



<http://dx.doi.org/10.35596/1729-7648-2020-18-5-71-79>

*Оригинальная статья*  
*Original paper*

УДК 681.2

## НАБЛЮДАТЕЛЬ ИЗМЕНЕНИЙ В ЛЕСАХ КРАТЧАЙШИХ ПУТЕЙ НА ДИНАМИЧЕСКИХ ГРАФАХ ТРАНСПОРТНЫХ СЕТЕЙ

ХАДЖИНОВА Н.В., РЕВОТЮК М.П., ШИЛИН Л.Ю.

*Белорусский государственный университет информатики и радиоэлектроники*  
*(г. Минск, Республика Беларусь)*

*Поступила в редакцию 29 июня 2020*

© Белорусский государственный университет информатики и радиоэлектроники, 2020

**Аннотация.** Цель работы – разработка базовых структур данных и эффективных по быстродействию и памяти алгоритмов слежения за изменениями предопределенных решений о множествах кратчайших путей на транспортных сетях, уведомления о которых получают автономные координируемые транспортные агенты с централизованным или коллективным управлением. Характерная особенность транспортных операций – независимость и асинхронность появления возмущений оптимальных решений, а также отсутствие глобального влияния отдельных возмущений на множество всех процессов на сети. Тем самым, очевидно, определяется целесообразность реализации идеи реоптимизации существующих решений в реальном времени по мере поступления информации о возмущениях структуры и параметров транспортной сети, различных ограничений на использование существующих кратчайших путей. В отличие от классических задач поиска кратчайших путей на статических или динамических графах, набор контролируемых наблюдателем ситуаций предлагается дополнить учетом ассоциаций деревьев кратчайших путей с фактически использующими такие пути агентами. Это позволит улучшить реактивность процессов уведомления агентов для своевременного переключения на новый путь. Пространство состояний поиска – динамически формируемый двудольный разреженный граф транспортной сети, представленный списком дуг. Базовый алгоритм поиска кратчайших путей использует схему Дейкстры, но реализует для формирования результата поиска метод бутстрэппинга. Компактность представления наблюдаемого леса кратчайших путей достигается отображением отдельных деревьев такого леса на проекцию вершин дерева в памяти, где позиция каждой вершины соответствует расстоянию от корня дерева. Предложенная версия построения процедуры поиска опирается на существующие в системах управления базами данных механизмы создания разных реляционных представлений физической модели данных. Это избавляет от необходимости решения технологических задач комплексирования гетерогенных моделей динамических транспортных сетей, распределения памяти. В результате спецификация различных правил логистики транспортных операций упрощается, так как такие операции в терминах объектно-ориентированных моделей легко определяются полиморфными классами переходов между узлами транспортной сети.

**Ключевые слова:** транспортная сеть, кратчайший путь, метод реоптимизации, мониторинг изменений.

**Конфликт интересов.** Авторы заявляют об отсутствии конфликта интересов.

**Для цитирования.** Хаджинова Н.В., Ревотюк М.П., Шилин Л.Ю. Наблюдатель изменений в лесах кратчайших путей на динамических графах транспортных сетей. Доклады БГУИР. 2020; 18(5): 71-79.

## OBSERVER OF CHANGES IN THE FOREST OF THE SHORTEST PATHS ON DYNAMIC GRAPHS OF TRANSPORT NETWORKS

NATALIA V. KHAJYNOVA, MIKHAIL P. REVOTJUK, LEONID Y. SHILIN

*Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)*

*Submitted 29 June 2020*

© Belarusian State University of Informatics and Radioelectronics, 2020

**Abstract.** The purpose of the work is the development of basic data structures, speed-efficient and memory-efficient algorithms for tracking changes in predefined decisions about sets of shortest paths on transport networks, notifications about which are received by autonomous coordinated transport agents with centralized or collective control. A characteristic feature of transport operations is the independence and asynchrony of the emergence of perturbations of optimal solutions, as well as the lack of global influence of individual perturbations on the set of all processes on the network. This clearly determines the feasibility of realizing the idea of reoptimizing existing solutions in real time as information is received about disturbances in the structure and parameters of the transport network, various restrictions on the use of existing shortest paths. In contrast to the classical problems of finding shortest paths on static or dynamic graphs, it is proposed to supplement the set of situations controlled by the observer by taking into account the associations of shortest path trees with agents that actually use such paths. This will improve the responsiveness of agent notification processes for timely switching to a new path. The space of search states is a dynamically generated bipartite sparse graph of the transport network, represented by a list of arcs. The basic algorithm for finding the shortest paths uses Dijkstra's scheme, but implements a bootstrapping method to generate the search result. The compactness of the representation of the observed forest of shortest paths is achieved by mapping individual trees of such a forest onto the projection of tree vertices in memory, where the position of each vertex corresponds to the distance from the tree root. The proposed version of the construction of the search procedure is based on the mechanisms existing in database management systems for creating different relational representations of the physical data model. This eliminates the need to solve technological problems of complexing heterogeneous models of dynamic transport networks, memory allocation. As a result, the specification of various rules for the logistics of transport operations is simplified, since such operations in terms of object-oriented models are easily determined by polymorphic classes of transitions between nodes of the transport network.

**Keywords:** transport network, shortest path, reoptimization method, monitoring of changes.

**Conflict of interests.** The authors declare no conflict of interests.

**For citation.** Khajynova N.V., Revotjuk M.P., Shilin L.Y. Observer of changes in the forest of the shortest paths on dynamic graphs of transport networks. Doklady BGUIR. 2020; 18(5): 71-79.

### Введение

Объект исследования – процессы управления системами взаимодействующих агентов, определяемые на динамических графах транспортных сетей в терминах задач поиска кратчайших путей. Подвижные агенты, автономно реализуя отдельные транспортные операции, при планировании своей деятельности нуждаются в актуальной информации об обстановке на транспортной сети. Такая информация может дополнять статические описания сетей в виде карт, дорожных знаков оперативными данными о различных ограничениях, природных или техногенных авариях, пробках. Оперативная информация из разных источников в полном объеме чаще всего может учитываться только централизованно (например, [1–3]), а специально создаваемый наблюдатель должен по мере ее поступления высылать уведомления зарегистрированным агентам о интересующих их изменениях. Интересы агента здесь и далее предполагаются отраженными на понятие кратчайшего пути на графе транспортной сети, узлы которой соответствуют фазам транспортного процесса. Способ отображения должен быть известным наблюдателю, а различные параметры модели

транспортных операций, а также ограничения, включая временные, должны проецироваться вычисляемыми функциями на дуги нагруженного ориентированного графа.

Пусть транспортная сеть задана нагруженным ориентированным графом  $G(V, E)$ , где  $V$  – множество вершин,  $E$  – множество дуг с весовой функцией  $W: E \rightarrow R^+$ . Процесс построения дерева кратчайших путей лучшим для этого алгоритмом Дейкстры [4] реализует рекурсивный однократный просмотр дуг до исчерпания возможности его развития из исходной вершины и характеризуется вычислительной сложностью  $O(m + n \cdot \log_2 n)$ , где  $m = |E|$ ,  $n = |V|$ .

Очевидно, что реализация алгоритма Дейкстры на графах с изменяемой структурой порождает проблему выбора способа представления графа и пространства состояний процесса поиска решения. Нагруженный граф  $G(V, E)$ , представляющий реальную транспортную сеть, обычно имеет незначительную степень связности вершин, поэтому практически задается списком дуг и полностью определяется тернарным отношением  $G(x, y, w)$ , где  $x, y$  – концевые вершины дуги с весом  $w$ . В таких случаях описание графа задано легко модифицируемым списком дуг с расширяемыми множествами  $V$  и  $E$ .

Задачи поиска кратчайших путей для координируемых операций на транспортных сетях объективно локальны. Такие задачи можно представить отношением  $F(a, o, d)$ , где  $a$  – агент, нуждающийся в описании кратчайшего пути из вершины  $o$  в вершину  $d$  графа сети,  $o, d \in V$ . В любой момент времени существует потребность в наличии информации о кратчайших путях из множества  $X = \{x | (? , x, ?) \in F\}$  до множества  $Y = \{y | (? , ?, y) \in F\}$ . На практике для отдельной транспортной системы всегда справедливо  $|X| \ll |V|$  и  $|Y| \ll |V|$ , а информация о кратчайших путях актуальна лишь для множества агентов  $Z = \{z | (z, ?, ?) \in F\}$ .

В установленном состоянии сети такая информация может выбираться из множества леса деревьев кратчайших путей, сохраняемых в таблице отношения  $T^s(s, x, d, p)$ , где  $s$  – корневой узел дерева,  $s \in \overline{X}$ ;  $x$  – произвольный узел дерева,  $x \in V$ ;  $d$  – расстояние от корня  $s$  до узла  $x$ ,  $p$  – номер предшествующего узла кратчайшего пути в узел  $x$ . Здесь  $\overline{X}$  – потенциально интересный для конкретной транспортной системы набор корневых вершин дерева из леса,  $\overline{X} \subseteq V$ . Кроме этого, полезно определить ограничение глубины деревьев кратчайших путей множеством конечных вершин  $\overline{F}$ , когда всегда справедливо  $Y \subseteq \overline{F}$ .

Информация о возмущениях транспортной сети может быть задана списком дуг с изменившимися весами, представленным отношением  $R(x, y, w)$ , где  $x, y$  – концевые вершины дуги с весом  $w$  (значение  $w = \infty$  означает запрос удаления дуги  $x \rightarrow y$ ). При этом не обязательно выполнение условия  $x, y \in V$ , что означает отсутствие запрета на коррекцию любой части графа транспортной сети. Если в этот момент выполняется условие  $(x \in X) \vee (y \in Y)$ , то множество агентов  $J = \{a | (a, o, d) \in T(o, d, ?, ?)\}$  должно получить информацию о новом пути после его реоптимизации.

Таким образом, фазы перехода между устанавливающимися состояниями процесса слежения за оптимальностью построенных на сети путей включают выявление факта  $R \neq \emptyset$ , этап реоптимизации и перестройки  $T^s(s, x, d, p)$  для изменившегося отношения  $G(x, y, w)$ , выявление множества заинтересованных в актуальности информации агентов  $J$  и передачу им описания изменения оптимального пути. Далее, в контекстах обсуждения отдельного дерева в лесу, верхний индекс  $s$ , помечающий корень дерева, будем опускать.

Предмет обсуждения – модели и алгоритмы контроля состояния транспортной сети с уведомлением о результатах решения задачи реоптимизации дерева путей. В отличие от известных постановок задач реоптимизации дерева путей [4–7], здесь будут учитываться интересы потребителей результатов решения отдельных локальных задач в лесу подграфов исходного графа. Для этого, кроме сокращения времени на перестройку отдельного дерева путей, необходимо обеспечить формирование актуальных для планирования и реализации конкретных транспортных операций уведомлений об изменении кратчайших путей.

## Методика решения задачи

*Модель графа и пространства состояния поиска.* Предлагаемая далее схема поиска дерева кратчайших путей основана на аналогии последовательного складывания ветвей дерева от корня до последнего постоянно включаемого (помеченного) в дерево узла методом бутстрэппинга. Такой метод отличается возможностью реализации на чисто реляционных моделях, что обеспечивает открытость их расширения для учета различных ограничений.

Рассмотрим вначале форму задания графа транспортной сети, а затем модель переменных состояния поиска кратчайших путей. Здесь будут использованы понятия отношения и представлений (view) таблиц в терминах реляционных баз данных.

Пространство состояний поиска решения в традиционных реализациях алгоритма Дейкстры является проекцией множества вершин графа всей транспортной сети на  $D$  – массив расстояний вершин от корня дерева,  $P$  – массив номеров предшествующих вершин пути,  $Q$  – приоритетная очередь вершин, где элементы упорядочены по текущему значению расстояния от корня дерева [4, 5]. При этом  $|D|=|V|$ ,  $|P|=|V|$ ,  $|Q|\leq|V|$ .

Метод бутстрэппинга принципиально работает с подграфом исходного графа, что вполне соответствует ограниченности размера искомого дерева. В случае поиска кратчайших путей на множестве вершин некоторой компактной части графа, несложно увидеть возможность модификации алгоритма Дейкстры, когда поиск проводится на расширяемом подграфе с последовательно включаемыми вершинами создаваемого дерева кратчайших путей от заданной корневой вершины.

Алгоритм Дейкстры для каждой вершины  $x$  требует представления множества непосредственно достижимых смежных вершин  $x', x'=\{k \mid w(x,k) > 0\}$ , где  $w(x,k)$  – вес дуги  $x \rightarrow k$ ,  $x, k \in V$ . На основе таблицы отношения  $G(x, y, w)$  для графа  $G$  легко построить его виртуальное представление  $G_x(x, y, w)$  в форме FSF (Forward Star Form). Индексация кортежей такой таблицы по ключу  $x$  формально определяет множество связей  $\{(x, y, w), y \in x'\}$ ,  $x \in V$ . После этого проверка условия  $x \in V$  состоит в оценке успеха поиска кортежа с ключом  $x$  в таблице  $G_x$ .

Обозначим для каждой вершины  $y$  множество входных смежных вершин  $'y$ ,  $'y=\{k \mid w(k,y) > 0\}$ , где  $k, y \in N$ . Виртуальное представление  $G_y(x, y, w)$  в форме BSF (Backward Star Form) инверсии графа образуется после индексации таблицы отношения  $G(x, y, w)$  по ключу  $y$ . Такое представление определяет множество кортежей  $\{(x, y, w), x \in 'y\}$ ,  $y \in V$ . Успех поиска кортежа с ключом  $y$  в таблице  $G_y$  означает истинность условия  $y \in V$ .

Таким образом, имеем  $V=V_x \cup V_y$ , что означает отсутствие необходимости явного отдельного определения множества  $V$ . Операция расширения двудольного графа  $G(V, E)$  реализуется на логическом уровне модели данных без использования понятия адресной функции. Это влечет отсутствие проблем комплексирования существующих моделей транспортной сети из разных источников и разной степени детализации. Кроме этого, доступно описание ограничений на условия перемещения на сети в терминах функций как отношений.

Так как любое дерево – связный граф, то если  $s$  – корневая вершина, а  $x$  – некоторый узел или лист дерева кратчайших путей,  $s, x \in V$ , то кратчайший путь  $s \rightarrow x$  можно восстановить обратным движением из листа  $x$ :

$$p(s, x) = \{x, P(x), P(P(x)), \dots, s\}. \quad (1)$$

По определению элементы множества  $p(s, x)$  в (1) упорядочены по неубыванию значений расстояния от корня  $s$ . Альтернативы расширения дерева кратчайших путей представлены листьями без постоянной пометки, путь от корня до которых необязательно кратчайший.

Пусть  $L_k$  – множество листьев существующего на этапе  $k$  дерева кратчайших путей, а  $L_k^*$  – подмножество листьев без постоянной пометки. Согласно (1), на любом этапе

$$L_k = \bigcup \{p(s, x), x \in L_k\}. \quad (2)$$

Рост дерева кратчайших путей происходит только из ближайшего к корню листа  $x_k$  без пометки,  $x_k = \arg \min_x \{d(x), x \in L_k^*\}$ . Лист  $x_k$  превращается в узел ветвления после операции  $L_{k+1}^* = L_k^* \setminus \{x_k\}$  и до окончания поиска становится пассивным (получает постоянную метку). Ветвление из узла  $x_k$  может привести к включению новых или коррекции в сторону уменьшения расстояния от корня  $s$  дерева существующих в  $L_{k+1}^*$  листьев из множества  $x_k'$ . Так как каждому элементу  $x, x \in L_k$ , соответствует  $d(x)$  – длина пути до корня  $s$ , то элементы множества  $L_k^* = \{x \mid d(x) \geq d(x_k), x \in L_k\}$  представляют приоритетную очередь [4, 5].

Очевидно, элементы  $x \in L_k \setminus L_k^*$  остаются упорядоченными по невозрастанию значений  $d(x)$ . Таким образом, необходимо отображение ассоциаций  $x \rightarrow p(x)$ ,  $x \rightarrow d(x)$  и  $d(x) \rightarrow x$  среди переменных состояния процесса построения дерева. Для этого определим отношение  $T(x, d, p)$ , где  $x$  – номер узла дерева,  $d$  – расстояние от корня до узла  $x$ ,  $p$  – номер предшествующего узла не обязательно кратчайшего пути в узел  $x$ .

Состояние процесса построения дерева на любом его этапе  $k$  представлено тройками  $(x, d(x), p(x)), x \in L_k^*$ . Начальное состояние процесса построения дерева соответствует тройке  $(s, 0, s)$ . Условие завершения процесса –  $(L_k^* = \emptyset)$  или  $(x_k = t)$ , если  $t$  – конечная вершина пути. В любом случае  $k \leq |V|$  после завершения поиска пути  $s \rightarrow t$ .

Важно отметить, что кортежи  $T(x, d, p)$  задают динамически порождаемые графы дерева кратчайших путей (проекция  $T(x, d, p) \rightarrow T_p(x, p)$ ) и дерева расстояний (проекция  $T(x, d, p) \rightarrow T_d(x, d)$ ). Учитывая, по определению  $L_k^*$ , уникальность атрибута  $x$  в исходном отношении  $T(x, d, p)$  и его упомянутых проекциях,  $T(x, d, p) \equiv T_d(x, d) \oplus T(x, p)$ , где символ  $\oplus$  обозначает операцию соединения отношений по атрибуту  $x$ . Отсюда следует, что для отображения результата построения дерева кратчайших путей достаточно использовать рутинный в современных технологиях баз данных механизм виртуальных представлений отношения  $T(x, d, p)$ .

Различные виртуальные представления отношения  $T(x, d, p)$  отличаются выражениями ключей упорядочения кортежей. В качестве таких ключей могут быть  $x$ ,  $d$  и конкатенация атрибутов  $p+x$ . В результате дерево кратчайших путей в табличной форме задается вариантами представлений:

$T_0$  – размещение кортежей в памяти на физическом уровне;

$T_x$  – справочник номеров (идентификаторов) начальных вершин дуг графа;

$T_d$  – таблица значений расстояния до вершин графа от его корневой вершины;

$T_p$  – справочник номеров начальных и конечных вершин дуг графа (инверсия графа дерева кратчайших путей, требующаяся для реализации алгоритмов реоптимизации).

Виртуальное представление отношения может определяться разными способами, конкретизируемыми в среде программирования баз данных. Например, поддержка понятия структурных индексов обеспечивает возможность отображения нескольких законов упорядочения одной таблицы отношения. Кортежи такой таблицы наряду с явно идентифицируемыми полями имеют два часто полезных дополнительных атрибута:

$k^0$  – номер кортежа в таблице на физическом уровне;

$k^o$  – номер кортежа в порядке последовательного перечисления кортежей в соответствии с заданным ключом  $o$  законом упорядочения (обычно явно не задается,

но в процессе перемещения любым способом по кортежам корректность номера текущего кортежа выявляется функциями обнаружения выхода за пределы допустимых номеров строк существующей таблицы).

Навигация по кортежам любого представления реализуется операторами поиска по ключу (либо части составного ключа), переходу на начало или конец таблицы, сканированию кортежей по заданным логическим условиям. В любой момент времени некоторый кортеж таблицы представляет текущую позицию с определенным значением  $k^0$ . Изменение значения любого ключа упорядочения меняет соответствие  $k^0 \leftarrow k^0$  без изменения  $k^0$  и выражения (2).

Поддержка понятия структурных индексов позволяет представить алгоритмы слежения за изменениями состояния сети средствами программирования реляционных баз данных.

*Алгоритм мониторинга изменений дерева путей.* Формирование в лесах подграфов графа транспортной сети дерева кратчайших путей с корнем в вершине  $s$  проводится по ранее рассмотренной схеме Дейкстры в представлении  $T_d$ . Начальное состояние нового поиска –  $T_0(x, d, p) = \{(s, 0, s)\}$ . Далее, в представлении  $T_x$  организуется сканирование определяемой остающимися кортежами приоритетной очереди с целью построения дерева кратчайших путей. Процесс сканирования прерывается после достижения всех целевых вершин, перечисленных во множестве  $Y = \{y \mid (?, ?, y) \in F\}$ . При этом целевые вершины могут встречаться на кратчайшем пути несколько раз и в любой момент времени исключаться из  $Y$ . Контроль необходимости продолжения поиска целевых вершин проводится после выделения при наличии из представления  $T_x$  (постоянной пометки) очередной вершины  $x$ .

После выборки очередной вершины из очереди в таблице отношения  $T_d$  фиксируется позиция кортежа  $(x, d_x, ?)$  и в контексте отношения  $T_x$  проводится попытка расширения дерева в смежные вершины из множества  $\{x'\}$ . Если для некоторых вершин справедливо  $(y \in x') \wedge (y \notin T_x(x, ?, ?))$ , то в  $T_x$  добавляется новый кортеж  $(y, d_x + c_{xy}, x)$ . Для вершин, включенных в  $T_x$  ранее, когда справедливо  $(y \in x') \wedge (y \in T_x(x, ?, ?))$ , выполняется попытка пересмотра кратчайшего пути: если  $d_x + c_{xy} < d_y$ , то выполняется  $(y, ?, ?) \leftarrow (y, d_x + c_{xy})$ .

Изменения дуг сети, определяемые отношением  $R(x, y, w)$ , должны быть учтены в новой версии графа  $G_x(x, y, w)$ . Пересмотр результата поиска кратчайшего дерева предлагается проводить посредством перестройки дерева, определяя начальное состояние по минимальному расстоянию от корня дерева до безусловно изменяемого поддерева с корнем  $x_r = \arg \min_k \{d_k \mid (k \in T_x) \wedge (k, ?, ?) \in R\}$ . Это означает возврат в состояние выборки вершины  $x_r$  из очереди и продолжение процесса ветвления по схеме Дейкстры для изменяемой части исходного дерева. Очевидно, что при равномерном распределении  $x_r$  количество шагов (вычислительная сложность) процесса ветвления сокращается в среднем в два раза.

Процедура перестройки дерева характеризуется надежностью учета любых расширений условий ветвления и ограничений на структуру кратчайших путей. Тем не менее, при малых объемах изменений графа, последняя процедура может быть заменена одной из известных процедур реоптимизации [4, 6, 7] текущего дерева кратчайших путей по представлениям  $T_x$  и  $T_p$  (в случае выполнения условий их корректного применения).

Особый вид запрашиваемых изменений существующего дерева – расширение списка конечных вершины. В этом случае необходимо продолжить процесс бутстрэппинга, восстановив состояние прерывания поиска в момент достижения наиболее удаленной от корня конечной вершины из  $T_x$ . Отсюда следует, что кажущееся полезным удаление из  $T_x$  дуг дерева кратчайших путей к листьям вне списка конечных вершин невозможно.

Итак, регулярная обработка фактов изменения дуг графа сети рассмотренными выше процедурами приводит к уведомлению зарегистрированных агентов об изменении структуры интересных им кратчайших путей. Причина появления изменений определяется вне графа, но ее сигнатура может быть включена в параметры дуг графа. Интерпретация таких сигнатур здесь рассматривается как требующая конкретизации рутинная процедура.

*Встречный поиск методом бутстрэппинга.* Известно, что в случае задач поиска кратчайшего пути между двумя вершинами целесообразно организовать встречный поиск на исходном графе и графе с инвертированными связями вершин. В результате вычислительная сложность поиска сокращается в два раза [4, 8, 9]. Рассмотрим далее способ реализации встречного поиска на реляционных представлениях модели сети и результата поиска.

Пометим элементы обозначения исходного графа символом '+', а инвертированного графа – символом '\*'. Можно заметить, что требующиеся графы могут быть легко заданы виртуальными представлениями списка дуг исходного графа  $G(x, y, w)$ , создаваемыми необходимыми для бутстрэппинга структурами смежности FSF ( $G^+ \equiv G_x$ ) и BSF ( $G^* \equiv G_y$ ).

Пусть на исходном графе  $G^+$  указаны начальная и конечная вершины искомого кратчайшего пути  $s \rightarrow t$ . Так как  $V^+ \cap V^* = \emptyset$ , то встречный поиск можно вести порождением двух независимых волн поиска от корней деревьев на несвязном графе  $G^+ \cup G^*$ . Для этого достаточно одновременно начать процессы ветвления из вершин  $s^+ \in V^+$  и  $t^* \in V^*$ . Последнее соответствует формальному объединению графов фиктивной дугой  $s^+ \rightarrow t^*$ , для которой  $w(s^+, t^*) = \infty$ . Таким образом, расширенный граф  $G^+ \cup G^*$  оказывается проекцией двух виртуальных графов на список дуг  $G(x, y, w)$ . Последнее упрощает организацию поиска на динамических графах.

С целью различения процессов на  $G^+$  и  $G^*$ , результат поиска кратчайшего пути от вершины  $s$  к вершине  $t$  при встречном поиске будет представлен деревьями, где узлы идентифицируются конкатенацией номеров вершин исходного графа и признаками принадлежности к виртуальным графам  $G^+$  или  $G^*$ :  $x^s = x+'+', x \in V^s$ ;  $x^t = x+'*', x \in V^t$ . Здесь множества вершин  $V^s \subseteq V^+$  и  $V^t \subseteq V^*$  встречных деревьев формируются рекурсивным переносом на очередном шаге ветвления буквенными символами признаков принадлежности определенному дереву (направления поиска) от корней из начального состояния:  $s^s = s+'+', t^t = t+'*'$ . Для этого представим переменные состояния процесса построения дерева кортежами расширенного отношения  $T(x, d, p) \rightarrow T(x, d, p, q, z)$ , где  $q$  – признак направления поиска ( $q \in \{+', '*'\}$ );  $z$  – признак постоянной пометки узла дерева ( $z \in \{0, 1\}$ ).

Состояние процесса построения деревьев встречного поиска на любом его этапе  $k$  должно быть представлено пятерками  $(x, d(x), p(x), q(x), ?), x \in L_k^*$ . Начальное состояние процесса построения деревьев соответствует пятеркам  $(s, 0, s, '+', 0)$  и  $(t, 0, t, '*', 0)$ . Условие завершения процесса – ( $L_k^* = \emptyset$ ). При попытке постоянной пометки вершины  $x$  некоторого дерева, когда сопряженная вершина уже является постоянно помеченной [8], процесс приостанавливается<sup>1</sup>. После этого необходимо завершить порождение одного из деревьев до встречи с оставшимися помеченными вершинами сопряженного дерева с целью окончательного выделения вершины сопряжения кратчайших путей на встречно растущих деревьях [4, 9]. Далее, следует вернуться по обратному пути до конечной вершины в исходном графе, используя (1).

Изменения дуг сети, принадлежащих графам  $G^+$  и  $G^*$ , целесообразно учитывать ранее рассмотренной операцией перестройки дерева с заданным корнем, определяя начальное состояние по минимальному расстоянию от корня дерева  $s$  до безусловно изменяемого поддерева с корнем  $x_r$  на графе  $G^+$ . В этом случае количество шагов встречного поиска сокращается пропорционально расстоянию от вершины  $x_r$  до корня  $s$ .

<sup>1</sup> Ревотюк М.П., Хаджинова Н.В. Встречный поиск кратчайших путей на больших динамических графах методом бутстрэппинга. BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: сб. материалов VI Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 20–21 мая 2020 г.). Минск: Бестпринт, 2020;(1):324-331. Revotjuk M.P., Khajynova N.V. Point-to-point searching of the shortest paths on the big dynamic graphs by the bootstrapping method. Proc. of VI Int. Conf. "BIG DATA and Advanced Analytics" (Republic Belarus, Minsk, 2020 May 20-21). Minsk: Bestprint, 2020;(1):324-331. (In Russ.)

## Результаты и их обсуждение

Представления модели поиска решения и результатов оптимизации опираются на современные технологические средства реляционных баз данных, обеспечивая открытость, мобильность и гибкость реализации. Делегирование операций манипулирования пространством состояний поиска и его элементами системам управления базами данных не привело к повышению порядка вычислительной сложности. Время реакции на возмущения графа сети сокращено в среднем в два раза. Объем данных уведомлений об изменении дерева кратчайших путей пропорционален количеству его обновленных дуг.

Список дуг графа транспортной сети может быть задан вариантами представлений таблицы  $G(x, y, w) - G_0$ , структур смежности прямого и инвертированного графов –  $G_x$  и  $G_y$ . Потребность в памяти для представлений графа транспортной сети –  $O(m+n)$ , где  $m=|V|$ ,  $n=|E|$ . Результат поиска кратчайших путей задается вариантами представлений отношения таблицы  $T(x, d, p, q, z) - T_0$ , структур смежности прямого и инвертированного дерева кратчайших путей –  $T_x$  и  $T_p$ , списком элементов приоритетной очереди  $T_d$ . Потребность в памяти для представления результата –  $O(n)$ .

Использование реляционной модели графа позволяет обеспечить независимость ее представлений от схемы идентификации вершин. Нет ограничений на отражение различных эвристик повышения эффективности реализации идей алгоритма Дейкстры, не касающихся понятий адресной функции (например, многоуровневый или направленный поиск).

## Заключение

Представленные результаты позволят в рамках современных технологий баз данных обеспечить наблюдение за изменением результатов оптимизации путей на динамических графах неоднородных транспортных сетей с минимальными задержками на обработку особых событий. Вычислительная сложность поиска оптимального решения снижается пропорционально объему локальных изменений данных задачи. Дополнительная память для хранения индексов таблиц виртуальных представлений линейно зависит от размерности модели сети. При этом сокращается потребность в информационном взаимодействии между потребителями результатов оптимизации путей.

## Список литературы

1. Sun Y., Yu X., Bie R., Song H. Discovering time-dependent shortest path on traffic graph for drivers towards green driving. *Journal of Network and Computer Applications*. 2017;83:204-212.
2. Рубанов И.В., Баркетов М.С., Ковалев М.Я. Моделирование движения объектов без остановок по сети пересекающихся маршрутов. *Информатика*. 2018;15(1):21-33.
3. Рыбак В.А., Рябычина О.П. Система экологического мониторинга атмосферного воздуха. *Доклады БГУИР*. 2020;18(4):36-43.
4. Ferone D., Festa P., Napoletano A., Pastore T. Shortest paths on dynamic graphs: a survey. *Pesquisa Operacional*. 2017;37(3):487-508.
5. Pugliese L.D.P., Guerriero F. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*. 2013;62(3):183-200.
6. Pallottino M.S., Scutelli M.G. A new algorithm for reoptimizing shortest paths when the arc costs change. *Operations Research Letters*. 2003;31(2):149-160.
7. Hooks E., Yang B. Updating paths in time-varying networks given arc weight changes. *Transportation Science*. 2005;39(4):451-464.
8. Nicholson T.A.J. Finding the Shortest Route between Two Points in a Network. *The Computer Journal*. 1966;9(3):275-280.
9. Ревотюк М.П., Кароли М.К., Хаджинова Н.В. Быстрый поиск кратчайших путей на графах с предопределенными решениями. *Доклады БГУИР*. 2014;4(82):73-79.

## References

1. Sun Y., Yu X., Bie R., Song H. Discovering time-dependent shortest path on traffic graph for drivers towards green driving. *Journal of Network and Computer Applications*. 2017;83:204-212.
2. Rubanov I.V., Barketau M.S., Kovalyov M.Y. [Modeling movement of objects without stops in a network of crossing routes]. *Informatika = Informatics*. 2018;15(1):21-33. (In Russ.)
3. Rybak V.A., Ryabichina O.P. [Ecological monitoring system of the atmosphere]. *Doklady BGUIR = Doklady BSUIR*. 2020;18(4):36-43. (In Russ.)
4. Ferone D., Festa P., Napoletano A., Pastore T. Shortest paths on dynamic graphs: a survey. *Pesquisa Operacional*. 2017;37(3):487-508.
5. Pugliese L.D.P., Guerriero F. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*. 2013;62(3):183-200.
6. Pallottino M.S., Scutelli M.G. A new algorithm for reoptimizing shortest paths when the arc costs change. *Operations Research Letters*. 2003;31(2):149-160.
7. Hooks E., Yang B. Updating paths in time-varying networks given arc weight changes. *Transportation Science*. 2005;39(4):451-464.
8. Nicholson T.A.J. Finding the Shortest Route between Two Points in a Network. *The Computer Journal*. 1966;9(3):275-280.
9. Revotjuk M.P., Qaraleh M.K., Khajynova N.V. [Quick search of the shortest paths on the graph with a predetermined decision]. *Doklady BGUIR = Doklady BSUIR*. 2014;4(82):73-79. (In Russ.)

## Вклад авторов

Все авторы в равной степени внесли вклад в написание статьи.

## Authors' contribution

All authors equally contributed to the writing of the article.

### Сведения об авторах

Хаджинова Н.В., старший преподаватель кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники.

Ревотюк М.П., к.т.н., доцент, доцент кафедры информационных технологий автоматизированных систем Белорусского государственного университета информатики и радиоэлектроники.

Шилин Л.Ю., д.т.н., профессор, декан факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники.

### Information about the authors

Khajynova N.V., Senior Lecturer of Information Technologies in Automated Systems Department of Belarusian State University of Informatics and Radioelectronics.

Revotjuk M.P., PhD, Associate Professor, Associate Professor of Information Technologies in Automated Systems Department of Belarusian State University of Informatics and Radioelectronics.

Shilin L.Y., D.Sci., Professor, Dean of Faculty of Information Technologies and Control of Belarusian State University of Informatics and Radioelectronics.

### Адрес для корреспонденции

220013, Республика Беларусь,  
г. Минск, ул. П. Бровки, 6,  
Белорусский государственный университет  
информатики и радиоэлектроники  
тел. +375-17-293-86-58;  
e-mail: rmp@bsuir.by  
Ревотюк Михаил Павлович

### Address for correspondence

220013, Republic of Belarus,  
Minsk, P. Brovka str., 6,  
Belarusian State University  
of Informatics and Radioelectronics  
tel. +375-17-293-86-58;  
e-mail: rmp@bsuir.by  
Revotjuk Mikhail Pavlovich