

УДК 681.324:519.613

## ОБЛАЧНОЕ УПРАВЛЕНИЕ ТРАНСПОРТОМ: ВЫБОР ОПТИМАЛЬНОГО МАРШРУТА

**А.Н. Зиарманд, С.В. Чумаченко, Е.И. Литвинова, В.И. Хаханов**

*Харьковский национальный университет радиоэлектроники*

Рассматривается алгоритм построения оптимального маршрута на дорожной инфраструктуре между двумя координатами, которые отождествляются с пунктом отправления и прибытия. Возможные ограничения связаны с дорожными коллизиями, авариями и ремонтом дорог, что также должно быть учтено при выборе оптимального маршрута. Предполагается, что реализация алгоритма будет представлена в качестве online облачного сервиса для водителей транспортных средств в рамках создания киберфизической системы интеллектуального облачного управления транспортом. Предлагается также критерий качества топологии, учитывающий суммы всех расстояний между каждой парой компонентов или вершин соответствующего графа, приведенной к числу ребер. Ставятся и решаются следующие задачи: 1) определить все кратчайшие пути и соответствующие расстояния из фиксированной вершины  $v$  до остальных вершин инфраструктуры; 2) найти кратчайшие расстояния между всеми парами вершин для графа с единичными весовыми коэффициентами, заданного матрицей смежности; 3) найти кратчайшие расстояния между всеми парами вершин и оценить инфраструктуры по разработанным критериям качества для различных типов графов, заданных матрицами смежности; 4) выполнить сравнение различных архитектур и графовых структур в целях определения эффективных решений для сокращения времени движения по заказанному маршруту. В результате проведенных исследований выполнен пошаговый синтез предложенного метода построения оптимального маршрута движения транспортного средства. Упрощенные критерии качества топологий показывают, что дорожная инфраструктура, построенная на треугольных соединениях пунктов отправления и прибытия, в два раза эффективнее, чем общепринятая топология “Манхеттен”.

**Ключевые слова:** киберфизическая система, облачное управление, трафик, метод Дейкстры, топология, маршрут, Smart Cloud Traffic Control.

**Введение.** Трудно предсказать и перечислить все позитивные социальные, технологические и технические последствия радикального преобразования существующего мира после внедрения online цифровых сервисов интеллектуального облака дорожного движения. К 2020 году следует ожидать массового появления беспилотного транспорта при исполнении маршрутов. Для ближайшего будущего ниже представлены отдельные и очевидные доказательные инновационные научно-технические решения социальных, гуманитарных, экономических и экологических проблем, связанные с появлением облачного мониторинга и

управления дорожным движением.

Иновация online облачного сервиса Smart Cloud Traffic Control характеризуется введением в цифровую карту планеты параметра времени, переносом светофоров и всех знаков в виртуальное киберпространство. Создание виртуальной интеллектуальной инфраструктуры дорожного движения позволяет:

- повысить качество жизни водителей;
- уменьшить время нахождения в пути;
- снизить затраты на топливо;
- сэкономить тысячи тонн металла для изготовления светофоров и дорожных знаков, миллионы киловатт-часов электроэнергии на поддержание их работоспособности, а также миллионы долларов на эксплуатационные расходы;
- уменьшить время установки и актуализации светофоров и дорожных знаков в виртуальной инфраструктуре городов до нескольких минут.

Все это в совокупности дает возможность автоматизировать процессы квазиоптимального управления транспортом и дорожным движением в режиме реального времени, что является *актуальной задачей* для решения социальных, гуманитарных, экономических и экологических проблем.

**Анализ литературных данных и постановка проблемы.** Киберфизическая система online облачного управления транспортом ориентирована на предоставление сервисов для 9 миллионов водителей Украины и 7 тысяч транспортных компаний. Аналогов таких систем в мире пока не существует. Однако имеются отдельные компоненты для создания инфраструктуры: электронные карты; спутниковые системы локации и навигации; базы данных в облаках; средства мониторинга, сбора и защиты информации; автомобильные компьютеры; водительские гаджеты; управляемые светофоры; сотовая связь; финансовая доступность аренды для водителей облачных мобильных сервисов навигации, управления и мониторинга движения транспортных средств - в пределах 100 долл. США в год; наличие программных, аппаратных и сетевых систем централизованного управления дорожным движением в масштабах страны; доступность облачных вычислительных технологий, постоянное совершенствование и удешевление для создания инфраструктуры дорог в киберпространстве; возрастание компьютерной, мобильной и интернет-грамотности населения; понимание со стороны государства, граждан и полиции о необходимости создания и использования интеллектуальной инфраструктуры и облачного сервиса для качественного и безопасного дорожного движения.

Основания для исследования:

1) Теоретические разработки интеллектуальных моделей, методов и программно-аппаратных средств анализа киберпространства, связанные с дискретной оптимизацией, поиском, распознаванием и принятием решений [1–4].

2) Опыт разработки и применения встроенных и RFID цифровых систем для мониторинга дорожного движения [5–9].

3) Опыт разработки и внедрения программных продуктов и облачных сервисов для оптимизации маршрутов транспортных средств украинских предприятий в целях минимизации материальных и временных затрат и повышения качества обслуживания пассажиров [10–15].

4) Разработки распределенной системы управления дорожным движением в условиях крупных городов и мегаполисов на основе высоконадежной вычислительной техники [11–13], прототипирование умных киберфизических систем управления трафиком [16, 17].

5) Методы оценивания вычислительных структур и поиска кратчайших путей между парой вершин [19–23], известные модификации алгоритма Дейкстры [18–24].

6) Существующие системы мониторинга дорожного движения в США, Канаде и Японии – OnStar и NEXCO Central. Система OnStar ориентирована на мониторинг отдельных машин. Ее сервисы доступны владельцам автомобилей: Acura, Audi, Isuzu, Subaru, Volkswagen. Система NEXCO Central осуществляет глобальный мониторинг трафика на главных магистралях. На данный момент насчитывается около 4 000 000 пользователей данного сервиса. Стоимость одного устройства мониторинга - порядка 200 *долл. США*. Используется CDMA канал связи, предоставляемый преимущественно Verizon Wireless в США и Bell Mobility в Канаде. Для определения местоположения используется GPS. Имеется возможность голосовой связи с операторами. Информация с сенсоров (в основном, это датчики ударов и срабатывания подушек безопасности) автоматически передается в call-центры. Это позволяет немедленно оповещать спасательные и правоохранительные органы о местоположении аварии. Кроме этого, все машины, оборудованные данной системой, имеют GPS передатчик, который позволяет отследить угнанный автомобиль. Также имеется возможность получения информации о скорости, расходе топлива, направлении движения и стиле вождения автомобиля. Данная информация используется страховыми компаниями для расчета стоимости индивидуальных страховых полисов. Новые модели автомобилей оборудуются системой удаленной остановки двигателя. Автомобиль можно завести только после ввода специального секретного кода. Стоимость сервисов – Safe & Sound – 18.95 *долл. США* в месяц. Тариф включает автоматическое оповещение об аварии, мониторинг угнанного автомобиля, аварийные сервисы. Directions & Connections – 28.90 *долл. США* в месяц. Дополнительно имеется возможность мониторинга направления движения и стиля вождения автомобилиста. Система NEXCO Central разработана компанией Japan Highway Public Corporation. Принцип работы заключается в глобальном мониторинге дорожного движения на главных автострадах страны.

Система покрывает порядка 2000 км дорог. Дата-центр обрабатывает данные, получаемые от датчиков с минутным интервалом с помощью глобальной IP сети. На дорогах установлены 744 точки снятия и передачи информации по телефонным каналам о дорожной ситуации.

7) Глобальные инфраструктурные проекты несколько веков подряд объединяют страны. Трансконтинентальные железные дороги и автомагистрали, подводные тоннели и гигантские мосты, космические и авиационные проекты служат сильнейшим катализатором для развития экономик многих государств. Развитие сети Интернет, мобильных и облачных технологий сделало возможным безграничное масштабирование вычислительных мощностей и объемов хранимых данных. Следующий слой киберпространства – Internet of Things, что предполагает общение вещей между собой без вмешательства человека, автоматизированный сбор, обработку и анализ big data, генерируемых умными сенсорами. Возможно построение высокоточных детерминированных моделей окружающей среды и умных городов, что позволит значительно повысить уровень жизни, комфорта и безопасности человека за счет постоянного мониторинга показаний интеллектуальных датчиков в реальном времени.

8) Развитие и внедрение концепции Internet of Things требует решения сложных проблем в области компьютерной инженерии (<http://eai.eu/>, <http://iot.ieee.org/>), которыми занимаются мировые научно-исследовательские сообщества (IEEE, EAI) и компании (Intel, NASA, IBM, Apple, Google, Samsung, Dell, AT&T, CISCO, THALES, WorldSensing, Aguila technologies, Connit, SigFox, Guglielmo, DQuid, BitCarrier). Идея Internet of things основана на использовании беспроводных гетерогенных сетей, работающих в различных частотных диапазонах по стандартам – ZigBEE, WiFi, LTE (3G, 4G, 5G), Bluetooth, которые обеспечивают дальность действия, скорость передачи данных и низкое энергопотребление. Основная функциональность инфраструктуры Internet of things – сбор и анализ данных - использует структуры: big data, open data, cloud computing. Первые ориентированы на анализ больших объемов неструктурированных данных. Вторые – это концепция форматирования информации для использования компьютерами без вмешательства человека. Здесь данные открыты и не лимитированы авторским правом путем свободного лицензирования. Облачные вычисления являются безгранично масштабируемой платформой для обработки и хранения big data, поступающих с умных сенсоров в реальном времени. Неотъемлемой частью Internet of Things является межмашинное взаимодействие M2M – набор технологий и подходов, обеспечивающих обмен информацией между механизмами для создания умных городов (<http://www.android.com/auto/>, <https://www.apple.com/ios/carplay/>, <http://java.dzone.com/articles/car-wars-connectedcars>). Здесь фигурируют компании, предоставляющие платформы и готовые аппаратные решения: Connit, Aguila

Technologies, WorldSensing. Проект WorldSensing FastPrk – это сервис умной парковки, позволяющий водителю с помощью мобильного устройства быстро находить свободное место, а городским властям эффективнее использовать парковочные пространства. Сенсор парковки работает в частотном диапазоне Sub-GHz с радиусом 500 м.

9) Сегодня автомобиль немислим без Internet-сервисов, ориентированных на совмещение приятной поездки, комфортного отдыха и реализации бизнес-операций. Машина исполняет роль места с полным набором функциональных сервисов делового человека: голосовые – связь, сообщения, почта, банкинг, навигация, музыка, игры, видео. Подключение автомобиля к Интернет путем синхронизации мобильного телефона с монитором (компьютером) транспортного средства дает возможность аутентифицировать водителя в реальном мире и виртуальном киберпространстве. Здесь лидируют компании Apple и Google. Они анонсировали платформы связи, навигации и развлечений, встроенные в автомобиль (Apple iOS CarPlay, Google Android Auto), которые используют микропроцессорную операционную систему Blackberry QNX. Для инсталляции системы необходимо подключить смартфон Android или iOS кабелем USB к компьютеру автомобиля. Появление на рынке автомобильных ОС позволяет сделать вывод, что через 3...4 года все машины будут подключены к глобальным сервисам сети Интернет, направленным на повышение качества жизни человека в процессе дорожного движения. Такая договоренность есть между 31 ведущей автомобильной компанией мира. Практически все, что было задекларировано в проекте iCTC [1–3] три года тому назад, постепенно реализуется в сервисах Apple CarPlay, кроме одного – виртуального уличного светофора на лобовом стекле – мониторе автомобиля. Светофоров нет на море и в воздухе, не будет их и на земле через 5...10 лет.

10) Обновленная структура всемирного общества инженеров IEEE включает 38 сообществ, 9 комитетов и 7 советов, в том числе интересные для рассматриваемого проекта: IEEE Geoscience and Remote Sensing Society, IEEE Intelligent Transportation Systems Society, IEEE Professional Communication Society, IEEE Systems, Man, and Cybernetics Society, IEEE Vehicular Technology Society, IEEE Cloud Computing Community, IEEE Electric Vehicles Community, IEEE Biometrics Council, IEEE Sensors Council, SAE. Это свидетельствует о тенденциях, связанных с Smart Cloud Traffic Control, что подтверждает также его рыночную и научно-техническую привлекательность.

**Цель и задачи исследования.** Цель исследования – синтез оптимального маршрута движения транспортного средства путем использования модифицированного алгоритма (метода) Дейкстры на топологической инфраструктуре дорожного движения, приводимой к графу, с учетом существования дорожных коллизий.

Для достижения отмеченной цели были поставлены следующие задачи:

- определить все кратчайшие пути и соответствующие расстояния из фиксированной вершины  $v$  до остальных вершин инфраструктуры;
- найти кратчайшие расстояния между всеми парами вершин для графа с единичными весовыми коэффициентами, заданного матрицей смежности;
- найти кратчайшие расстояния между всеми парами вершин и оценить инфраструктуру по разработанным критериям качества для различных типов графов, заданных матрицами смежности;
- выполнить сравнение различных архитектур и графовых структур в целях определения эффективных решений для сокращения времени движения по заказанному маршруту.

**Упрощенный критерий качества топологии.** Средняя длина пути между компонентами топологии есть основной параметр, влияющий на быстродействие выполнения маршрутов между узлами инфраструктуры. При рассмотрении вариантов реализации топологии необходимо определить интегральную характеристику в виде суммы всех расстояний между каждой парой компонентов или вершин соответствующего графа, приведенной к числу ребер или дуг. Здесь интерес представляют примитивные фигуры: четырехугольник, треугольник и тетраэдр. Последний обладает уникальным свойством – каждая вершина тетраэдра имеет три соседних, в то время как треугольник обладает уникальными двумя смежными вершинами на плоскости. Для оценки стоимости информационной транзакции между двумя компонентами системы в графе из  $n$  вершин вводится критерий качества структуры. Он определяется отношением суммы длин минимальных путей  $\min P_i$  между всеми парами вершин  $\frac{1}{2}(n^2 - n)$  к количеству ребер топологии  $k$ :

$$L = \frac{\sum_{i=1}^{\frac{1}{2}(n^2 - n)} \min P_i}{\frac{1}{2}(n^2 - n)} \times \frac{1}{k} = \frac{\sum_{i=1}^{\frac{1}{2}(n^2 - n)} \min P_i}{k} = \frac{1}{k} \sum_{i=1}^{\frac{1}{2}(n^2 - n)} \min P_i \quad (1)$$

Здесь аргумент  $k$  – количество ребер в графе, является затратной частью или стоимостной характеристикой топологической структуры, поэтому она фигурирует в знаменателе. Но творческое увеличение данного параметра может существенно уменьшить числитель дроби, в котором записана интегральная длина минимальных путей между всеми парами вершин графа, стремящаяся к минимуму. Таким образом, чем меньше интегральная стоимость транзакции  $L$ , тем выше эффективность топологической структуры.

*Пример 1.* Определить критерии качества трех структурных реализаций, имеющих 4 вершины-компонента (a, b, c, d):  $G_1 = \{ab, bc, ad, cd\}$ ,  $G_2 = \{ab, ac, ad,$

cd, bc, bd},  $G_3 = \{ab, bc, ac, ad, cd\}$ :

$$\begin{array}{c}
 \begin{array}{c|cccc}
 & a & b & c & d \\
 \hline
 a & . & 1 & 1 & . \\
 G_1 = b & 1 & . & . & 1 \\
 c & 1 & . & . & 1 \\
 d & . & 1 & 1 & .
 \end{array}, G_2 = \begin{array}{c|cccc}
 & a & b & c & d \\
 \hline
 a & . & 1 & 1 & 1 \\
 b & 1 & . & 1 & 1 \\
 c & 1 & 1 & . & 1 \\
 d & 1 & 1 & 1 & .
 \end{array}, G_3 = \begin{array}{c|cccc}
 & a & b & c & d \\
 \hline
 a & . & 1 & 1 & . \\
 b & 1 & . & 1 & 1 \\
 c & 1 & 1 & . & 1 \\
 d & . & 1 & 1 & .
 \end{array}
 \end{array} \quad (2)$$

Используя приведенную формулу, достаточно просто подсчитать отношения суммы длин минимальных путей для каждой из 6 пар: {ab, ac, ad, cd, bc, bd}, соединенных  $k=(4,6,5)$  ребрами топологии:

$$L_1 = \frac{1+2+1+1+2+1}{4} = \frac{8}{4} = 2,0; \quad (3)$$

$$L_2 = \frac{1+1+1+1+1+1}{6} = \frac{6}{6} = 1,0; \quad (4)$$

$$L_3 = \frac{1+1+1+1+2+1}{5} = \frac{7}{5} = 1,4. \quad (5)$$

Важной характеристикой L-критерия является его чувствительность, когда незначительная модификация графа – добавление одного ребра - может привести к существенному изменению оценки в сторону уменьшения. Рыночная привлекательность анализа эффективности структур актуальна для транспортной инфраструктуры городов при проектировании и перестройке коммуникаций.

*Пример 2.* Показано применение критерия качества к оцениванию трех графовых структур, имеющих 6 вершин и различные топологии соединения, представленные в виде матриц:

$$\begin{array}{c}
 \begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
 \hline
 a & . & 1 & 1 & . & . & . \\
 G_1 = b & 1 & . & 1 & 1 & . & 1 \\
 c & 1 & 1 & . & . & . & 1 \\
 d & . & 1 & . & . & 1 & 1 \\
 e & . & . & . & 1 & . & 1 \\
 f & . & 1 & 1 & 1 & 1 & .
 \end{array}, G_2 = \begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
 \hline
 a & . & 1 & . & 1 & . & . \\
 b & 1 & . & 1 & . & 1 & . \\
 c & . & 1 & . & . & . & 1 \\
 d & 1 & . & . & . & 1 & . \\
 e & . & 1 & . & 1 & . & 1 \\
 f & . & . & 1 & . & 1 & .
 \end{array}, G_3 = \begin{array}{c|cccccc}
 & a & b & c & d & e & f \\
 \hline
 a & . & 1 & . & 1 & 1 & . \\
 b & 1 & . & 1 & 1 & 1 & 1 \\
 c & . & 1 & . & . & 1 & 1 \\
 d & 1 & 1 & . & . & 1 & . \\
 e & 1 & 1 & 1 & 1 & . & 1 \\
 f & . & 1 & 1 & . & 1 & .
 \end{array}
 \end{array} \quad (6)$$

Здесь три графа имеют 9, 7 и 11 дуг соответственно. Подсчет критерия в соответствии с L-формулой дает следующие результаты:

$$L(G_1) = \frac{9 \times 1 + 6 \times 2}{9} = \frac{21}{9} = 2,33; \quad (7)$$

$$L(G_2) = \frac{7 \times 1 + 6 \times 2 + 2 \times 3}{7} = \frac{25}{7} = 3,57; \quad (8)$$

$$L(G_3) = \frac{11 \times 1 + 4 \times 2}{11} = \frac{19}{11} = 1,73. \quad (9)$$

***Получены следующие результаты:***

1) Предложенный критерий качества топологии достаточно просто реализовать в облачном сервисе, доступном для широкого круга пользователей, желающих определить эффективность старой или инновационной транспортной инфраструктуры.

2) Критерий показывает очевидные преимущества треугольной топологии взаимодействия системных компонентов для обслуживания маршрутов в киберфизическом пространстве.

3) Треугольная метрика пространства есть самая экономичная, поскольку создает самые короткие расстояния и пути между координатами благодаря наличию транзитивного замыкания. Треугольная топология городов выигрывает у метрики Манхэттена по сокращению маршрутов движения от 33 до 66%.

4) Внедрение данной топологии в структурную организацию новых городов обеспечит сокращение длины соединительных путей и уменьшение времени исполнения маршрута не менее, чем на 33%.

5) Данная топология обеспечивает повышение надежности перевозок до 66% благодаря треугольной организации связей структурных компонентов, когда любая коллизия на компоненте инфраструктуры имеет не менее двух соседей.

6) Объемная организация системных компонентов по треугольной тетраэдрной топологии предоставляет еще большие преимущества по оптимизации маршрутов движения.

7) Треугольная инфраструктура обеспечивает повышение надежности инфраструктуры благодаря возможности восстановления до 66% путей и компонентов без потери транспортных потоков. Иначе, 66% неисправных связей в треугольной топологии транспортной системы оставляет ее в работоспособном состоянии.

8) Недостатком треугольной топологии можно считать необычность для человека треугольной системы координат, привыкшего к декартовой метрике. Здесь речь идет об идентификации точки на плоскости тремя координатами. Декартова метрика предполагает использование двух координат для позиционирования объекта.

***Модификация алгоритма Дейкстры для избранных топологий.*** Известно, что алгоритм Дейкстры определяет кратчайшие цепи из выбранной вершины графа (пункта отправления) до всех остальных его вершин (пунктов прибытия). Усовершенствование алгоритма Дейкстры для отыскания цепей наименьшей длины связано с сокращением количества операций (сложений и сравнений) и сохранением полученной информации на одном из этапов для последующих вы-



числений. Это достигается процедурой расстановки меток алгоритма Дейкстры, которая уменьшает сложность алгоритма до  $O(n^2)$ . Известно, что при определении кратчайших путей между всеми парами вершин используется алгоритм Флойда-Уоршелла, сложность которого составляет  $O(n^3)$ .

*Задача 1.* Определить все кратчайшие пути и расстояния из вершины  $v_1$  (на рис. 1 – кратко 01) до всех остальных вершин треугольной топологии графа  $G_1$ , содержащего 21 вершину.

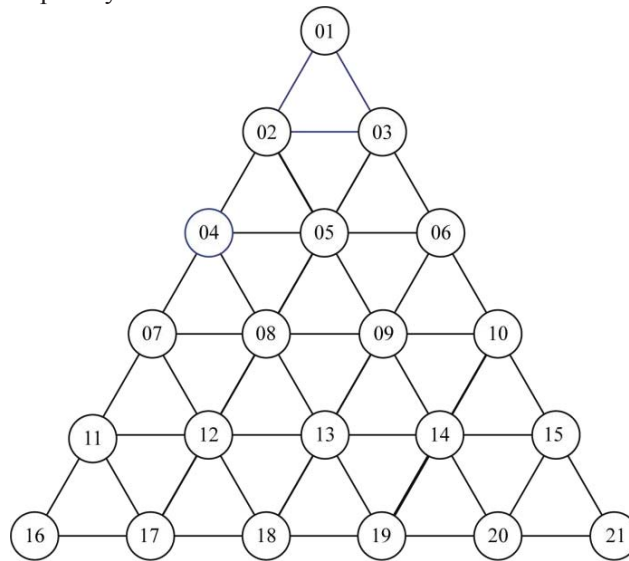


Рис. 1. Граф  $G_1$  – конфигурация сети

*Решение.* Рассматриваемый на рис. 1 граф  $G_1$  является неориентированным. Применение алгоритма Дейкстры позволяет найти решение поставленной задачи. В процессе реализации алгоритма Дейкстры заполняется таблица, количество строк и столбцов которой определяется мощностью множества вершин графа,  $21 \times 21$ . В заголовках строк таблицы указываются вершины, до которых предстоит найти кратчайшее расстояние (табл. 1).

Комментарии к заполнению табл. 1:

1) В заголовке первого столбца в качестве пункта отправления указывается начальная вершина  $u=v_1$ , пройденное расстояние принимается равным нулю:  $r=0$ . Каждое ребро идентифицируется столбцовой (начало) и строчной (конец) вершинами и читается “столбец-строка”. При заполнении первого столбца учитываются следующие правила: начальная вершина  $v_1$  не изменяется и повторяется в каждой ячейке; расстояние совпадает с длиной ребра ( $=1$ ), соединяющего начальную (столбцовую) вершину с вершиной из строки. При отсутствии такого ребра расстоянию присваивается временная метка  $\infty$ .



После заполнения первого столбца из всех конечных числовых меток выбирается минимальная – 1 (если таких несколько, то выбирается любая из них). Такой метке соответствует постоянно помеченная вершина в строке –  $v_2$ . Это означает, что кратчайшее расстояние до вершины  $v_2$  уже найдено и равно 1. Постоянно помеченная вершина принимается в качестве нового отправного пункта  $u=v_2$  и вместе с минимальным расстоянием  $r=1$  заносится в заголовок очередного столбца.

2) При заполнении второго столбца таблицы числовые метки модифицируются согласно правилу: к длине ребра  $(v_2, j)$  добавляется пройденное расстояние 1, результат сравнивается с предыдущей числовой меткой в строке. Если суммарное расстояние меньше предыдущего, то оно принимается в качестве нового, при этом вершина изменяется на вершину из текущего столбца. Так, ребро  $(v_2, v_4)$  имеет длину 1, тогда пройденное расстояние  $r=1+1=2$  имеет конечное значение, следовательно, оно принимается в качестве текущего вместо  $\infty$ , при этом вершина  $v_1$  изменяется на вершину  $v_2$  из текущего столбца.

Ребро  $(v_2, v_3)$  имеет длину 1, тогда пройденное расстояние  $r=1+1=2 > 1$ . Поэтому в строке  $v_3$  сохраняется предыдущая числовая метка, обозначение вершины при этом также не изменяется. *Алгоритм Дейкстры в этом случае претерпевает такую модификацию: если конечная числовая метка определена, далее она не изменяется (поскольку может только увеличиваться). Это значит, минимальное расстояние (длина цепи), оказывается, определено найденным значением числовой метки. Таким образом, выполнять сложение и сравнение для вершин с конечными числовыми метками на последующих этапах не имеет смысла. Следует рассматривать вершины поочередно (в порядке возрастания индексов). На каждой следующей итерации в качестве постоянно помеченной будет выбрана очередная такая вершина, пока все они не будут исчерпаны. Далее рассматриваются вершины с конечными числовыми метками со значениями на 1 больше.*

Из всех конечных числовых меток второго столбца выбирается минимальная – 1, которой соответствует вершина  $v_3$ . Теперь вершина  $u=v_3$  является постоянно помеченной, она выбирается в качестве нового отправного пункта и записывается в заголовке третьего столбца одновременно с пройденным расстоянием  $r=1$ .

3) Третий столбец заполняется аналогично предыдущему. Здесь модифицируется метка в строке  $v_6$ . Все числовые метки оказываются одинаковыми, поэтому выбирается первая по порядку, а именно -  $v_4$ . Теперь  $u=v_4$  является постоянно помеченной, рассматривается в качестве нового отправного пункта и записывается в заголовок четвертого столбца вместе с пройденным расстоянием  $r=2$ .

4) В четвертом столбце не изменяются постоянные метки из строк  $v_5$  и  $v_6$ , равные 2. При этом модифицируются метки в строках  $v_7$  и  $v_8$ . Поэтому выбирается вершина  $u=v_5$ , которая, теперь постоянно помеченная, принимается как новый отправной пункт и сохраняется в заголовке следующего столбца наряду с пройденным расстоянием  $г=2$ .

5) В пятом столбце числовая метка 2 в строке  $v_6$  не изменяется и является минимальной. При этом появляется конечная метка 3 в строке  $v_9$ . Таким образом, в качестве постоянно помеченной вершины теперь выбирается  $v_6$ , которая заносится в заголовок шестого столбца вместе с пройденным расстоянием 2.

6) В шестом столбце появляется новая конечная числовая метка 3 в строке  $v_{10}$ , остальные метки не изменяются. В качестве постоянно помеченной вершины теперь выбирается  $v_7$ , которая заносится в заголовок седьмого столбца вместе с пройденным расстоянием 3.

7) В седьмом столбце появляются две новые конечные числовые метки 4 в строках  $v_{11}$  и  $v_{12}$ , остальные метки не изменяются. В качестве постоянно помеченной следует выбрать вершину  $v_8$ , которая вместе с пройденным расстоянием 3 заносится в заголовок восьмого столбца.

8) В восьмом столбце появляется одна новая конечная числовая метка 4 в строке  $v_{13}$ , остальные метки не изменяются. Теперь в качестве постоянно помеченной следует выбрать вершину  $v_9$ , которая вместе с пройденным расстоянием 3 заносится в заголовок девятого столбца.

9) В девятом столбце добавляется еще одна новая конечная числовая метка 4 в строке  $v_{14}$ , остальные метки не изменяются, поэтому в качестве постоянно помеченной выбирается вершина  $v_{10}$ , которая вместе с пройденным расстоянием 3 заносится в заголовок десятого столбца.

10) В десятом столбце добавляется еще одна новая конечная числовая метка 4 в строке  $v_{15}$ , остальные метки не модифицируются, поэтому в качестве постоянно помеченной выбирается вершина  $v_{11}$ , которая вместе с пройденным расстоянием 4 заносится в заголовок одиннадцатого столбца.

11) В одиннадцатом столбце появляются сразу две новые конечные числовые метки 5 в строках  $v_{16}$  и  $v_{17}$ , остальные метки не модифицируются. В качестве постоянно помеченной выбирается вершина  $v_{12}$ , которая вместе с пройденным расстоянием 4 заносится в заголовок двенадцатого столбца.

12) В двенадцатом столбце появляется еще одна числовая метка 5 в строке  $v_{18}$ , остальные метки не изменяются. В качестве постоянно помеченной выбирается вершина  $v_{13}$ , которая вместе с пройденным расстоянием 4 заносится в заголовок тринадцатого столбца.

13) В тринадцатом столбце появляется еще одна числовая метка 5 в строке  $v_{19}$ , остальные метки не изменяются. В качестве постоянно помеченной выби-

рается вершина  $v_{14}$ , которая вместе с пройденным расстоянием 4 заносится в заголовок четырнадцатого столбца.

14) В четырнадцатом столбце появляется еще одна числовая метка 5 в строке  $v_{20}$ , остальные метки не изменяются. В качестве постоянно помеченной выбирается вершина  $v_{15}$ , которая вместе с пройденным расстоянием 4 заносится в заголовок пятнадцатого столбца.

15) В пятнадцатом столбце появляется конечная числовая метка 5 в строке  $v_{21}$ , остальные метки не изменяются. Все строки имеют конечные числовые метки. Следовательно, процесс поиска минимальных расстояний завершен.

Таким образом, из табл. 1 можно получить информацию обо всех кратчайших цепях и соответствующих им длинах.

Например, требуется найти кратчайшую цепь из вершины  $v_1$  в вершину  $v_{21}$ . Последовательность вершин в цепи выписывается с конца: последняя заполненная ячейка в строке  $v_{21}$  содержит информацию о длине кратчайшей цепи  $r_{\min}=5$  и предпоследней вершине в этой цепи –  $v_{15}$ . Информация о предыдущей вершине находится в последней ячейке строки  $v_{15}$ , в данном случае – вершина  $v_{10}$ , далее –  $v_6$ ,  $v_3$  и  $v_1$ , которая является началом маршрута:

$$v_1 \xrightarrow{1} v_3 \xrightarrow{1} v_6 \xrightarrow{1} v_{10} \xrightarrow{1} v_{15} \xrightarrow{1} v_{21}, \text{ dist} = 5.$$

Данные всех кратчайших цепей представлены в табл. 2.

Граф, иллюстрирующий дерево кратчайших цепей из вершины  $v_1$ , представлен на рис. 2.

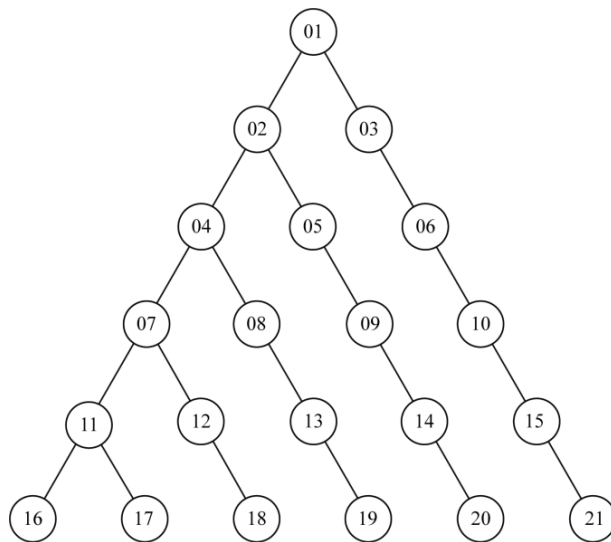


Рис. 2. Дерево кратчайших цепей из вершины  $v_1$

Таблица 2

Кратчайшие цепи из вершины $v1$ графа $G_1$	
Цепь	Длина
$\overset{1}{v1} \rightarrow v2, r_{\min} = 1$	$r(v1, v2) = 1$
$\overset{1}{v1} \rightarrow v3, r_{\min} = 1$	$r(v1, v3) = 1$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow v4, r_{\min} = 2$	$r(v1, v4) = 2$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow v5, r_{\min} = 2$	$r(v1, v5) = 2$
$\overset{1}{v1} \rightarrow \overset{1}{v3} \rightarrow v6, r_{\min} = 2$	$r(v1, v6) = 2$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow v7, r_{\min} = 3$	$r(v1, v7) = 3$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow v8, r_{\min} = 3$	$r(v1, v8) = 3$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v5} \rightarrow v9, r_{\min} = 3$	$r(v1, v9) = 3$
$\overset{1}{v1} \rightarrow \overset{1}{v3} \rightarrow \overset{1}{v6} \rightarrow v10, r_{\min} = 3$	$r(v1, v10) = 3$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v7} \rightarrow v11, r_{\min} = 4$	$r(v1, v11) = 4$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v7} \rightarrow v12, r_{\min} = 4$	$r(v1, v12) = 4$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v8} \rightarrow v13, r_{\min} = 4$	$r(v1, v13) = 4$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v5} \rightarrow \overset{1}{v9} \rightarrow v14, r_{\min} = 4$	$r(v1, v14) = 4$
$\overset{1}{v1} \rightarrow \overset{1}{v3} \rightarrow \overset{1}{v6} \rightarrow \overset{1}{v10} \rightarrow v15, r_{\min} = 4$	$r(v1, v15) = 4$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v7} \rightarrow \overset{1}{v11} \rightarrow v16, r_{\min} = 5$	$r(v1, v16) = 5$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v7} \rightarrow \overset{1}{v11} \rightarrow v17, r_{\min} = 5$	$r(v1, v17) = 5$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v7} \rightarrow \overset{1}{v12} \rightarrow v18, r_{\min} = 5$	$r(v1, v18) = 5$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v4} \rightarrow \overset{1}{v8} \rightarrow \overset{1}{v13} \rightarrow v19, r_{\min} = 5$	$r(v1, v19) = 5$
$\overset{1}{v1} \rightarrow \overset{1}{v2} \rightarrow \overset{1}{v5} \rightarrow \overset{1}{v9} \rightarrow \overset{1}{v14} \rightarrow v20, r_{\min} = 5$	$r(v1, v20) = 5$
$\overset{1}{v1} \rightarrow \overset{1}{v3} \rightarrow \overset{1}{v6} \rightarrow \overset{1}{v10} \rightarrow \overset{1}{v15} \rightarrow v21, r_{\min} = 5$	$r(v1, v21) = 5$

Поскольку все ребра в графе (рис. 2) имеют вес 1, из табл. 1 видно, что расстояния при вычислении могут увеличиваться только на 1. Поэтому фактически,

как только бесконечная метка изменилась на конечную числовую метку, она впоследствии уже не модифицируется, т.е. не может уменьшиться. Это значит, что соответствующее кратчайшее расстояние между вершинами уже определено. Таким образом, количество сложений и сравнений в алгоритме Дейкстры можно сократить. Это достигается за счет исключения сравнений с полученными ранее конечными числовыми метками. При этом постоянно помеченные вершины с минимальными метками последовательно выбираются из списка до их исчерпания. Затем выполняется переход к меткам, большим на единицу. Можно утверждать, что время реализации алгоритма сокращается на 30%.

*Задача 2.* Определить все кратчайшие пути и расстояния из вершины  $v_1$  (на рис. 3 – кратко 01) до всех остальных вершин прямоугольной топологии графа  $G_2$ , содержащего 21 вершину.

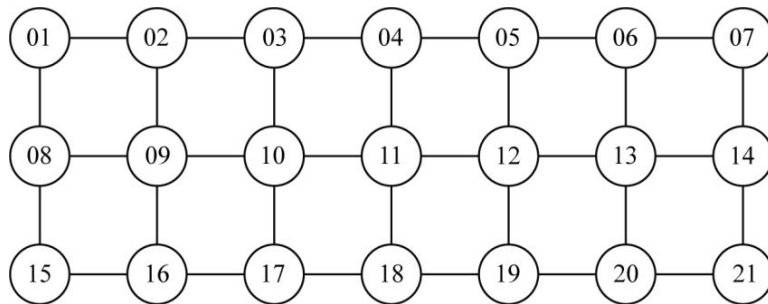


Рис. 3. Граф  $G_2$  – конфигурация сети

*Решение.* Рассматриваемый на рис. 3 граф  $G_2$  является неориентированным. Применение алгоритма Дейкстры позволяет найти решение поставленной задачи.

В процессе реализации алгоритма Дейкстры заполняется таблица, количество строк и столбцов которой определяется мощностью множества вершин графа, т.е.  $21 \times 21$ . В заголовках строк таблицы указываются вершины, до которых предстоит найти кратчайшее расстояние (табл. 3).

Следует заметить, что алгоритм Дейкстры здесь претерпевает такую же модификацию, что и в задаче 1.

Таким образом, из табл. 3 можно получить информацию обо всех кратчайших цепях и соответствующих им длинах. Например, требуется найти кратчайшую цепь из вершины  $v_1$  в вершину  $v_{21}$ . Последовательность вершин в цепи выписывается с конца: последняя заполненная ячейка в строке  $v_{21}$  содержит информацию о длине кратчайшей цепи  $r_{\min}=8$  и предпоследней вершине в этой цепи –  $v_{14}$ .

Таблица 3

Вычисление кратчайших расстояний из вершины  $v_1$  графа  $G_2$

$G_2$	$U=V_1$ $r=0$	$U=V_2$ $r=1$	$U=V_8$ $r=1$	$U=V_3$ $r=2$	$U=V_9$ $r=2$	$U=V_{15}$ $r=2$	$U=V_4$ $r=3$	$U=V_{10}$ $r=3$	$U=V_{16}$ $r=3$	$U=V_5$ $r=4$	$U=V_{11}$ $r=4$	$U=V_{17}$ $r=4$	$U=V_6$ $r=5$	$U=V_{12}$ $r=5$	$U=V_{18}$ $r=5$	$U=V_7$ $r=6$	$U=V_{13}$ $r=6$	$U=V_{19}$ $r=6$	$U=V_{14}$ $r=7$	$U=V_{20}$ $r=7$
$v_2$	$v_{1,1}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_3$	$v_{1,00}$	$v_{2,2}$	$v_{2,2}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_4$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{3,3}$	$v_{3,3}$	$v_{3,3}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_5$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{4,4}$	$v_{4,4}$	$v_{4,4}$	-	-	-	-	-	-	-	-	-	-	-
$v_6$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{5,5}$	$v_{5,5}$	$v_{5,5}$	-	-	-	-	-	-	-	-
$v_7$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{6,6}$	$v_{6,6}$	$v_{6,6}$	$v_{6,6}$	-	-	-	-	-
$v_8$	$v_{1,1}$	$v_{1,1}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_9$	$v_{1,00}$	$v_{2,2}$	$v_{2,2}$	$v_{2,2}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_{10}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{3,3}$	$v_{3,3}$	$v_{3,3}$	$v_{3,3}$	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_{11}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{4,4}$	$v_{4,4}$	$v_{4,4}$	$v_{4,4}$	-	-	-	-	-	-	-	-	-	-
$v_{12}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{5,5}$	$v_{5,5}$	$v_{5,5}$	$v_{5,5}$	-	-	-	-	-	-	-
$v_{13}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{6,6}$	$v_{6,6}$	$v_{6,6}$	$v_{6,6}$	-	-	-	-
$v_{14}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{7,7}$	$v_{7,7}$	$v_{7,7}$	-	-
$v_{15}$	$v_{1,00}$	$v_{6,2}$	$v_{6,2}$	$v_{6,2}$	$v_{6,2}$	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_{16}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{9,3}$	$v_{9,3}$	$v_{9,3}$	$v_{9,3}$	-	-	-	-	-	-	-	-	-	-	-	-	-
$v_{17}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	$v_{10,4}$	-	-	-	-	-	-	-	-	-
$v_{18}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	$v_{11,5}$	-	-	-	-	-	-
$v_{19}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{12,6}$	$v_{12,6}$	$v_{12,6}$	$v_{12,6}$	-	-	-
$v_{20}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{13,7}$	$v_{13,7}$	$v_{13,7}$	$v_{13,7}$	-
$v_{21}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{1,00}$	$v_{14,8}$	$v_{14,8}$



Информация о предыдущей вершине находится в последней ячейке строки  $v_{14}$ , в данном случае – вершина  $v_7$ , далее –  $v_6, v_5, v_4, v_3, v_2$  и  $v_1$ , которая является началом маршрута:

$$v_1 \xrightarrow{1} v_2 \xrightarrow{1} v_3 \xrightarrow{1} v_4 \xrightarrow{1} v_5 \xrightarrow{1} v_6 \xrightarrow{1} v_7 \xrightarrow{1} v_{14} \xrightarrow{1} v_{21}, r_{\min} = 8.$$

Данные обо всех кратчайших цепях представлены в табл. 4.

Граф, иллюстрирующий дерево кратчайших цепей из вершины  $v_1$ , представлен на рис. 4.

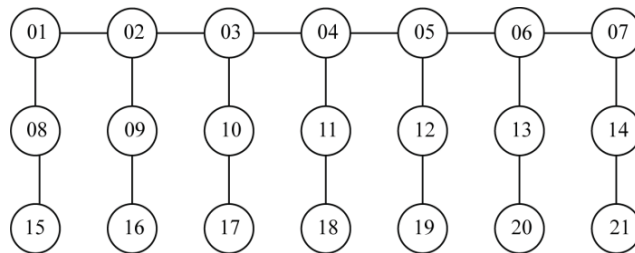


Рис. 4. Дерево кратчайших цепей из вершины  $v_1$  графа  $G_2$

*Описание модифицированного алгоритма Дейкстры.* Каждой вершине из множества  $V$  ставится в соответствие метка, которая определяет минимальное известное расстояние от этой вершины до начальной вершины  $a$  (пункта отправления). Алгоритм выполняется пошагово. На каждом шаге он “посещает” одну вершину и пытается уменьшить числовые метки. Реализация алгоритма завершается, когда все вершины посещены.

*Инициализация.* Числовая метка самой вершины  $a$  полагается равной 0, остальным вершинам присваивается временная метка – бесконечность. Это означает, что расстояния от  $a$  до других вершин пока неизвестны. На данном этапе все вершины графа помечаются как непосещенные.

*Шаг алгоритма.* Если все вершины посещены, алгоритм завершает работу. В противном случае, из еще не посещенных вершин выбирается вершина  $u$  с минимальной меткой. При этом рассматриваются все возможные маршруты, где  $u$  является предпоследним пунктом. Вершины, в которые ведут ребра из  $u$ , называются соседними по отношению к  $u$ . Для каждого соседа вершины  $u$ , кроме отмеченных как посещенные, рассматривается новая длина пути, равная сумме значений текущей метки вершины  $u$  и длины ребра, соединяющего  $u$  с этим соседом. В традиционном алгоритме Дейкстры далее учитывается такой шаг: если полученное значение длины меньше значения метки соседа, то метка соседа заменяется полученным значением длины.

Таблица 4

Кратчайшие цепи из вершины  $v_1$  графа  $G_2$ 

Цепь	Длина
$v_1 \rightarrow v_2, r_{\min} = 1$	$r(v_1, v_2) = 1$
$v_1 \rightarrow v_2 \rightarrow v_3, r_{\min} = 2$	$r(v_1, v_3) = 2$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4, r_{\min} = 3$	$r(v_1, v_4) = 3$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5, r_{\min} = 4$	$r(v_1, v_5) = 4$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6, r_{\min} = 5$	$r(v_1, v_6) = 5$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7, r_{\min} = 6$	$r(v_1, v_7) = 6$
$v_1 \rightarrow v_8, r_{\min} = 1$	$r(v_1, v_8) = 1$
$v_1 \rightarrow v_2 \rightarrow v_9, r_{\min} = 2$	$r(v_1, v_9) = 2$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_{10}, r_{\min} = 3$	$r(v_1, v_{10}) = 3$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_{11}, r_{\min} = 4$	$r(v_1, v_{11}) = 4$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_{12}, r_{\min} = 5$	$r(v_1, v_{12}) = 5$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_{13}, r_{\min} = 6$	$r(v_1, v_{13}) = 6$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_{14}, r_{\min} = 7$	$r(v_1, v_{14}) = 7$
$v_1 \rightarrow v_8 \rightarrow v_{15}, r_{\min} = 2$	$r(v_1, v_{15}) = 2$
$v_1 \rightarrow v_2 \rightarrow v_9 \rightarrow v_{16}, r_{\min} = 3$	$r(v_1, v_{16}) = 3$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_{10} \rightarrow v_{17}, r_{\min} = 4$	$r(v_1, v_{17}) = 4$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_{11} \rightarrow v_{18}, r_{\min} = 5$	$r(v_1, v_{18}) = 5$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_{12} \rightarrow v_{19}, r_{\min} = 6$	$r(v_1, v_{19}) = 6$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_{13} \rightarrow v_{20}, r_{\min} = 7$	$r(v_1, v_{20}) = 7$
$v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_7 \rightarrow v_{14} \rightarrow v_{21}, r_{\min} = 8$	$r(v_1, v_{21}) = 8$

Для графов с ребрами единичной длины (веса) сумма расстояний каждый раз может увеличиваться только на 1. Поэтому в упомянутом пункте только бесконечные метки соседа могут изменяться на конечные числовые метки, которые впоследствии остаются постоянными, т.е. уменьшаться уже не могут. Соответ-

ствующие расстояния являются числами натурального ряда. По этой причине сравнение целесообразно проводить только в целях определения конечных числовых меток для тех вершин, которые таковых пока не имеют, т.е. их временные метки равны бесконечности. Если отсутствует ребро между постоянно помеченной вершиной и вершиной с бесконечной меткой, то выбирается постоянно помеченная вершина с минимальной меткой в текущем столбце, как и ранее. Это позволяет реализовать попытку найти минимальный маршрут через другую вершину. При этом сложение и сравнение меток с уже имеющимися в столбце конечными метками не проводятся, что сокращает время поиска.

Если конечная числовая метка уже определена на текущем этапе, далее она уже не изменяется (поскольку может только увеличиваться), иначе минимальное расстояние (длина цепи) уже оказывается определено этим значением числовой метки. Таким образом, выполнять сложение и сравнение для вершин с конечными числовыми метками на последующих этапах не имеет смысла. Такие вершины следует рассматривать поочередно (в порядке возрастания индексов) и выбирать на каждой следующей итерации в качестве постоянно помеченных, пока все они не будут исчерпаны. Далее рассматриваются вершины с конечными числовыми метками со значениями на 1 больше.

После рассмотрения всех соседей вершина  $u$  помечается как посещенная, и шаг алгоритма повторяется.

LABEL – массив для хранения текущих меток вершин. PERM – массив для указания постоянно помеченных вершин (вершины становятся постоянно помеченными, когда они оказываются равными  $u_i$  для какого-либо  $i$ ). Если  $PERM(v)=1$ , то  $v$  – постоянно помеченная вершина, и ее метка равна  $d(s,v)$ . Вначале  $PERM(s)=1$  и  $PERM(v)=0$  при  $v \neq s$ . PRED – массив указателей на вершины, из которых осуществлен переход в вершины с постоянной меткой. Если вершина  $v$  помечена постоянной меткой, то последовательность  $v, PRED(v), PRED(PRED(v)), \dots, s$  – вершины, составляющие кратчайший ориентированный путь из  $s$  в  $v$ .

1. Начало. Положить  $LABEL(s)=0, PERM(s)=1, PRED(s)=s; \forall v \neq s$  положить  $LABEL(v)=\infty, PERM(v)=0, PRED(v)=v$ .

2. Пусть  $i=0, u=s$  ( $u$  – последняя из вершин с неизменной меткой. Теперь это вершина  $s$ ).

3. Вычисление LABEL и изменение элементов массива PRED. Положить  $i=i+1$ . Выполнить для каждой вершины  $v$  с бесконечной меткой следующие дей-

ствия:

3.1. Положить  $M = \min\{\text{LABEL}(v), \text{LABEL}(u) + w(u,v)\}$ , где  $w(u,v) = 1$  – длина ребра, соединяющего вершины  $u$  и  $v$ , если такое существует, иначе (т.е. когда не существует ребра  $(u,v)$ ) – в качестве постоянно помеченной выбирается вершина с минимальной конечной числовой меткой, а если таких несколько, то выбирается одна из них;

3.2. Если  $M < \text{LABEL}(v)$ , то положить  $\text{LABEL}(v) = M$ ,  $\text{PRED}(v) = u$ .

4. Выделение вершины  $u_i$ . Среди всех вершин, которые не помечены неизменной меткой, найти вершину  $w$  с наименьшей меткой (если таких вершин несколько, то выбор можно сделать произвольно). Положить  $\text{PERM}(w) = 1$ ,  $u_i = w$  (является последней вершиной с неизменной меткой).

5. Если  $i < n - 1$ , то возврат к шагу 3, иначе – конец (все кратчайшие пути найдены).

Метки вершин представляют собой длины кратчайших путей;  $v$ ,  $\text{PRED}(v)$ ,  $\text{PRED}(\text{PRED}(v))$ , ...,  $s$  есть вершины кратчайшего ориентированного  $s$ - $t$  пути.

*Замечание:* в традиционном алгоритме Дейкстры пункт 3 выполнялся для всех вершин  $v$ , кроме вершин с неизменной меткой. В модифицированном алгоритме он применяется только к вершинам с временными метками  $\text{LABEL}(v) = \infty$ , поскольку остальные метки модифицироваться не будут.

Таким образом, приведена модификация алгоритма Дейкстры для неориентированных взвешенных графов с единичной длиной ребра треугольной и прямоугольной топологий.

**Обсуждение полученных результатов.** Предложенный метод построения оптимального маршрута на дорожной инфраструктуре между двумя координатами, который имеет дорожные коллизии, можно реализовать в качестве online облачного сервиса для водителей транспортных средств, что составляет перспективу исследования.

Упрощенные критерии  $L$  качества топологий, примененные к двум типам рассмотренных графовых структур, при исполнении транспортных маршрутов дают следующие результаты. Для прямоугольной топологии:  $L = \text{sum}/k$ ,  $k = 32$ ,  $\text{Sum} = 714$ ;  $L = 714/32 = 22,3125$ ; для треугольной топологии:  $L = \text{sum}/k$ ,  $k = 45$ ,  $\text{Sum} = 545$ ;  $L = 545/45 = 12,1$ . Это означает, что дорожная инфраструктура, построенная на треугольных соединениях пунктов отправления и прибытия, в два раза эффективнее  $L = 12,1$ , чем общепринятая топология Манхеттен, имеющая оценку  $L = 22,3125$ .

### **Выводы**

В результате проведенных исследований выполнен пошаговый синтез предложенного метода построения оптимального маршрута движения транспортного средства путем использования модифицированного алгоритма Дейкстры на топологической инфраструктуре дорожного движения, приводимой к графу, с учетом существования дорожных коллизий. Решены следующие задачи:

1. Определены все кратчайшие пути и соответствующие им расстояния из фиксированной вершины  $v$  до остальных вершин инфраструктуры, представленной треугольной и прямоугольной топологиями.
2. Определены кратчайшие расстояния между всеми парами вершин для графов треугольной и прямоугольной топологий с единичными весовыми коэффициентами, которые компактно можно представить в виде матрицы.
3. Разработаны критерии качества и выполнена оценка инфраструктуры для различных типов графов, заданных матрицами смежности.
4. Выполнено сравнение различных архитектур и графовых структур и показана эффективность решений с точки зрения сокращения времени движения по заказанному маршруту.

### **Литература**

1. **Хаханов, В.И., Меликян В.Ш., Саатчян А.Г., Шахов Д.В.** «Зеленая волна» – облако мониторинга и управления дорожным движением // Вестник НПУА. Серия “Информ. технологии, электроника, радиотехника”. – 2013. – Вып. 16, №1. – С.53–60.
2. Cloud Traffic Control System / **V.I. Hahanov, O.A. Guz, A.N. Ziarmand, et al** // Proc. of IEEE East-West Design and Test Symposium. – 2013. – P. 72–76.
3. Cloud Traffic Monitoring and Control / **V. Hahanov, W. Gharibi, Baghdadi Ammar Awni Abbas, et al** // Proceedings of the 2013 IEEE 7th International conference on intelligent data acquisition and advanced computing systems (IDAACS). – Berlin, 2013. – P. 244–248.
4. **Бондаренко М.Ф., Хаханов В.И., Литвинова Е.И.** Структура логического ассоциативного мультипроцессора // Автоматика и телемеханика. – 2012. – № 10. – С. 71–92.
5. Research on the Integrated Management of Highway Based on Radio Frequency Identification Technology / **Antao Lu, Li Yushan, Sun Yufang, et al** // Third International Conference on Measuring Technology and Mechatronics Automation (ICMTMA). – 2011. – Vol. 3. – P. 116–119.
6. **Pandit A.A., Talreja J., Mundra A.K.** RFID Tracking System for Vehicles (RTSV) [Text] // First International Conference on Computational Intelligence, Communication Systems and Networks. – 2009. – P. 160–165.
7. **Jiang Lin-ying, Wang Shuai, Zhang Heng, Tan Han-qing.** Improved Design of Vehicle

- Management System Based on RFID // International Conference on Intelligent System Design and Engineering Application (ISDEA).- 2010. – Vol. 1. – P. 844–847.
8. **Chen Xue-Mei, Wei Zhong-Hua** Vehicle management system based on multi-node RFID cards // 30th Chinese Control Conference (CCC). – 2011. – P. 5497–5499.
  9. **Дудников С., Боечко И.** Бесконтактная идентификация транспорта, основанная на RFID // Компоненты и технологии.– 2007.– №1.– <http://www.kite.ru/assets/files/pdf/200701140.pdf>
  10. **Manikonda P., Yerrapragada A.K., Annasamudram S.S.** Intelligent traffic management system // IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (Student). – 2011. – P. 119–122.
  11. **Samad T.** Perspectives in Control Engineering Technologies, Applications, and New Directions // Intelligent Transportation Systems: Roadway Applications.- Wiley-IEEE Press, 2001.– P. 348–369.
  12. **Schutte J.** Recent trends in automatic train controls // IEEE Intelligent Transportation Systems.– 2001.– P. 813–819.
  13. **Zingirian N., Valenti C.** Sensor clouds for Intelligent Truck Monitoring // IEEE Intelligent Vehicles Symposium (IV). – 2012. – P. 999–1004.
  14. An Intelligent Autonomous Vehicle Management System / **L.B. Branisso, E.R.R. Kato, E.C. Pedrino, et al** // Second Brazilian Conference on Critical Embedded Systems (CBSEC). – 2012. – P.42–47.
  15. **Brizgalov V.V., Chukhantsev V., Fedorkin E.** Architecture of traffic control systems using cloud computing // International Conference and Seminar on Micro/Nanotechnologies and Electron Devices (EDM). – 2010. – P. 215–216.
  16. Cyber physical system – smart cloud traffic control / **V. Hahanov, W. Gharibi, L.S. Abramova, et al** // Proceedings of IEEE East-West Design & Test Symposium (EWDTS). – 2014. – P. 1–18. – DOI: 10.1109/EWDTS.2014.7027107
  17. Intelligent road control and monitoring / **V. Hahanov, W. Gharibi, K.L. Man, E. Litvinova, et al** // Lecture Notes in Electrical Engineering (LNEE). – 2013. – Vol. 235.– P. 327–335.
  18. **Gorobets A., Chugurov I., Scherbin D., Chumachenko S.** Dijkstra Algorithm for cyber structures analysis [Text] // Experience of Designing and Application of CAD Systems in Microelectronics (CADSM). – 2013. – P. 61-65.
  19. **Dijkstra E.W.** A note on two problems in connexion with graphs // Numerische Mathematik.– 1959.– Vol. 1.– P. 269-271.
  20. **Кормен Т.Х., Лейзерсон Ч.И., Ривест Р.Л., Штайн К.М.** Алгоритмы: построение и анализ = Introduction to Algorithms.– М.: Вильямс, 2006. – 1296 с.
  21. **Левитин А.В.** Глава 9. Жадные методы: Алгоритм Дейкстры // Алгоритмы: введение в разработку и анализ = Introduction to The Design and Analysis of Algorithms. – М.: Вильямс, 2006. – С. 189-195.

22. **Кузнецов Н.А., Фетисов В.Н.** Алгоритм Дейкстры с улучшенной робастностью для управления маршрутизацией в IP-сетях // Автоматика и телемеханика.– 2008.– № 2.– С. 80–85.
23. Модели анализа эффективности вычислительных структур / **В.И. Хаханов, С.В. Чумаченко, Мурад Али Аббас и др.** // Радиозлектроника и информатика. – 2012. – №3. – С. 66-71.
24. **Томас Т.М.** Структура и реализация сетей на основе протокола OSPF: Руководство Cisco = OSPF Network Design Solutions.– М.: Вильямс, 2004.– 816 с.

*Поступила в редакцию 05.11.2017.*

*Принята к опубликованию 21.12.2017.*

## **ՏՐԱՆՍՊՈՐՏԻ ԱՄՊԱՅԻՆ ԿԱՌԱՎԱՐՈՒՄ. ՕՊՏԻՄԱԼ ԵՐԹՈՒՂՈՒ ԸՆՏՐՈՒԹՅՈՒՆԸ**

**Ա.Ն. Զիարմանդ, Ա.Վ. Չումաչենկո, Ե.Ի. Լիտվինովա, Վ.Ի. Հահանով**

Դիտարկվում է ճանապարհային ինֆրակառուցվածքի երկու կոորդինատների միջև օպտիմալ երթուղու կառուցման ալգորիթմը, որոնք նույնականացվում են մեկնման և ժամանման կետերի հետ: Հնարավոր սահմանափակումները կապված են ճանապարհային բախումների, վթարների և ճանապարհային վերանորոգումների հետ, ինչը նույնպես պետք է հաշվի առնվի օպտիմալ երթուղին ընտրելիս: Ենթադրվում է, որ ալգորիթմի իրականացումը պետք է ներկայացվի որպես առցանց ամպային ծառայություն տրանսպորտային միջոցների վարորդների համար՝ տրանսպորտի բանական ամպային կառավարման կիբեռնֆիզիկական համակարգի ստեղծման շրջանակներում: Առաջարկվում է տոպոլոգիայի որակի չափանիշ, որը հաշվի է առնում յուրաքանչյուր զույգ բաղադրիչների կամ էլ համապատասխան գրաֆի կողերի թվին հանգեցված գագաթների միջև բոլոր հեռավորությունների գումարը: Առաջարկվում և լուծվում են հետևյալ խնդիրները. 1) որոշել բոլոր կարճագույն ուղիները և համապատասխան հեռավորությունները  $\nu$  ամրակայված գագաթից մինչև ինֆրակառուցվածքի մնացած գագաթները, 2) որոշել կարճագույն հեռավորությունները հարևանության մատրիցով տրված և միավոր կշռային գործակիցներով գրաֆի բոլոր զույգ գագաթների միջև, 3) որոշել կարճագույն ուղիները բոլոր զույգ գագաթների միջև և գնահատել ինֆրակառուցվածքներն ըստ մշակված որակի չափանիշների՝ հարևանության մատրիցներով տրված զանազան տեսակի գրաֆների համար, 4) իրականացնել տարբեր ճարտարապետությունների և գրաֆային կառույցների համեմատություն՝ նպատակ ունենալով ստանալու արդյունավետ լուծումներ տրված երթուղով շարժման ժամանակի կրճատման համար: Իրականացված հետազոտությունների

արդյունքում կատարված է տրանսպորտային միջոցի շարժման օպտիմալ երթուղու կառուցման առաջարկված եղանակի քայլ առ քայլ սինթեզ: Տոպոլոգիաների պարզեցված որակի չափանիշները ցույց են տալիս, որ մեկնման և ժամանման կետերի միջև եռանկյունաձև միացումներով կառուցված ճանապարհային ինֆրակառուցվածքը կրկնակի արդյունավետ է, քան բոլորի կողմից ընդունված «Մանհեյտեն» տոպոլոգիան:

**Առանցքային բառեր.** կիբեռֆիզիկական համակարգ, ամպային կառավարում, տրաֆիկ, Դեյկատրայի մեթոդ, տոպոլոգիա, երթուղի, Smart Cloud Traffic Control:

## CLOUD CONTROL OF TRANSPORT: SELECTION OF AN OPTIMAL ROUTE

**A.N. Ziarmand, S.V. Chumachenko, E.I. Litvinova, V.I. Hahanov**

The algorithm for designing the optimal route on the road infrastructure between two coordinates is considered which are identified with the departure and arrival points. The possible restrictions are connected with road collisions, accidents and road repairs which should also be considered at selecting the optimal route. It is assumed that the algorithm implementation will be introduced as an on-line cloud service for vehicle drivers in the frames of creating a cyber-physical system of intellectual cloud control of traffic. A criterion for the topology quality, taking into account the sums of all distances between each pair of components or the nodes of the corresponding graph reduced to the number of sides is proposed. The following tasks are set and solved: 1) determining all the shortest ways and the corresponding distances from a fixed node  $v$  to the other nodes of the infrastructure; 2) finding the shortest distances between all the pairs of the nodes for a graph with single weight coefficients of the set by the matrix contiguity; 3) finding the shortest distances between all the node pairs and estimate the infrastructures in accordance with the developed quality criteria for different types of graphs of the set by the matrices contiguity; 4) comparing different architectures and graph structures for the purpose of determining efficient solutions to reduce the time of movement along the ordered route. As a result of investigations carried out, a step-by-step synthesis of the proposed method for designing an optimal route of a vehicle movement is carried out. The simplified criteria for the quality topologies show that the road infrastructure designed on triangular connections of the departure and arrival points is twice as efficient as the generally accepted topology “Manhattan”.

**Keywords:** cyberphysical system, cloud control, graph, Dexter method, topology, route, Smart Cloud Traffic Control.