### **Попов Антон Николаевич**

старший преподаватель академического департамента индивидуальных образовательных траекторий ТюмГУ, старший преподаватель кафедры Энергообеспечения сельского хозяйства ГАУ Северного Зауралья, <http://orcid.org/0000-0003-0774-411X>.

### **Information about the authors:**

#### **Belyaev Igor Sergeevich**

Candidate of Economic Sciences, Associate Professor, Head of the Department “Information Systems and Technologies” of Fedorovsky Polar State University, <http://orcid.org/0000-0001-9149-4357>.

#### **Petrov Alexey Mikhailovich**

Candidate of Technical Sciences, Associate Professor, Head of the Department “Electric Power and Automation” of Fedorovsky Polar State University, <http://orcid.org/0000-0002-3265-520X>.

#### **Popov Anton Nikolaevich**

Senior lecturer of the academic department of individual educational trajectories of Tyumen State University, senior lecturer of the Department “Energy Supply of Agriculture” of the Northern Trans-Ural State Agricultural University, <http://orcid.org/0000-0003-0774-411X>.

**Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации.**

**Contribution of the authors: the authors contributed equally to this article.**

**Авторы заявляют об отсутствии конфликта интересов.**

**The authors declare no conflicts of interests.**

**Статья поступила в редакцию 13.01.2024; одобрена после рецензирования 03.02.2024; принята к публикации 14.02.2024.**

**The article was submitted 13.01.2024; approved after reviewing 03.02.2024; accepted for publication 14.02.2024.**

**Рецензент** – Горбунов А.Н., кандидат технических наук, доцент, Брянский государственный технический университет.

**Reviewer** – Gorbunov A.N., Candidate of Technical Sciences, Associate Professor, Bryansk State Technical University.