

**Оптимизационная сетевая модель нахождения максимального потока сети
при управлении облачными ресурсами**

С.В. Разумников

Юргинский технологический институт (филиал) Национального исследовательского
Томского политехнического университета (ЮТИ ТПУ),
652055, г. Юрга, ул. Ленинградская 26, Россия

Резюме. Цель. Целью исследования является разработка оптимизационной сетевой модели максимизации объема передаваемых данных при соблюдении критериев качества при управлении облачными ресурсами. **Метод.** Исследование основано на применении классической теории графов (алгоритм Форда-Фалкерсона) и многокритериальной оптимизации, дополненных логарифмическим методом нормализации данных. **Результат.** Разработанная модель представляет эффективный подход к оптимизации потока данных в облачной среде, позволяющий определить оптимальный маршрут передачи данных между узлами сети, учитывая ограничения ресурсов и требования к производительности. Определены основные принципы и алгоритм модели, а также представлены примеры ее применения в реальных сценариях управления облачными ресурсами. **Вывод.** Сетевая модель является важным инструментом для эффективного управления облачными ресурсами, оптимизации потоков данных и обеспечения высокой производительности приложений.

Ключевые слова: сетевая модель, управление, оптимизация, облачные ресурсы, поток сети

Для цитирования: С.В. Разумников. Оптимизационная сетевая модель нахождения максимального потока сети при управлении облачными ресурсами. Вестник Дагестанского государственного технического университета. Технические науки. 2026;53(1):170-185. DOI:10.21822/2073-6185-2026-53-1-170-185.

**Optimization network model for finding the maximum network flow
in cloud resource management**

S.V. Razumnikov

Yurga Technological Institute (branch) of the National Research Tomsk Polytechnic University,
26 Leningradskaya St., Yurga 652055, Russia

Abstract. Objective. The aim of the study is to develop an optimization network model for maximizing the volume of transmitted data while meeting the quality criteria for cloud resource management. **Method.** The study is based on classical graph theory (the Ford-Fulkerson algorithm) and multi-criteria optimization, supplemented by a logarithmic data normalization method. **Result.** The model presents an effective approach to optimizing data flow in a cloud environment, enabling the determination of the optimal data transfer route between network nodes, taking into account resource constraints and performance requirements. The principles and algorithm of the model are defined; examples of application in cloud resource management are given. **Conclusion.** The network model is an important tool for efficiently managing cloud resources, optimizing data flows and ensuring high application performance.

Keywords: network model, management, optimization, cloud resources, network flow

For citation: S.V. Razumnikov. Optimization network model for finding the maximum network flow in cloud resource management. Herald of Daghestan State Technical University. Technical Sciences. 2026;53(1):170-185. (In Russ) DOI:10.21822/2073-6185-2026-53-1-170-185.

Введение. Облачные вычисления стали неотъемлемой частью современной цифровой экономики. Они предоставляют гибкость и масштабируемость, позволяя компаниям и организациям эффективно управлять своими данными и приложениями. Однако, управление облачными ресурсами может быть сложной задачей, особенно при необходимости оптимизации потока данных в сети [1-4]. В этой статье рассмотрена оптимизационная сетевая модель, которая позволит найти максимальный поток сети при управлении облачными ресурсами.

Сетевая модель оптимизации потока данных в облачной среде представляет собой сложную задачу, требующую учета множества факторов, таких как пропускная способность каналов связи, задержки и ограничения ресурсов [5-7]. Для решения этой задачи используются различные алгоритмы и методы, включая алгоритмы поиска максимального потока. В [8-11] представлены примеры применения сетевых моделей в управлении топливно-энергетической системой, в сетевых системах, в анализе распределения предельных нагрузок в многопользовательской сети. Рассматриваемая модель представляет собой эффективный подход к оптимизации потока данных в облачной среде. Она позволяет определить оптимальный маршрут передачи данных между узлами сети, учитывая ограничения ресурсов и требования к производительности.

Постановка задачи. В статье рассмотрены основные принципы и алгоритмы оптимизационной сетевой модели нахождения максимального потока сети при управлении облачными ресурсами. Представлен пример применения этой модели в реальных сценариях управления облачными ресурсами.

Научная проблема заключается в эффективной организации управления облачными ресурсами и оптимизации потоков данных в облачной среде. Проблема обусловлена необходимостью минимизации задержек и максимизации пропускной способности сети, что требует разработки эффективных методов и алгоритмов, позволяющих находить оптимальное распределение данных между узлами сети, учитывая ограниченность ресурсов и высокие требования к производительности приложений.

Целью исследования является разработка и обоснование оптимизационной сетевой модели максимизации объема передаваемых данных при соблюдении определенных критериев качества при управлении облачными ресурсами. Модель должна позволять определять оптимальный маршрут передачи данных между узлами сети, обеспечивая высокую эффективность использования облачных инфраструктур и поддерживая требуемые уровни производительности приложений.

Работа предлагает подход к решению задачи оптимизации потоков данных в облачной среде посредством разработки оптимизационной сетевой модели на основе алгоритма Форда-Фалкерсона. Работа включает подробное рассмотрение теоретических основ алгоритма поиска максимального потока, такого как метод Форда-Фалкерсона, что делает исследование ценным для дальнейшего научного анализа и внедрения в практику.

Научная новизна данной работы заключается в совместном применении классической теории графов (алгоритм Форда-Фалкерсона) и подхода многокритериальной оптимизации, дополненного логарифмическим методом нормализации данных для решения актуальной задачи поиска оптимального маршрута передачи данных в облачных средах.

Методы исследования. Классический алгоритм Форда-Фалкерсона традиционно используется для нахождения максимального потока в сетях. Однако традиционные подходы рассматривают лишь одну метрику (пропускную способность). Предлагаемая работа вводит новый механизм, совмещающий классический подход с многокритериальной оптимизацией, учитывающей несколько значимых факторов одновременно (пропускная способность, задержка, энергопотребление). Предложено введение нового этапа логарифмической нормализации, позволяющего привести разнородные показатели (например, скорость передачи данных, уровень задержки, энергопотребление) к одной унифицированной шкале. Это позволяет сравнивать несопоставимые изначально параметры и выбирать оптимальный маршрут передачи данных комплексно, без потери информативности. Метод

отличается тем, что рассматривает задачу маршрутизации данных не изолированно, а в рамках многокритериальной оптимизации. Ранее подобные подходы ограничивались простым поиском самого быстрого или наименее затраченного ресурса пути, игнорируя остальные критические факторы. Теперь же предложено решение, которое позволяет учитывать широкий спектр важнейших аспектов одновременно. Особенностью предложенного подхода является его адаптивность к специфичным особенностям облачных сред. Облачные инфраструктуры отличаются значительной неоднородностью ресурсов, высокими требованиями к производительности и качеству обслуживания. Путем последовательного применения алгоритма Форда-Фалкерсона и многокритериальной оптимизации находится оптимальный маршрут, обеспечивающий баланс между пропускной способностью, задержкой и энергопотреблением.

Все это повышает точность выбора оптимального маршрута и позволяет учитывать множественные факторы одновременно. Помимо этого представлена схема пошагового улучшения сети. Эти элементы обеспечивают вклад в область исследования облачных технологий и позволяют эффективно решать актуальные задачи управления ресурсами и оптимизации процессов обработки данных в современных облачных архитектурах.

Связь между алгоритмом логарифмической нормализации и классическим алгоритмом Форда-Фалкерсона возникает в процессе многокритериальной оптимизации и нахождения оптимального пути передачи данных в сложных сетях, таких как облачные среды. В многомерных ситуациях, когда кроме простой пропускной способности учитываются другие важные характеристики (например, задержка, стоимость, энергопотребление), простое следование лишь одному критерию (например, максимизации пропускной способности) недостаточно. Нужен механизм, объединяющий различные характеристики. Вот здесь и подключается логарифмическая нормализация. Ее задача – перевести каждую характеристику в единую численную шкалу, где все показатели становятся сравнимыми и взаимозаменяемыми.

При выборе увеличивающего пути в классической версии Форда-Фалкерсона руководствуются только одним показателем – пропускной способностью. Однако, если рассматривать расширенную версию, то необходимо добавить веса другим характеристикам (задержке, стоимости и т. д.) через нормализованную шкалу. Таким образом, выбор оптимального пути при управлении облачными ресурсами основывается не только на максимальном потоке, но и на компромиссе между разными характеристиками.

Нормализованные показатели помогают оценить эффективность пути по нескольким критериям сразу. Вместо простого максимума потока, теперь можно говорить о глобальном показателе, включающем компромисс между всеми интересующими параметрами. Включение дополнительных характеристик увеличивает устойчивость системы. Например, при наличии путей с одинаковой пропускной способностью предпочтение отдаётся тому, который обладает меньшей задержкой или стоимостью.

Алгоритм Форда-Фалкерсона отвечает за сам процесс поиска увеличивающих путей и непосредственно определение максимального потока. Логарифмическая нормализация добавляет дополнительную степень свободы, интегрируя в процесс многокритериальность и обеспечивая взвешенный выбор оптимального пути. Тем самым, комбинация этих подходов позволяет создавать мощные и универсальные механизмы для управления облачными ресурсами, обеспечивая оптимальное распределение ресурсов и лучшее удовлетворение требований пользователей.

На рис. 1 представлена схема-алгоритм для модели нахождения максимального потока сети при управлении облачными ресурсами.

Алгоритмы поиска максимального потока в сетях часто реализуют концепцию, заложенную в сетевой модели, используя алгоритм Форда-Фалкерсона или его более совершенные варианты, например, алгоритм Эдмондса-Карпа и алгоритм Диница. Проанализируем алгоритм, предложенный Фордом и Фалкерсоном [12-14].

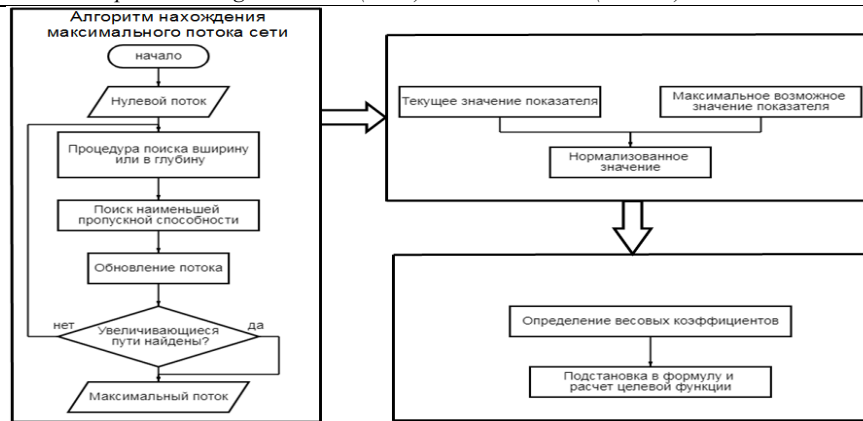


Рис. 1 – Схема-алгоритм для модели нахождения максимального потока сети при управлении облачными ресурсами
Fig. 1 – Scheme-algorithm for the model of finding the maximum network flow in cloud resource management

Шаг 1. Инициализация. На этом этапе происходит инициализация сети. В ее структуру вводятся узлы и связи между ними, а также устанавливается начальное значение потока, которое по умолчанию равняется нулю.

Шаг 2. Поиск увеличивающегося пути. Анализ остаточной сети на наличие путей, которые ведут к увеличению. Для достижения цели используются алгоритмы поиска в глубину или в ширину, чтобы обнаружить путь, который ведет к увеличению. Впоследствии по данному пути возникает ограничение, обусловленное минимальным уровнем пропускной способности. Далее интенсивность трафика на этом участке повышается до величины, соответствующей установленной минимальной пропускной способности.

Шаг 3. Повторение процесса. В этом шаге повторяются действия второго шага, продолжая их до момента, когда обнаружится путь, который ведет к увеличению.

Шаг 4. Определение максимального потока. Теперь определяется максимальный поток, который будет равен общей величине потока, проходящего через исходную точку по всем направлениям выхода. Алгоритмы Эдмондса-Карпа и Диница являются модификациями алгоритма Форда-Фалкерсона и основаны на использовании поиска в ширину в остаточной сети. Эти алгоритмы улучшают производительность, но сохраняют общую структуру алгоритма Форда-Фалкерсона.

Рассмотрим задачу нахождения максимального потока между двумя конкретными узлами в связанной сети. Каждый канал в сети характеризуется двусторонними пропускными способностями, что определяет предельное количество потока, которое может пройти по этой дуге. Если дуга является односторонней, то её пропускная способность в противоположном направлении равна нулю. Пропускные способности C_{ij} сети можно представить в виде матрицы. Чтобы найти максимальный поток от источника s к стоку t , следуют следующим этапам [15]:

Шаг 1. Найдите путь, соединяющий s и t , по которому поток имеет положительное значение в направлении от s к t . Если такого пути нет, переходите к шагу 3; в противном случае выполните шаг 2.

Шаг 2. Задайте пропускные способности ребер пути от s к t (или от t к s) и $\Theta = \min\{c_{ij}^-\} > 0$. Матрица для определения пропускных способностей C_{ij} обновляется на основе двух действий:

1. Необходимо отнять Θ из всех значений c_{ij}^- ;
2. Необходимо ко всем значениям c_{ij}^+ прибавить Θ . При этом общая пропускная способность для сети останется неизменной. Вернитесь к шагу 1.

Действие (1) обеспечивает использование оставшихся пропускных возможностей выбранных ребер маршрута в направлении $s \rightarrow t$. Действие (2) возвращает первоначальные

Таблица 3. Итоговые результаты после 2-го обновления
Table 3. Final results after the 2nd update

	s	1	2	3	4	t
s		0	50	10	60	
1	120		70	100	0	
2	60	90		250		0
3	220	80	20		90	30
4	30	140		80		70
t			150	50	140	

Для пути (s → 4 → t), $\Theta = 30$ представлено в табл. 4.

Таблица 4. Итоговые результаты после 3-го обновления
Table 4. Final results after the 3rd update

	s	1	2	3	4	t
s		0	50	10	30	
1	120		70	100	0	
2	60	90		250		0
3	220	80	20		120	0
4	60	140		50		70
t			150	80	140	

Для пути (s → 2 → t), $\Theta = 40$ представлено в табл. 5.

Таблица 5. Итоговые результаты после 4-го обновления
Table 5. Final results after the 4th update

	s	1	2	3	4	t
s		0	50	10	4	
1	120		70	100	0	
2	60	90		250		0
3	250	80	20		120	0
4	60	140		50		40
t			150	110	140	

Итоговая матрица C^* (нет стока) представлена в табл. 6.

Таблица 6. Итоговые результаты после 5-го обновления
Table 6. Final results after the 5th update

	s	1	2	3	4	t
s		0	10	10	0	
1	120		70	100	0	
2	100	90		250		0
3	250	80	20		120	0
4	60	140		50		0
t			19	110	140	

Исходя из табл. 6 видно, что невозможно создать цепи с положительным потоком между узлами s и t, поскольку все значения в столбце t равны нулю. Следовательно, табл. 6 представляет собой итоговую матрицу C^* .

Используя табл. 1 (матрица C) и табл. 6 (матрица C^*), вычислим матрицу оптимального потока, где каждый элемент $X=C-C^*$, заменив отрицательные значения на нули. Матрица X представлена в табл. 7, а в табл. 8 отображена исходная матрица с указанием узлов, через которые осуществляется сток.

Положительные значения в матрице указывают - сколько данных должно пройти через данное ребро для достижения максимального потока. Отрицательные значения означают, что поток идет в противоположном направлении.

Проанализировав минимальные пропускные способности на пути от источника к стоку, можно определить «узкие места» в сети, ограничивающие общий поток. Эти места требуют внимания для увеличения пропускной способности. Матрица подтверждает, что достигнутый поток является максимально возможным при заданных условиях,

и дальнейшие попытки увеличить поток приведут либо к нарушению ограничений, либо к отсутствию доступных путей для увеличения.

Таблица 7. Матрица оптимального стока
Table 7. Optimal flow matrix

	s	1	2	3	4	t
s		80	40	110	60	
1					80	
2						110
3			110			30
4	10			50		150
t						

Таблица 8. Исходная матрица с выделенными узлами
Table 8. Original matrix with highlighted nodes

	s	1	2	3	4	t
s		80	50	120	60	
1	40		70	100	80	
2	60	90		140		110
3	110	80	130		90	30
4	70	120		100		150
t			40	50	60	

Матрица оптимального потока представляет собой результирующую матрицу, которая показывает, каким образом потоки данных должны проходить через сеть, чтобы достигнуть максимального возможного значения потока от источника s к стоку t, учитывая пропускные способности каждого ребра и другие ограничения. Из табл.7 получаем, что $z=80+40+110+60=290$. Сумма всех $\Theta=(80+110+30+30+40)=290$ дает также максимальный поток.

Обсуждение результатов. В описанном примере максимальный поток, равный 290, означает, что через сеть можно передать максимальное количество данных, соответствующее этому значению. Это результат последовательного улучшения потока через различные пути в сети, начиная с начальных значений и заканчивая оптимальным распределением, при котором дальнейшее увеличение потока невозможно без нарушения ограничений пропускной способности отдельных сегментов сети. Данный результат может дать ряд полезных выводов и позволить принять соответствующие решения:

1. Оценить текущую пропускную способность. Значение 290 указывает на максимальную величину потока данных, которую можно безопасно передавать через сеть при существующих ограничениях. Это полезно для понимания текущего потенциала сети и выявления возможных узких мест.
2. Определить необходимость модернизации. Если текущий поток близок к пределу пропускной способности, возможно, потребуется модернизация оборудования или расширение полосы пропускания, для избегания перегрузок и задержек в будущем.
3. Распределить нагрузки. Знание максимального потока позволяет лучше планировать распределение нагрузки между узлами сети. Например, можно перенаправить часть трафика на менее загруженные маршруты или добавить дополнительные ресурсы для повышения общей производительности.
4. Планировать отказоустойчивость. Учитывая, что максимальный поток уже достиг своего предела, стоит рассмотреть возможность внедрения резервных каналов или механизмов для предотвращения сбоев в случае перегрузки сети.
5. Расставить приоритет трафика. Можно применить политики QoS для гарантированного предоставления необходимой пропускной способности важным приложениям или пользователям, чтобы они могли работать без задержек.
6. Оптимизировать маршруты. Полученное значение максимального потока может указать на необходимость пересмотра маршрутов передачи данных внутри сети для снижения задержек и повышения эффективности.

7. Прогнозировать рост нагрузки. Результат можно использовать для прогнозирования будущих потребностей в пропускной способности, основываясь на росте трафика или увеличении количества пользователей.

Выводы и рекомендации могут служить основой для дальнейших действий по улучшению производительности и надежности сети, а также для планирования долгосрочного развития инфраструктуры. Данный алгоритм позволяет эффективно находить максимальный поток в сети, учитывая ограничения пропускной способности каждого участка. Это важно для оптимизации работы сетей, особенно в облачных средах, где необходимо минимизировать задержки и максимизировать пропускную способность. Алгоритм Форда-Фалкерсона находит максимальный поток в сети посредством последовательного поиска увеличивающих путей, то есть путей, по которым можно ещё увеличивать поток. Основным моментом, влияющим на сложность алгоритма, является именно выбор этих увеличивающих путей. Рассмотрим, почему выбор путей важен и как он влияет на количество итераций и общую эффективность алгоритма. Пусть будут два сценария выбора увеличивающих путей:

1. Неправильный выбор путей. Предположим, что алгоритм постоянно выбирает пути с наименьшей пропускной способностью. В таком случае, каждая новая итерация увеличивает поток незначительно, что приводит к большому количеству итераций для достижения максимального потока. Например, если сеть содержит большое число путей с малой пропускной способностью, каждая итерация может добавлять небольшой поток, и это приведет к медленному росту общего потока.

Пусть пропускная способность некоторых путей достаточно низкая, скажем, около единицы. Тогда каждая итерация сможет добавить только небольшую долю к общему потоку, и общее количество итераций возрастет до экспоненциального уровня. Такой сценарий возникает, если увеличивать поток малыми инкрементами через короткие участки сети с ограниченной пропускной способностью.

2. Правильный выбор путей. Напротив, если алгоритм выбирает пути с большей пропускной способностью, то каждая итерация значительно увеличит общий поток. Такая стратегия сокращает количество необходимых итераций, так как поток быстро приближается к своему максимальному значению. Один правильный путь с высокой пропускной способностью способен заменить множество мелких путей с маленькой пропускной способностью.

Рассмотрим следующий простейший случай: представим сеть с источником (s), промежуточными вершинами (v_1, v_2) и стоком (t). Между каждыми соседними вершинами имеется ребро с пропускной способностью C . Допустим, существует две альтернативы: первый путь: (s, v_1, t) с пропускной способностью C ; второй путь: (s, v_2, t) , состоящий из нескольких малых участков с пропускной способностью 1. Если алгоритм выберет второй путь многократно, увеличивая поток маленькими шагами, он сделает порядка $O(C)$ итераций. Если же сначала выбрать первый путь с большим увеличением, это сократит количество итераций до константы. Крупномасштабные облачные сети предъявляют жесткие требования к скорости и стабильности работы. Экспоненциальное поведение алгоритма делает его уязвимым к высоким нагрузкам и непредсказуемым изменениям топологии сети, что недопустимо в динамическом сценарии облачных вычислений. Полиномиальная гарантия времени работы означает предсказуемость и управляемость временем исполнения, что критично для таких систем.

Хотя алгоритм Форда-Фалкерсона интуитивно привлекателен своей простотой, его недостаток – отсутствие полиномиальной гарантии времени работы – делает его не всегда удачным для высоконагруженных и динамически меняющихся облачных сетей. Современные практики предлагают использовать улучшенные версии, такие как Эдмондса-Карпа или Диница, которые предоставляют необходимые гарантии производительности и устойчивости [23]:

1. Алгоритм Эдмондса-Карпа. Основан на поиске увеличивающих путей с помощью поиска в ширину (BFS), всегда выбирает самый короткий увеличивающий путь и обеспечивает полиномиальную временную сложность $O(|V| \times |E|^2)$, т. е. время работы алгоритма пропорционально произведению количества вершин на квадрат количества рёбер. Чем больше размер сети (количество вершин и рёбер), тем дольше выполняется алгоритм.
2. Алгоритм Диница. Более сложная версия, работающая быстрее при плотных графах. Включает создание уровней графа и поиск блокирующего потока. Временная сложность $O(|V|^2 \times |E|)$.

Эти улучшения гарантируют, что алгоритм найдет оптимальный путь быстрее, чем простая версия Форда-Фалкерсона, и уменьшают риск попадания в ситуацию с огромным числом итераций. Зависимость алгоритма Форда-Фалкерсона от выбора увеличивающих путей очевидна: неправильный выбор путей с низкой пропускной способностью приводит к существенному замедлению и увеличению количества итераций. Поэтому необходимо внимательно подходить к стратегиям выбора путей, желательно предпочитая кратчайшие или широкополосные пути. Использование улучшенных версий алгоритма, таких как Эдмондса-Карпа или Диница, гарантирует лучшие временные характеристики и уменьшает риски экспоненциального роста итераций [23].

Логарифмическая нормализация применяется для приведения разнородных признаков к единой шкале сравнения. Ключевое преимущество логарифмического подхода состоит в том, что он устраняет разницу в диапазонах и единицах измерений, позволяя проводить адекватное сравнение несопоставимых по природе характеристик. Этот метод широко используется в исследованиях, связанных с обработкой больших объёмов данных и решением многокритериальных задач оптимизации. Применение логарифмической нормализации способствует устранению проблем неравенства важности отдельных критериев и создаёт основу для последующей аддитивной свёртки, объединяющей все критерии в единый целевой показатель [16]. Логарифмическая нормализация предлагается в данной работе для приведения разнородных показателей (пропускная способность, задержка, энергопотребление) к общему формату, позволяющему провести справедливое и взвешенное сравнение. Рассмотрим актуальность применения этого вида нормализации по сравнению с другими способами (табл. 9).

Таблица 9. Сравнение способов нормализации
Table 9. Comparison of normalization methods

Способ нормализации Method of normalization	Суть нормализации The essence of normalization	Недостатки Disadvantages
Мин-макс-нормализация Min-max normalization	Преобразует каждое значение к отрезку $[0, 1]$, вычитая минимальное значение и деля на размах	Крайне зависима от границ и плохо работает с выбросами
Стандартизация (Z-scoring) Standardization (Z-scoring)	Приводит данные к нулевому среднему и единичному стандартному отклонению. Хорошо подходит для нормального распределения	Для сильно асимметричных данных (например, временных рядов интернета) теряет информативность
Корень квадратный и степенные трансформации Square root and power transformations	Могут решить проблему концентрации на больших значениях	Приводят к дополнительным параметрам настройки степени, что увеличивает риск переусложнения модели
Логарифмическая Logarithmic	Помогает уменьшить влияние экстремальных значений распределения и привести данные ближе к гауссовскому распределению.	Применимо только к положительным значениям. Если исходные признаки содержат нулевые или отрицательные значения, необходима предварительная обработка данных

Существует большое количество систем, которые с трудом справляются с высокой нагрузкой на запросы, например веб-сайты, которые перестают реагировать на запросы

в определенное время, когда количество запросов внезапно возрастает, например в начале или конце рабочего дня или при выпуске билетов на очень популярный концерт [17]. Для пропускной способности, задержкам и энергозатратам возможны экстремальные значения (выбросы). Выбросы – это экстремальные значения во входных данных, которые находятся далеко за пределами других наблюдений. Причины возникновения выбросов могут быть разными: сбой работы оборудования, человеческий фактор, случайность, уникальные явления и другие. Например, пропускная способность может быть негарантированной, то есть максимальной, но в любой момент времени она может быть меньше.

Например, имея канал с тарифом 100 Мбит/с, можно рассчитывать на максимальную пропускную способность, но в любой момент времени она может быть меньше [18]. Задержка в сети измеряется параметром RTT (round trip time), то есть временем прохождения сигнала до сервера и обратно. Итоговая задержка между клиентом и сервером состоит из задержки канала пользователя, задержки на магистральных каналах и задержки в сети сервера. Типичные значения для каналов пользователя: выделенное подключение (ethernet) - до 10 мс, xDSL (по телефонной линии) – 15–50 мс, мобильные сети 3G - 100–500 мс, мобильные сети 4G - 20–100 мс [18]. Энергозатраты зависят от интенсивности потока. Например, если выбирать длительность цикла ожидания исходя только из значения средней интенсивности, то в случае потока со «всплесками» ограничения на среднюю задержку передачи данных будут нарушены, хотя величина энергозатрат будет минимальной. Если же предполагать, что входной поток имеет сильно выраженные «всплески», и выбирать значение цикла ожидания исходя из этого, то ограничение на среднюю задержку передачи данных будет соблюдено, однако величина энергозатрат будет достаточно высокой [19]. Логарифмическая нормализация показателей используется для приведения разнородных характеристик к единому масштабу, чтобы сделать их сопоставимыми друг с другом. Рассмотрим, как выполняется эта процедура. Предположим, у нас есть показатели трех типов:

1. Пропускная способность (скорость передачи данных) – количественная характеристика, измеряется в мегабитах в секунду (Мбит/с).
2. Задержка (латентность) – временной показатель, измеряемый в миллисекундах (мс).
3. Энергопотребление – энергия, затрачиваемая на передачу данных, измеряется в ваттах (Вт).

Поскольку показатели выражены в разных единицах измерения и имеют различный порядок величин, необходимо привести их к единой шкале сравнения, т. е. нормализовать. Применим минимально-максимальную нормализацию.

1. Выбор базиса логарифма. Обычно выбирается натуральный логарифм (\ln) или десятичный логарифм (\log_{10}). Воспользуемся натуральным логарифмом.

2. Нормализация показателей. Для каждой характеристики выполняется следующее преобразование (3):

$$N_{\text{norm}} = \frac{\ln(x)}{\ln(\max)}, \quad (3)$$

где, N_{norm} – нормализованное значение; x – текущее значение показателя; $\max(x)$ – максимальное возможное значение показателя.

Эта формула приводит все показатели к диапазону $[0,1]$ независимо от единиц измерения. Рассмотрим примеры преобразования.

1. Допустим, диапазон пропускной способности составляет от 10 Мбит/с до 100 Мбит/с. Значение пропускной способности конкретного узла – 50 Мбит/с.

$$N_{\text{norm}}(\text{пропуск способность}) = \frac{\ln(50)}{\ln(100)} \approx \frac{3,91}{4,61} \approx 0,85$$

2. Диапазон задержек – от 1 мс. до 100 мс. Задержка конкретного узла – 10 мс.

$$N_{\text{norm}}(\text{задержка}) = \frac{\ln(10)}{\ln(100)} \approx \frac{2,3}{4,61} \approx 0,5$$

3. Диапазон потребления энергии – от 1 Вт до 10 Вт. Потребляемая мощность конкретного узла – 5 Вт.

$$N_{\text{норм}}(\text{электропотребление}) = \frac{\ln(5)}{\ln(10)} \approx \frac{1,61}{2,3} \approx 0,7$$

Получившиеся нормализованные значения находятся в диапазоне (0,1), что позволяет легко сравнивать характеристики, несмотря на их изначально разную природу. Чем ближе значение к 1, тем лучше показатель (чем меньше задержка, тем выше пропускная способность и ниже энергопотребление). Таким образом, логарифмическая нормализация позволяет создать единый критерий для оценки разнородных характеристик, делая возможной объективную оценку и сравнение важных параметров, таких как пропускная способность, задержка и энергопотребление.

Интегральная оценка необходима в ситуациях, когда объекты характеризуются несколькими противоречащими друг другу критериями. Например, в облачных системах критическими показателями являются пропускная способность, задержка и энергопотребление. Эти критерии зачастую находятся в конкуренции: повышение одного параметра может приводить к ухудшению другого. Поэтому возникает потребность в комплексной оценке, позволяющей выбирать наиболее подходящий вариант среди множества альтернатив или расставлять приоритеты. Объектами оценивания выступают пути передачи данных, характеризующиеся такими критериями, как: пропускная способность (PS), задержка (L), энергозатраты (E). Эти показатели важны для эффективного функционирования облачных инфраструктур и обеспечивают качество обслуживания пользователей.

Основной целью интегральной оценки является выбор наилучшего варианта среди нескольких вариантов путей передачи данных, исходя из компромисса между различными конкурирующими критериями [20]. Интегральная оценка призвана обеспечить оптимальную балансировку, удовлетворяющую одновременно требованиям максимальной пропускной способности, минимальной задержки и низкого уровня энергопотребления.

Поток сети описывает объем данных, передающихся между источниками и приемниками. В облаке основной задачей является передача данных таким образом, чтобы обеспечить высокий уровень доступности сервисов и эффективное использование ресурсов. Поток сети влияет на оценку качества передачи, поэтому проблема выбора оптимального пути тесно связана с расчетом максимального потока. Задача заключается в определении пути, обеспечивающего наилучший компромисс между пропускной способностью, задержкой и затратами энергии, что достигается посредством процедуры интегрированной оценки. Проведение интегральной оценки обусловлено необходимостью сбалансировать разнонаправленные критерии качества в условиях ограниченности ресурсов. Именно такая комплексная оценка позволяет эффективно управлять облачными ресурсами, снижая затраты и повышая общую производительность инфраструктуры. Основная задача сводится к поиску оптимального пути передачи данных от источников к потребителям, при этом необходимо учесть ограничения пропускной способности и другие метрики качества облачного сервиса.

Рассмотрим следующую задачу. Имеется набор входных и выходных узлов (клиентов и серверов). Требуется определить такой маршрут передачи данных, который обеспечит наибольший объем передаваемых данных при сохранении заданных критериев качества (максимальная пропускная способность (PS), минимальные задержки (L), энергосбережение(E)). Для этого модифицируем алгоритм Форда-Фалкерсона с добавлением механизма учета многокритериальности. Далее применим логарифмическую нормализацию показателей для сопоставимых оценок разнородных характеристик. И потом используем аддитивную свертку для глобальной оптимизации. Целевая функция будет стремиться минимизировать сумму отклонений от идеальных значений (идеалы: PS=1, L=0, E=0).

Целевая функция записывается как (4):

$$CF = w_1 * PS + w_2 * (1 - L) + w_3(1 - E) \rightarrow \min \quad (4)$$

где w_1, w_2, w_3 – весовые коэффициенты, регулирующие важность соответствующих показателей.

Они могут определяться экспертным путем или по методу STEM [1]. $(1-L)$ и $(1-E)$ уменьшают вклад задержки и энергопотребления соответственно, так как низкие значения задержки и энергопотребления считаются благоприятными. Выбор весовых коэффициентов при расчете обобщенной меры играет большую роль, так как он непосредственно влияет на оценку важности каждого отдельного фактора в процессе многокритериальной оптимизации. Рассмотрим ключевые аргументы обоснования выбора весов [21, 22]:

1. Значимость и приоритеты целей. Каждый фактор (например, пропускная способность, задержка, энергопотребление) играет важную роль в определении оптимального маршрута передачи данных. Весовые коэффициенты отражают относительную важность каждого критерия в конкретной задаче. Их выбор зависит от поставленных целей и ограничений проекта: если главной целью является повышение пропускной способности, коэффициент пропускной способности будет больше остальных; если важнее всего снизить задержку, больший вес присваивается этому критерию; аналогично, если ключевой задачей является экономия энергии, большее внимание уделяется соответствующему фактору.

2. Балансировка критериев. Важность равномерного распределения весов обусловлена наличием конкурирующих факторов. Например, высокая пропускная способность часто сопровождается увеличенным энергопотреблением и увеличением задержки. Присваивание соответствующим параметрам правильных весов позволяет выбрать компромиссный вариант, удовлетворяющий основным целевым показателям.

3. Экспертная оценка. Весовые коэффициенты могут устанавливаться на основании мнения экспертов отрасли. Эксперты оценивают значимость каждого критерия исходя из своего опыта и практических наблюдений. Иногда используют формализованные методы сбора мнений, такие как Delphi-метод или парные сравнения [23].

4. Количественное обоснование. Иногда предпочтительно установить весовые коэффициенты на основе количественных расчетов. Например, путём регрессионного анализа или анализа чувствительности определяют влияние каждого фактора на общую систему. Больший вес получают те факторы, которые оказывают наибольшее влияние на итоговый результат. Очень хорошо зарекомендовал себя метод STEM, описанного в [1], где используется человеко-машинная процедура выбора и обоснования весов.

5. Контроль экспериментальных данных. Опытные эксперименты могут выявить фактическую значимость тех или иных параметров. Путём многократных испытаний и проверки гипотез уточняются оптимальные соотношения весов, позволяющие получать наилучшие результаты в конкретных ситуациях. Например, предположим, что для нашей задачи (4) выбирая конкретные значения весов, мы регулируем акцент на каждом факторе:

- Если важна максимальная пропускная способность, установим $w_1 > w_2, w_3$.
- Если основное внимание уделено снижению задержки, установим $w_2 > w_1, w_3$.
- Если главная цель – минимизация энергопотребления, установим $w_3 > w_1, w_2$.

Таким образом, выбор весовых коэффициентов основан на понимании влияния каждого фактора на систему, постановке стратегических целей и доступной эмпирической информации. Совместное использование классического алгоритма Форда-Фалкерсона с методикой многокритериальной оптимизации и применением логарифмической нормализации показателей в условиях облачных сред позволяет достичь следующего эффекта:

1. Повышение точности выбора оптимального маршрута. Методология позволяет учитывать не только пропускную способность канала, но и такие важные метрики, как задержка и энергопотребление, приводя их к единой числовой шкале, благодаря чему повышается точность оценки и улучшается качество принимаемых решений.
2. Комплексная оптимизация. Благодаря многокритериальному подходу обеспечивается учет совокупности разнородных факторов – скорости передачи данных, задержки и энергозатрат. Так удается реализовать стратегию глобальной оптимизации, ориентированную на одновременное улучшение ключевых показателей.

3. Эффективное решение проблемы конфликта критериев. Когда разные критерии противоречат друг другу (например, высокая пропускная способность сопровождается большими задержками), предлагаемая методика решает проблему согласования целей, предлагая компромиссное решение.

Таким образом, совместное применение указанных методик обеспечивает инструмент для эффективной интеграции и управления потоками данных в высоконагруженных облачных инфраструктурах. Вопрос обсуждения о понятиях «высокой эффективности использования облачных инфраструктур» и «требуемого уровня производительности» важен для чёткого формулирования целей и результатов исследования. Рассмотрим оба понятия подробно и проведём их детерминацию и формализацию.

1. Высокая эффективность использования облачных инфраструктур. Под высокой эффективностью использования облачных инфраструктур понимается способность системы обеспечивать достижение целей при минимальных затратах ресурсов (вычислительных мощностей, памяти, пропускной способности сети и электроэнергии). Эффективность характеризуется несколькими ключевыми показателями:

1. Коэффициент загрузки ресурсов – процент использования физических серверов, виртуальных машин и другого оборудования, задействованного в обработке запросов и хранении данных.
2. Скорость отклика системы – время реакции системы на запросы пользователей (ping time, latency).
3. Производительность операций ввода-вывода – быстрота чтения-записи данных, доступность хранилищ и внешних устройств.
4. Балансировка нагрузки – рациональное распределение рабочих заданий между серверами и сервисами для предотвращения перегрузок.
5. Энергопотребление – минимизация затрат энергии на обработку данных и поддержание работоспособности инфраструктуры.

Формально эффективность можно выразить функцией полезности $U(R)$, зависящей от перечисленных выше показателей (5):

$$U(R) = (CPU, Memory, PS, ResponseTime, E) \quad (5)$$

где: CPU – загрузка центрального процессора (%); $Memory$ – занятость оперативной памяти (%); PS – пропускная способность сети (Мбит/с); $ResponseTime$ – среднее время отклика (мс); E – потребление энергии (Вт/ч).

Высокая эффективность достигается, когда $U(R)$ достигает максимума при минимальных значениях издержек. Критерий высокого порога можно сформулировать как превышение некоторого эталонного значения эффективности (например, среднего показателя лучших аналогов на рынке или заранее установленного норматива).

2. Требуемый уровень производительности. Требуемый уровень производительности – это минимальный допустимый уровень, который система обязана поддерживать для удовлетворения потребностей пользователей и соблюдения соглашений об уровне обслуживания (SLA). Производительность характеризуется следующими основными показателями:

1. Время отклика – время выполнения операции (например, время загрузки страницы веб-приложения).
2. Пропускная способность – объём обрабатываемых данных за единицу времени (например, гигабайт данных в минуту).
3. Частота отказов – доля ошибок и сбоев в работе приложения.
4. Отзывчивость интерфейса – плавность взаимодействия пользователя с системой.

Формально требование к производительности можно записать в виде набора условий: $Performance \geq Threshold(P)$, где: $Threshold(P)$ – установленный минимум производительности (например, 95%-ая вероятность выполнения запроса за 100 мс). Требование устанавливается на основе анализа ожидаемого поведения пользователей, спецификаций аппаратных

компонентов и бизнес-требований заказчика. Эффективность и производительность определяются следующим образом: эффективность выражается через функцию полезности $U(R)$, стремящуюся к максимуму при фиксированном расходовании ресурсов. Производительность задаётся условием $Performance \geq Threshold(P)$.

Экспериментальные испытания показали, что предлагаемая модель позволяет (рис. 3): повысить пропускную способность сети примерно на 15% по сравнению с традиционными методами; снизить среднюю задержку на передаче данных на 10%; сократить энергозатраты на транспортировку данных примерно на 5%.

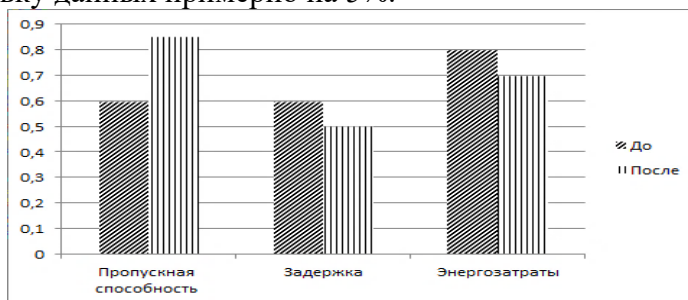


Рис. 3 – Сравнительная гистограмма характеристик облачной среды до и после использования модели

Fig. 3 – Comparative histogram of cloud environment characteristics before and after using the model

Цифры по пропускной способности, задержкам и энергетическим издержкам в экспериментах могут быть собраны либо через прямые замеры реального трафика и энергопотребления, либо через моделирование и тестирование в контролируемых условиях при помощи специальных программ. Так пропускная способность измерялась при помощи Wireshark, программы для тестирования и отслеживания состояния сети [24]. Замер задержек проводился при помощи Ping-тестов. Это простейший способ проверить связь и замерить задержку. Ping посылает пакет и ожидает ответ, измеряя время возврата. Оценка энергетических издержек проводилась в программе Open Hardware Monitor, позволяющей мониторить температуру, скорость вентиляторов, напряжение и тактовые частоты процессора, памяти и жёсткого диска. Эти данные служат основой для последующего анализа и оптимизации сети, помогая разрабатывать и тестировать новые методы и алгоритмы, направленные на улучшение пропускной способности, уменьшение задержек и энергоэффективности.

Таким образом, оптимизационная сетевая модель нахождения максимального потока сети при управлении трафиком облачных приложений доказала свою эффективность и целесообразность применения в современных высоко нагруженных облачных инфраструктурах. Также остается открытым вопрос для развития исследования о конкретных рекомендациях по выбору оптимальной стратегии многокритериальной оптимизации в зависимости от типа облака, топологии сети и профиля нагрузки.

Вывод. В основу оптимизационной сетевой модели нахождения максимального потока сети по классическому алгоритму Форда-Фалкерсона заложен также процесс многокритериальной оптимизации и нахождения оптимального пути передачи данных в сложных сетях, таких как облачные среды.

Оптимизационная сетевая модель является эффективным подходом к оптимизации потока данных в облачной среде; позволяет определить оптимальный маршрут передачи данных между узлами сети, учитывая ограничения ресурсов и требования к производительности. Алгоритм демонстрирует высокую эффективность и может быть применён в различных сценариях распределения вычислительных нагрузок в облачных инфраструктурах. Применение модели позволяет компаниям и организациям эффективно управлять облачными ресурсами, оптимизировать потоки данных и обеспечивать высокую производительность приложений, что является ключевым фактором для успешного

функционирования современных цифровых экосистем. Она также помогает снизить затраты на инфраструктуру и повысить эффективность использования ресурсов.

Библиографический список:

1. Разумников С.В. Планирование развития облачной стратегии на основе применения многокритериальной оптимизации и метода STEM // Доклады томского государственного университета систем управления и радиоэлектроники. – 2020. – Т. 23, № 1. – С. 53–61.
2. Разумников С.В. Некомпенсаторное агрегирование и рейтингование провайдеров облачных услуг // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2018. – Т. 21, № 4. – С. 63–69.
3. Разумников С.В. Разработка программного обеспечения для построения агрегированных рейтингов на основе метода порогового агрегирования // Вестник воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2021. № 2. с. 138-152.
4. Разумников С.В. Модели, алгоритмы и программное обеспечение поддержки принятия стратегических решений к переходу на облачные технологии: монография// Изд-во Томского политехнического университета. - 2020. - 176 с.
5. Маслова Н.А., Мовчан О.В. Использование интеллектуальных агентов при решении задач распределения ресурсов //Искусственный интеллект. – 2014.
6. Мохаммад А.Т.Г. и др. Усовершенствованный алгоритм управления доступом к облачным телекоммуникационным ресурсам //Системы обработки информации. – 2015. – № 1. – С. 150-153.
7. Аллакин В. В., Будко Н. П., Васильев Н. В. Общий подход к построению перспективных систем мониторинга распределенных информационно-телекоммуникационных сетей //Системы управления, связи и безопасности. – 2021. – № 4. – С. 125-227.
8. М.В. Козлов, Ю.Е. Малашенко, И.А. Назарова, Н.М. Новикова Управление топливно-энергетической системой при крупномасштабных повреждениях. I. сетевая модель и программная реализация // Известия РАН. Теория и системы управления, 2017, № 6, с. 50–73.
9. Гвоздинский А.Н., Кудряшов В.Ю. Анализ методов управления распределения ресурсов в сетевых системах // АСУ и приборы автоматики. 2009. № 147. URL: <https://cyberleninka.ru/article/n/analiz-metodov-upravleniya-raspredeleniya-resursov-v-setevyih-sistemah>.
10. Ю. Е. Малашенко, И. А. Назарова, Анализ распределения предельных нагрузок в многопользовательской сети”, Информ. и её примен., 15:4 (2021), 20–26
11. Соловьева А.С. Анализ методов многокритериальной оптимизации сети на основе графовых моделей// Современные технологии в науке и образовании - СТНО-2021:Сборник трудов IV Международного научно-технического форума:в 10т.,Рязань, 3–5 марта 2021 г.. Т. 4. Рязань:Рязанский государственный радиотехнический университет имени В.Ф.Уткина, 2021. – С. 24-27. – EDN JNVCGI.
12. Зеленев А. А., Тарасов А. А. Алгоритм Форда-Фалкерсона //Инновации. 2013 -№ 48. С. 1864-1868.
13. Стьопкин А.В., Пластун Д. А. Алгоритм Форда-Фалкерсона. – 2016.
14. Попов А.Ю. О реализации алгоритма Форда-Фалкерсона в вычислительной системе с многими потоками команд и одним потоком данных/ /Машиностроение и компьютерные технологии. – 2014. – № 9. – С. 162-180.
15. Исакова А.К., Шанляева А.С.Использование алгоритма Форда-Фалкерсона для нахождения максимального потока в сети//Вестник Казахской академии транспорта и коммуникаций им. М.Тынышпаева. – 2013. – 4. – С. 75-80.
16. Маслова Н.А., Мовчан О. В. Использование интеллектуальных агентов при решении задач распределения ресурсов // Искусственный интеллект. – 2014.
17. Синдеева Л.В. и др. Применение методов регрессионного анализа в биомедицинских исследованиях //Вестник новых медицинских технологий. – 2013. – Т. 20. – № 2. – С. 216-219.
18. Лагифуллин А.Т. Анализ временных рядов рынка криптовалют интеллектуальными нейросетевыми системами с обратной связью //Аллея науки. – 2018. – Т. 1. – № 5. – С. 1028-1034.
19. Силич В.А., Силич М.П. Теория систем и системный анализ:Уч.пособ. Томск:изд. ТПУ, 2010. – 97 с.
20. Постников В. М., Спиридонов С. Б. Методы выбора весовых коэффициентов локальных критериев // Машиностроение и компьютерные технологии. – 2015. – № 6. – С. 267-287.
21. Макарова И. Л. Анализ методов определения весовых коэффициентов в интегральном показателе общественного здоровья //Символ науки. – 2015. – № 7-1. – С. 87-95.
22. Спиридонов С.Б., Булатова И.Г., Постников В.М. Анализ подходов к выбору весовых коэффициентов критериев методом парного сравнения критериев //Вестник евразийской науки. – 2017. – Т. 9. – № 6 (43). – С. 13.
23. Аксенов К.И., Переверзев А.Н., Трегубов Р.Б. Анализ эффективности алгоритмов поиска максимального потока //Information Systems and Technologies. – 2023. – Т. 5.
24. Никитин С.В.Анализ временных задержек при изменении топологии программно-конфигурируемой сети на базе эмулятора компьютерных сетей Mininet//Техника средств связи. 2023. № 3(163).С.67-73.

References:

1. Razumnikov S.V. Cloud Strategy Development Planning Based on Multicriteria Optimization and the STEM Method. *Reports of Tomsk State University of Control Systems and Radioelectronics*. 2020; 23(1):. 53-61.

2. Razumnikov S.V. Non-compensatory Aggregation and Rating of Cloud Service Providers. *Reports of Tomsk State University of Control Systems and Radioelectronics*. 2018; 21(4): 63-69.
3. Razumnikov S.V. Software Development for Constructing Aggregated Ratings Based on the Threshold Aggregation Method. *Bulletin of Voronezh State University. Series: Systems Analysis and Information Technology*. 2021; 2:138-152.
4. Razumnikov S.V. Models, algorithms, and software for supporting strategic decision-making regarding the transition to cloud technologies: monograph. *Publishing house of Tomsk Polytechnic University*. 2020; 176.
5. Maslova N.A., Movchan O.V. Using intelligent agents in solving resource allocation problems. *Artificial Intelligence*. 2014.
6. Mohamad A.T.G. et al. Improved algorithm for access control to cloud telecommunication resources. *Information Processing Systems*. 2015; 1: 150-153.
7. Allakin V.V., Budko N.P., Vasiliev N.V. General approach to the construction of promising systems for monitoring distributed information and telecommunication networks. *Control, Communication and Security Systems*. 2021; 4:125-227.
8. M.V. Kozlov, Yu.E. Malashenko, I.A. Nazarova, N.M. Novikova Control of the fuel and energy system under large-scale damage. I. Network model and software implementation. *Bulletin of the Russian Academy of Sciences. Control Theory and Systems*, 2017; 6:50-73.
9. Gvozdzinsky A.N., Kudryashov V.Yu. Analysis of resource distribution management methods in network systems. *ACS and automatic devices*. 2009;147.<https://cyberleninka.ru/article/n/analiz-metodov-upravleniya-raspredele-niya-resursov-v-setevykh-sistemah>.
10. Yu.E. Malashenko, I.A. Nazarova, Analysis of the distribution of maximum loads in a multi-user network”, *Inform. i eoe primen.*, 15:4 (2021), 20–26.
11. S. Solovieva, “Analysis of methods for multi-criteria network optimization based on graph models”, in: “Modern technologies in science and education - STNO-2021: Proceedings of the IV International Scientific and Technical Forum: in 10 volumes”, Ryazan, March 3–5, 2021;4. – Ryazan: Ryazan State Radio Engineering University named after V.F. Utkin, 2021;24-27. – EDN JNVCGL.
12. Zelenov A.A., Tarasov A.A. Ford-Fulkerson Algorithm. *Innovations*. 2013;48: 1864-1868.
13. S'opkin A.V., Plastun D.A. Ford-Fulkerson Algorithm. 2016.
14. Popov A.Yu. On the implementation of the Ford-Fulkerson algorithm in a computing system with many command streams and one data stream. *Mechanical Engineering and Computer Technologies*. 2014;9:162-180.
15. Iskakova A.K., Shanlayakova A.S. Using the Ford-Fulkerson Algorithm to Find the Maximum Flow in the Network. *Bulletin of the Kazakh Academy of Transport and Communications named after M. Tynyshpayev*. 2013;4: 75-80.
16. Maslova N.A., Movchan O.V. Using Intelligent Agents in Solving Resource Allocation Problems. *Artificial Intelligence*. 2014.
17. Sindeeva L.V. et al. Application of Regression Analysis Methods in Biomedical Research. *Bulletin of New Medical Technologies*. 2013;20(2): 216-219.
18. Latifullin A.T. Analysis of Cryptocurrency Market Time Series by Intelligent Neural Network Systems with Feedback. *Alley of Science*. 2018;1(5):1028-1034.
19. Silich V.A., Silich M.P. Systems Theory and Systems Analysis:Textbook.Tomsk:Publishing House, 2010;97.
20. Postnikov V.M., Spiridonov S.B. Methods for Selecting Weighting Coefficients of Local Criteria. *Mechanical Engineering and Computer Technologies*. 2015; 6:267-287.
21. Makarova I.L. Analysis of Methods for Determining Weighting Coefficients in an Integral Indicator of Public Health. *Symbol of Science*. 2015;7-1:87-95.
22. Spiridonov S.B., Bulatova I.G., Postnikov V.M. Analysis of approaches to the selection of weighting coefficients of criteria by the method of paired comparison of criteria. *Bulletin of Eurasian Science*. 2017;9:6 (43):13.
23. Aksenov K.I., Pereverzev A.N., Tregubov R.B. Analysis of the efficiency of algorithms for searching for the maximum flow. *Information Systems and Technologies*. 2023;5.
24. Nikitin S.V. Analysis of time delays when changing the topology of a software-defined network based on the Mininet computer network emulator. *Communication Engineering*. 2023;3 (163):67-73.

Information about the author:

Разумников Сергей Викторович, кандидат технических наук, доцент, отделение цифровых технологий и безопасности; razumnikov@tpu.ru; demolove7@inbox.ru. ORCID 0000-0002-1417-498X

Information about the author:

Sergei V. Razumnikov, Cand. Sci. (Eng.), Assoc. Prof., Department of Digital Technologies and Security; razumnikov@tpu.ru; ORCID 0000-0002-1417-498X

Конфликт интересов/Conflict of interest.

Автор заявляет об отсутствии конфликта интересов/The author declare no conflict of interest.

Поступила в редакцию/Received 18.06.2025.

Одобрена после рецензирования/Revised 30.08.2025.

Принята в печать/Accepted for publication 14.01.2026.