УДК 519.168

А.Н. Трофимчук, В.А. Васянин, Л.П. Ушакова

ОБЗОР МЕТОДОВ И АЛГОРИТМОВ ПОСТРОЕНИЯ КРАТЧАЙШИХ ПУТЕЙ И ПЕРСПЕКТИВЫ ИХ РАЗВИТИЯ

Ключевые слова: задачи построения кратчайших путей, методы и алгоритмы, вычислительная эффективность.

Введение

Задачи построения кратчайших путей (КП) — одни из фундаментальных задач оптимизации на графах и сетях, используемые зачастую во многих приложениях как подзадачи в различных областях естествознания, а также в обыденной жизни. В большей части работ, посвященных вопросам нахождения КП, авторы рассматривают задачу построения КП по одному критерию — минимум или максимум суммы длин дуг в пути. При этом под длиной дуги понимается некоторый ее параметр, например расстояние по дуге, затраты при перевозке по дуге, пропускная способность дуги и т.п. Несмотря на многочисленность работ, связанных с проблемой нахождения КП, внимание к разработке эффективных по быстродействию алгоритмов построения КП не ослабевает. Это, в первую очередь, объясняется тем, что в подавляющем большинстве случаев время решения общей оптимизационной задачи в значительной степени определяется временем построения КП.

За последние десятилетия разработано достаточно много эффективных алгоритмов построения КП по одному критерию. Множество известных алгоритмов построения КП условно можно разбить на три группы: сетевые комбинаторные алгоритмы, основанные на процедурах расстановки и изменения меток в процессе построения КП и использующие эффективные структуры абстрактных типов данных (АТД); алгебраичные или матричные алгоритмы, оперирующие матричными представлениями данных о структуре и характеристиках графа или сети; алгоритмы, базирующиеся на методах решения задач линейного программирования — различных модификациях симплекс-метода.

Для удобства изложения примем единую терминологию — будем говорить о неориентированных сетях с узлами и дугами, подразумевая, что сказанное в большей или меньшей мере относится как к неориентированным, так и ориентированным графам.

Первая и третья группы алгоритмов, в основном, используются для нахождения деревьев КП между заданными подмножествами узлов сети (от одного ко всем и, наоборот, между двумя заданными узлами и т.п.), вторая — для построения КП между всеми узлами сети. В зарубежной литературе часто употребляются термины Single Source (or Sink) Shortest Path (SSSP) — алгоритмы для нахождения деревьев кратчайших путей от заданных источников; All Pairs Shortest Paths (APSP) — алгоритмы для нахождения КП между всеми парами узлов. Для краткости, когда это удобно, будем использовать эту терминологию.

Первые систематизированные библиографические обзоры по алгоритмам построения КП опубликованы в 1969 г. Дрейфусом (Dreyfus S.), в 1975 г. — Пирсом (Pierce A.) за период с 1956 по 1974 гг., в 1984 г. — Део (Deo N.)

и Пангом (Pang C.) и в 1988 г. — Галло (Gallo G.) и Паллоттино (Pallottino S.). Подробное описание основных алгоритмов построения КП и ссылки на их авторов можно найти в книгах [1–4]. Последующие обзоры и таксономия разработки алгоритмов КП приведены в работах [5–8]. Полезно также ознакомиться с презентацией А. Гольдберга (Goldberg A.) «Basic Shortest Path Algorithms» на сайте http://forskning.diku.dk/PATH05/GoldbergSlides.pdf.

Сетевые комбинаторные алгоритмы

Рассмотрим более подробно развитие комбинаторных методов построения КП. Некоторые ссылки в обзоре взяты из [9]. Основная идея всех комбинаторных методов заключается в создании корневого узла с очередью связанных с ним других узлов с метками расстояний до них, которые на каждой итерации алгоритма обновляются в очереди при наращивании кратчайшего дерева. Процедура наращивания дерева заканчивается, когда очередь узлов оказывается пустой. При этом главной задачей алгоритма является выбор из очереди наиболее подходящих узлов — кандидатов для дальнейшего наращивания кратчайшего дерева за минимальное время. Обычно различают алгоритмы расстановки меток — Label-Setting и алгоритмы изменения (обновления, корректировки) меток — Label-Correcting. Все эти алгоритмы итерационные и отличаются друг от друга способами обновления меток на последовательных итерациях и выбором направлений для поиска КП. Алгоритмы расстановки меток на каждом шаге назначают узлам постоянные кратчайшие метки, которые на последующих итерациях не изменяются. Алгоритмы изменения меток итеративно обновляют значения меток узлов вплоть до последней итерации, на которой определяются кратчайшие метки. Первые, как правило, применяются для нахождения КП в ациклических сетях с произвольными и неотрицательными длинами дуг, а вторые — для более широкого класса задач, а также в случае отрицательных весов дуг. Два комбинаторных подхода отличаются алгоритмическими стратегиями и используемыми структурами данных.

Метод расстановки меток узлов с кратчайшими расстояниями до них впервые предложили Дейкстра (1959 г.) и Данциг (1960 г.). Сложность алгоритма Дейкстры заключается в двух основных операциях: нахождение узла с наименьшей величиной расстояния и совершение релаксации, т.е. изменение значений меток связанных узлов. При простейшей реализации эти операции потребуют соответственно $O(n)$ и $O(1)$ времени. Учитывая, что первая операция выполняется $O(n)$ раз, а вторая — $O(m)$, где n и m — число узлов и дуг сети, получим асимптотическую оценку трудоемкости алгоритма $O(n^2 + m)$. Такая асимптотика оптимальна для плотных сетей, когда $m \approx n^2$. Для разреженных сетей ее можно улучшить за счет уменьшения времени выполнения операций первого типа. Ясно, что временная сложность алгоритма Дейкстры зависит от структур данных, используемых для реализации очереди и представления входной сети. Как было показано, она равна $O(n^2)$ для сетей, представленных весовыми матрицами и неупорядоченной очередью в виде массива. Для сетей, представленных связанными списками смежности и очередью с приоритетами, реализованной как неубывающая пирамида, эффективность равна $O(m \log n)$. Лучшую верхнюю границу можно получить для алгоритма Дейкстры, если реализовать более сложные структуры данных. Развитие методов с использованием очередей с приоритетами имеет длинную историю, включая исследования бинарных пирамид (binary heaps) — Джонсон [10], пирамид Фибоначчи (Fibonacci heaps) — Фридман и Тарьян [11], структур «bucket heaps» [12] и «radix heaps» [13]. Для неотрицательных длин дуг технология «radix heaps» позволила Торупу [14, 15] получить оценку времени $O(m+n\log\log n)$ и линейное время решения задачи для одного специального случая [16]. На основе алгоритма Торупа, Петти и Рамашандран [17], используя технику представления структур данных для построения минимального остовного дерева, разработали новый алгоритм для построения КП от одного источника с неотрицательными длинами дуг. Их алгоритм имеет лучшую оценку, чем $O(m+n\log\log n)$, когда длины дуг мало отличаются друг от друга.

Мейер [18] получил первый алгоритм построения КП в среднем за линейное время. Более ранний такой алгоритм ALD найден Голдбергом [19]. Для графов с ограниченным размером [20] получен более простой Δ-ступенчатый метод. Он использует «bucket queues» и дает хороший параллельный алгоритм для ограниченных графов с малым диаметром.

Другие подходы к решению задачи построения КП, основанные на корректировке меток (Label-Correcting), предложены в работах [21–24]. В отличие от алгоритма Дейкстры, алгоритм Беллмана–Форда решает задачу SSSP и в случае, когда длины дуг могут быть отрицательными, и находит циклы с отрицательной длиной, достижимой из истока. Все они используют процедуры включения и исключения из очереди, приведенные в [23, 25–30]. Операции исключения из очереди, реализованные в [25, 30], эффективны на практике, но имеют псевдополиномиальную оценку времени выполнения. Другие алгоритмы, такие как [27, 29], гарантируют полиномиальное время выполнения операций, однако на практике ведут себя хуже.

В алгоритме, предложенном в [31], комбинируется техника расстановки и обновления меток. Эффективность этого алгоритма сильно зависит от способов выбора пороговых значений меток расстояний, поэтому в каждом конкретном случае требуется тщательная настройка алгоритма.

Несмотря на то что алгоритмы расстановки меток имеют лучшие теоретические оценки трудоемкости, чем алгоритмы обновления меток, их накладные расходы на сортировку узлов-кандидатов могут значительно ухудшить быстродействие на практике, и особенно для разреженных сетей.

Одними из наилучших известных результатов, полученных в настоящее время для решения задачи SSSP комбинаторными методами с неотрицательными длинами дуг, являются $O(m+n\log n)$ [11], $O(m\log\log w)$ [32], $O(m+n\sqrt{\log w})$ [13], где w — максимальная длина дуги, а также $O(mn)$ для большинства алгоритмов обновления меток в случае сетей, не содержащих циклов отрицательной длины [1]. Лучшие временные границы для сетей с отрицательными длинами дуг получены Габовым, Тарьяном [33] и Гольдбергом [34]. Ими соответственно предложены алгоритмы с временной сложностью $O(\sqrt{n}m\log(nw))$ и $O(\sqrt{n}m\log(w))$, где $w\in R$ — максимальное абсолютное значение длины дуги сети, $w\geq 2$.

Обширные экспериментальные исследования по сравнению эффективности различных алгоритмов построения КП между заданными подмножествами узлов сети комбинаторными методами проведены много раз различными авторами [35–38]. В работе [36] указывается, что не существует наилучшего единого алгоритма для решения всего класса задач нахождения КП. Авторы работы исследовали робастность реализованного с применением технологии «double bucket» алгоритма Дейкстры, названного DIKB, и использующего идею топологической сортировки метода обновления меток, названного GOR1, при многочисленных тестах на сетях различной конфигурации и размерности. В [37], пользуясь этими же программами, исследовали поведение алгоритмов DIKB и GOR1 на реальных дорожных сетях. Эксперименты показали, что на сетях с небольшими длинами дуг лучше работает алгоритм Дейкстры с технологией «bucket», а для сетей с более длинными дугами — алгоритм DIKB с «double bucket».

Алгебраичные (матричные) алгоритмы

Рассмотрим кратко матричные APSP — алгоритмы Флойда–Уоршалла [39, 40] и Карре [41, 42], чаще всего используемые для построения КП между всеми узлами сети.

Известно [2], что для решения задачи нахождения всех КП, сформулированной в виде алгебраического (матричного) уравнения Беллмана, может быть использована техника, аналогичная схемам прямого или итеративного метода решения системы линейных уравнений. В частности, использование схем методов Жордана–Гаусса и исключения Гаусса приводит к алгоритмам Флойда–Уоршалла и Карре соответственно.

Итеративные методы Якоби и Гаусса–Зейделя [42] дают схему алгоритмов Беллмана [23] и Форда–Фалкерсона [24]. Метод релаксации Бертсекаса [2] также может быть интерпретирован схемой вычислений Гаусса–Зейделя [43].

Если матричное уравнение Беллмана решать с помощью вычисления обратной матрицы, то схема Морриса [44] для обращения матрицы приводит к индуктивному алгоритму Данцига [45]. Декомпозиционные алгоритмы предложены в работах [46, 47]. В этих алгоритмах сеть разбивается на части, для каждой части находится дерево КП, а затем все деревья объединяются.

Все перечисленные алгоритмы APSP (исключая алгоритмы Беллмана и Форда для нахождения КП между подмножествами узлов) имеют оценку сложности порядка $O(n^3)$ и весьма эффективны для плотных сетей. Для разреженных сетей известен алгоритм Джонсона [48], который использует списки смежных узлов и находит все пути за время $O(n^2 \lg n + nm)$. Он также работает с отрицательными длинами дуг и определяет наличие циклов с отрицательной длиной.

Рассмотрим подробнее алгоритм Карре, поскольку он имеет некоторые преимущества по сравнению с другими матричными алгоритмами. Алгоритм включает LU-процедуру декомпозиции, n последовательных операций «вперед» и «назад» и находит все кратчайшие деревья от всех узлов в сети к одному i, $i = \overline{1, n}$. Такой подход аналогичен схеме решения системы линейных уравнений методом исключения Гаусса, и столбец i в матрице расстояний соответствует кратчайшему дереву к узлу i. В [41] дана графическая интерпретация работы алгоритма и алгебраически доказана его корректность. Преимущество алгоритма заключается в том, что, во-первых, для полных сетей он выполняет меньшее число сравнений из всех возможных — $n(n-1)(n-2)$ [49], в то время как алгоритм Флойда–Уоршалла, выбирая узлы в другом порядке, — все операции сравнения. Во-вторых, поиск всех КП заключается в поиске n кратчайших деревьев к каждому узлу, что дает возможность избежать многих излишних операций при построении КП от заданного числа узлов.

Из линейной алгебры известно, что проблема решения системы линейных уравнений связана с размерностью решаемых задач и точностью получаемых решений. Итеративные методы не гарантируют получение точного решения, но сходятся гораздо быстрее прямых методов. Поскольку при решении матричного уравнения Беллмана не возникает аналогичных проблем, на практике более популярны итеративные алгоритмы построения КП (например, алгоритмы расстановки и обновления меток), чем матричные (алгоритмы Флойда–Уоршалла и Карре).

Известны также алгоритмы, основанные на операциях перемножения матриц [50–54], но все же они хуже алгоритма Флойда–Уоршалла. В [55] показано, что использование технологии перемножения матриц для нахождения всех КП так же сложно, как и нахождение КП от одного источника, и требует $O(n^3)$ времени. В [56] предложен $O(n^{2,5})$-алгоритм для построения КП от одного источника, требующий экспоненциальной памяти. Улучшенный алгоритм Фридмана с оценкой $O(n^3((\log\log n)/\log n)^{1/2})$ предложен в работе [57].

Много алгоритмов разработано с использованием блочной декомпозиции и техники быстрого перемножения матриц. Все эти алгоритмы имеют улучшенные оценки сложности, однако требуют либо целочисленности, либо небольших абсолютных значений длин дуг сети [58–60] или могут быть применены только для невзвешенных или неориентированных сетей [61–63]. Алгоритмы, основанные на технике перемножения матриц, наиболее применимы к плотным сетям и не используют возможностей улучшения быстродействия за счет учета степени разреженности.

До сих пор не было работ, посвященных тщательному анализу экспериментальных исследований эффективности алгоритмов построения всех КП в сети. Принято считать, что матричные алгоритмы требуют большей памяти, применяются к плотным сетям и не подходят для решения задач построения КП для разреженных сетей большой размерности.

Алгоритм Флойда–Уоршалла — лучший матричный алгоритм, однако его применение для сильно разреженных сетей не оправданно. С другой стороны, матричный алгоритм Карре может быть с успехом применен и для разреженных сетей, и его дальнейшее развитие могло бы составить конкуренцию лучшим комбинаторным алгоритмам [64]. Возможность развития алгоритма Карре [9, 65] связана с тем, что, для сетей с фиксированной топологией, КП между всеми узлами сети могут многократно пересчитываться при изменении длин дуг. Поэтому, если соответствующим образом использовать неизменность структуры сети, в алгоритме Карре можно уменьшить трудоемкость операции LU-декомпозиции и на стадии предварительной обработки узлов создать схему построения КП, не требующую полного перебора узлов. Экспериментальные исследования [9, 64] показали, что реализованный на основе такого подхода алгоритм построения всех КП в несколько раз лучше некоторых комбинаторных алгоритмов, используемых для разреженных сетей.

Алгоритмы, базирующиеся на методах решения задач линейного программирования

Рассмотрим алгоритмы построения кратчайших путей, основанные на методах решения задач линейного программирования (Linear programming methods, LP-methods). Чаще всего они применяются для нахождения деревьев кратчайших путей. Если проанализировать работу алгоритма Дейкстры, то можно увидеть сходство алгоритмов расстановки и обновления меток с прямым сетевым симплекс-методом [66]. Такой симплексный алгоритм с трудоемкостью $O(n^3)$ предложен в работе [67]. Другой алгоритм для нахождения КП от одного источника, сравнимый с лучшими алгоритмами корректировки меток и временной сложностью $O(nm)$, можно найти в [68]. Предложен также ряд двойственных симплекс-алгоритмов [2, 43, 69, 70]. Алгоритм Дейкстры может быть представлен и как прямо-двойственный [71]. Практические исследования [66, 72, 73] показали, что для решения реальных задач комбинаторные алгоритмы все-таки лучше алгоритмов, основанных на методах линейного программирования, включая прямой, двойственный и прямо-двойственный симплекс-методы.

Разработки бывшего СССР

В СССР разработка алгоритмов построения КП, в основном, проводилась в 1970–1980-х гг. Среди них следует отметить алгоритм двусторонней очереди Левита [25], работы Филлера [74], Диница и Карзанова [75, 76], посвященные разработке ряда новых эффективных реализаций алгоритма Дейкстры, в основе которых заложена техника использования АТД для представления структур данных [77].

Для сравнения вычислительной эффективности различных алгоритмов построения КП по инициативе и под руководством Л.В. Канторовича в 1983–1984 гг. проводился всесоюзный конкурс «Транспорт-83» [78], на котором были представлены программы, реализующие, в основном, все известные лучшие алгоритмы, в том числе и программа авторов этой статьи. Результаты конкурса показали, что наилучшим, в смысле скорости получения решения, оказался метод двусторонней очереди [25]. Весьма эффективной оказалась также модификация алгоритма Флойда, составившая конкуренцию лучшей реализации алгоритма Дейкстры [75, 76]. Отмечается, что с увеличением размеров сетей разница во времени счета между алгоритмами двусторонней очереди и Дейкстры сокращается, поэтому для сетей, содержащих более 4–5 тысяч узлов, делать выводы о преимуществах первого алгоритма преждевременно. Для нахождения КП от одного источника к одному стоку наилучшим оказался алгоритм Дейкстры.

Основные направления дальнейших исследований

За последние годы эволюция методов решения задачи нахождения КП связана с разработкой и дальнейшим усовершенствованием специальных структур данных (АТД) для представления объектов задачи, позволяющих максимально сократить время построения КП, и созданием параллельных алгоритмов для многопроцессорного решения задачи. Так, например, под руководством Б. Черкасского, А. Фиата, К. Харрелсона, Х. Каплана, Т. Паера, И. Резенштейна группой разработчиков И. Абрахамом, Д. Деллингом, А. Гольдбергом и Р. Верником интенсивно разрабатывался проект, посвященный теоретическому и экспериментальному изучению алгоритмов КП и связанных с ними задач (http://research. microsoft.com/en-us/projects/SPA/). Показано, что для решения отдельных SSSP-задач существуют «почти оптимальные» алгоритмы в теории и на практике, в то же время для решения более широкого класса задач, включая и APSP-проблему, имеются предпосылки для улучшения уже существующих алгоритмов. Основные усилия проекта направлены на решение задачи нахождения КП между заданной парой вершин при ограничениях на доступный объем оперативной памяти для проведения процедуры препроцессинга очень большого графа. Такая задача характерна для вычисления расстояний на интернет-картах, таких как Google Maps, Yahoo! Maps, Bing Maps. Были также разработаны методы ускорения классических алгоритмов построения КП, включая методы landmark-based A* search и reach-based pruning и их комбинации. Полученные алгоритмы очень практичны и могут успешно использоваться на серверах, настольных и портативных компьютерах, включая планшеты, смартфоны, автомобильные системы навигации. В рамках проекта также были рассмотрены задачи более гибкого планирования и выбора разумных альтернативных маршрутов передвижения в масштабе реального времени, в случае затора на дорогах, а также выбора индивидуальных маршрутов по заданным критериям и предпочтениям. Другие задачи проекта были связаны с нахождением более дешевых путей передвижения, минимальных циклов, с аппаратным ускорением построения дерева кратчайших путей (Parallel Hardware-Accelerated Shortest Path Trees, PHAST) с использованием многоядерных процессоров и новейших графических адаптеров. Всю информацию о результатах выполнения проекта можно найти в Интернете, а также в многочисленных работах руководителей и исполнителей проекта [79–81].

Наряду с задачами построения КП по одному критерию, значительный интерес представляют многокритериальные задачи о путях (multicriteria (multiobjective) shortest path problems, MSPP). Такие задачи естественным образом возникают во многих приложениях, когда имеется несколько различных параметров, характеризующих узлы и дуги сети, и интенсивно исследуются при проектировании транспортных, информационных и телекоммуникационных сетей. В большинстве случаев выбор «оптимальных» кратчайших путей по экстремальным значениям этих параметров представляет сложную задачу, так как параметры могут быть противоречивыми. Основываясь на интуиции, лицу, принимающему решение (ЛПР). трудно отдать предпочтение какому-либо конкретному варианту КП по нескольким критериям. Поэтому возникает необходимость в применении некоторой методологии выбора наиболее приемлемых вариантов, базирующейся на строгих математических методах и здравой интуиции.

Для решения многокритериальных задач КП как начальное ядро используются различные модификации уже рассмотренных алгоритмов построения КП по одному критерию (комбинаторные Label-Setting, Label-Correcting, матричные и LP-алгоритмы, основанные на современных АТД, таких как пирамиды Фибоначчи, d-арные кучи и т.п.). Сначала находится множество допустимых вариантов решения по критериям Парето, Слейтера, вероятностным, лексикографическим и др., и далее с использованием множества определенных методик отбираются наиболее приемлемые.

Известно, что в общем случае проблема MSPP является NP-трудной [82]. Одной из первых работ, посвященных систематическому исследованию многокритериальных задач о путях, является обзорная статья [83], в которой определяется класс задач с экспоненциальными оценками сложности в худшем случае и приведена полностью полиномиальная схема решения (Fully Polynomial Time Approximation Schemes, FPTAS) для двухкритериальных задач. Для многокритериальных задач на ориентированных ациклических сетях также известна полиномиальная схема решения [84]. В настоящее время, судя по литературным источникам и публикациям в Интернете, для случаев, когда число критериев оптимизации превышает два, пока не достигнуто значительных результатов.

Из-за ограниченного объема в статье не приводится обзор по многокритериальным задачам, отметим только, что достаточно хороший обзор по их решению можно найти в [85].

Наряду с методами и алгоритмами построения однокритериальных и многокритериальных КП, значительный интерес представляет также разработка алгоритмов лексикографической оптимизации нахождения кратчайших путей, удовлетворяющих требованиям упорядоченной последовательности критериев по нескольким параметрам [86]. Такие алгоритмы, например, востребованы для решения ряда задач распределения и маршрутизации потоков в многопродуктовых сетях [87–90].

Заключение

Проведенный обзор и анализ алгоритмов построения однокритериальных кратчайших путей на сетях с неотрицательными длинами дуг показал, что в настоящее время одними из наилучших являются алгоритмы расстановки меток (Label-Setting) с оценками временной сложности $O(m+n\log n)$ [11], $O(m\log\log w)$ [32], $O(m+n\sqrt{\log w})$ [13], где n и m — число узлов и дуг в сети, а w — максимальная длина дуги. Для сетей с отрицательными дугами, не содержащих циклов отрицательной длины, для большинства алгоритмов обновления меток (Label-Correcting) получены оценки порядка $O(mn)$ [1]. Лучшие временные границы $O(\sqrt{n}m\log(nw))$ и $O(\sqrt{n}m\log(w))$, где $w\in R$ — максимальное абсолютное значение длины дуги, получены в работах [33, 34]. В [36] указывается, что не существует наилучшего единого алгоритма для решения всего класса задач нахождения КП. За последние годы эволюция методов решения задачи нахождения КП была связана с разработкой и дальнейшим усовершенствованием специальных структур данных для представления объектов задачи, позволяющих максимально сократить время построения КП, и созданием параллельных алгоритмов для многопроцессорного решения задачи. Показано, что для решения отдельных SSSP-задач существуют «почти оптимальные» алгоритмы в теории и на практике, в то же время для решения более широкого класса задач, включая и APSP-проблему, имеются предпосылки для улучшения уже существующих алгоритмов за счет аппаратного ускорения построения дерева кратчайших путей (PHAST) с использованием многоядерных процессоров и новейших графических адаптеров. Для многокритериальных задач о путях известны полностью полиномиальные схемы решения только для двухкритериальных задач [85]. Для случаев, когда число критериев оптимизации превышает два, пока не достигнуто значительных результатов.

О.М. Трофимчук, В.О. Васянін, Л.П. Ушакова

ОГЛЯД МЕТОДІВ І АЛГОРИТМІВ ПОБУДОВИ НАЙКОРОТШИХ ШЛЯХІВ ТА ПЕРСПЕКТИВИ ЇХ РОЗВИТКУ

Незважаючи на численність робіт, пов'язаних з проблемою знаходження найкоротших шляхів (НШ), увага до розробки ефективних за швидкодією алгоритмів побудови НШ не зменшується. Це, в першу чергу, пояснюється тим, що в переважній більшості випадків такі алгоритми часто використовуються для вирішення окремих підзадач в багатьох додатках в різних областях природознавства і час вирішення загальної оптимізаційної задачі в значній мірі визначається часом побудови НШ. Розглядається три групи однокритеріальних алгоритмів: мережеві комбінаторні алгоритми; алгебраїчні або матричні алгоритми; алгоритми, що базуються на методах вирішення задач лінійного програмування (Linear programming, LP-methods). У статті дано огляд, аналіз та класифікацію методів і алгоритмів побудови найкоротших шляхів на мережах і графах між заданими підмножинами вузлів мережі (Single Source Shortest Path, SSSP) і між усіма парами вузлів (Shortest Path Tree, SPT або All Pairs Shortest Paths, APSP). Наведено оцінки часової складності найкращих відомих алгоритмів для вирішення задач SSSP і APSP комбінаторними, матричними і LP-методами для мереж з невід'ємними довжинами дуг і мереж з від'ємними довжинами дуг і циклами від'ємної довжини. Відзначається, що для вирішення окремих SSSP-задач існують «майже оптимальні» алгоритми в теорії і на практиці, в той же час для вирішення більш широкого класу задач, включаючи і APSP-проблему, є передумови для поліпшення вже існуючих алгоритмів. За останні роки еволюція методів вирішення задачі знаходження НШ була пов'язана з розробкою та подальшим удосконаленням ефективних структур абстрактних типів даних для представлення об'єктів задачі і створенням паралельних алгоритмів для багатопроцесорного розв'язання задачі. Визначено основні напрямки подальших досліджень з розробки ефективних методів і алгоритмів вирішення завдань знаходження найкоротших шляхів.

Ключові слова: задачі побудови найкоротших шляхів, методи і алгоритми, обчислювальна ефективність.

A.N. Trofymchuk, V.A. Vasyanin, L.P. Ushakova

OVERVIEW OF METHODS AND ALGORITHMS OF CONSTRUCTING SHORTEST PATHS AND PROSPECTS OF THEIR DEVELOPMENT

Despite the numerous works related to the problem of finding the shortest paths (SP), attention to the development of speed-efficient algorithms for constructing SP is not reduced. This is primarily due to the fact that in the overwhelming majority of cases, such algorithms are often used to solve individual subtasks in many applications in various fields of natural science, and the time to solve the general optimization problem is largely determined by the time of constructing the SP. Three groups of single-criterion algorithms are considered: network combinatorial algorithms; algebraic or matrix algorithms; algorithms based on methods for solving linear programming problems (LP-methods). The article provides an overview, analysis and classification of methods and algorithms for constructing the shortest paths on networks and graphs between given subsets of the network nodes (Single Source Shortest Path, SSSP) and between all pairs of nodes (Shortest Path Tree, SPT or All Pairs Shortest Paths, APSP). Estimates of the time complexity of the best known algorithms for solving SSSP and APSP problems by combinatorial, matrix, and LP methods for networks with non-negative arcs lengths and networks with negative arcs lengths and cycles of negative lengths are given. It is noted that for solving individual SSSP problems, there are «almost optimal» algorithms in theory and practice, while at the same time, for solving a wider class of problems, including the APSP problem, there are prerequisites for improving existing algorithms. In recent years, the evolution of methods for solving the problem of finding SP has been associated with the development and further improvement of effective structures of abstract data types for representing objects of the problem and the creation of parallel algorithms for multiprocessor solving the problem. The main directions of further research on the development of effective methods and algorithms for solving the problems of finding the shortest paths are determined.

Keywords: shortest path problems, methods and algorithms, computational efficiency.

1. Ahuja R.K., Magnanti T.L., Orlin J.B. Network flows: theory, algorithms, and applications. Prentice-Hall, Inc. Upper Saddle River, New Jersey, 1993. 846 p.
2. Bertsekas D. Network optimization: continuous and discrete models. Belmont : Athena Scientific, 1998. 608 p.
3. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е изд.: Пер. с англ. М. : Издательский дом «Вильямс», 2005. 1296 с.
4. Mehlhorn K., Sanders P. Algorithms and Data Structures. The Basic Toolbox. Karlsruhe; Saarbrucken : Springer, 2008. 300 p. DOI 10.1007/978-3-540-77978-0.
5. Fakcharoenphol J., Rao S. Planar graphs, negative weight edges, shortest paths, and near linear time. *Journal of Computer and System Sciences*. 2006. **72,** N 5. P. 868–889. DOI: 10.1016/j.jcss.2005.05.007.

6. Sanders P., Schultes D. Engineering fast route planning algorithms. *In 6th Workshop on Experimental Algorithms*, *Lecture Notes in Computer Science*. Berlin; Heidelberg : Springer, 2007. **4525**. P. 23–36. DOI: 10.1007/978-3-540-72845-0_2.
7. Schultes D. Route planning in road networks. PhD thesis, 2008. 235 p.
8. A survey of shortest-path algorithms. A. Madkour, W. Aref, F. Rehman, M. Rahman, S. Basalamah. Cornell University, 2017. 26 p. DOI: arXiv:1705.02044.
9. Wang I-Lin. Shortest paths and multicommodity network flows: School of industrial and systems engineering Georgia Institute of technology: In Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy. 2003. 227 p.
10. Johnson E. On shortest paths and sorting. *In Proceedings of the ACM 25th annual conference*. 1972. **1**. P. 510–517. https://doi.org/10.1145/800193.569965.
11. Fredman M., Tarjan R. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*. 1987. **34**, N 3. P. 596–615. https://doi.org/10.1145/28869.28874.
12. Dial R. Algorithm 360 shortest path forest with topological ordering. *Communications of the ACM*. 1969. **12**, N 11. P. 632–633. DOI: 10.1145/363269.363610.
13. Ahuja R., Mehlhorn K., Orlin J., Tarjan R. Faster algorithms for the shortest path problem. *Journal of ACM*. 1990. **37**, N 2. P. 213–223. https://doi.org/10.1145/77600.77615.
14. Thorup M. Floats, integers, and single source shortest paths. *Journal of Algorithms*. 2000. **35**, N 2. P. 189–201. https://doi.org/10.1006/jagm.2000.1080.
15. Thorup M. Integer priority queues with decrease key in constant time and the single source shortest paths problem. *Journal of Computer and System Sciences*. 2004. **69**, N 3. P. 330–353. https://doi.org/10.1016/j.jcss.2004.04.003.
16. Thorup M. Undirected single-source shortest paths with positive integer weights in linear time. *Jounral of ACM*. 1999. **46**, N 3. P. 362–394. https://doi.org/ 10.1145/316542.316548.
17. Pettie S., Ramachandran V. Computing shortest paths with comparisons and additions. *SODA '02 Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, San Francisco, California. January 06–08, 2002. P. 267–276. DOI: 10.1145/545381.545417.
18. Meyer U. Average-case complexity of single-source shortest-path algorithms: Lower and upper bounds. *Journal of Algorithms*. 2003. **48**, N 1. P. 91–134. https://doi.org/10.1016/S0196-6774(03)00046-4.
19. Goldberg A.V. A simple shortest path algorithm with linear average time. *In 9th European Symposium on Algorithms*: of Lecture Notes in Computer Science : Springer, 2001. **2161**. P. 230–241. https://doi.org/10.1007/3-540-44676-1_19.
20. Meyer U., Sanders P. Δ-stepping: a parallelizable shortest path algorithm. *Journal of Algorithms*. 2003. **49**, N 1. P. 114–152. https://doi.org/10.1016/S0196-6774(03)00076-2.
21. Ford Jr. L. Network flow theory. Santa Monica, California: The RAND Corp., 1956. P. 923.
22. Moore E. The shortest path through a maze. Part II. *In Proceedings of the International Symposium on theory of Switching*: *The Annals of the Computation Laboratory of Harvard University*, 1957. P. 285–292.
23. Bellman R. On a routing problem. *Quarterly of Applied Mathematics*. 1958. **16**. P. 87–90. DOI: https://doi.org/10.1090/qam/102435.
24. Ford Jr. L., Fulkerson D. Flows in networks. Princeton, NJ : Princeton University Press, 1962. 332 p.
25. Левит Б.Ю., Лившиц В.Н. Нелинейные сетевые транспортные задачи. М. : Транспорт, 1972. 144 с.
26. Bertsekas D. A simple and fast label correcting algorithm for shortest paths. *Networks*. 1993. **23**, N 8. P. 703–709. https://doi.org/10.1002/net.3230230808.
27. Goldberg A.A., Radzik T. A heuristic improvement of the Bellman-Ford algorithm. *Applied Mathematics Letters*. 1993. **6**, N 3. P. 3–6. https://doi.org/10.1016/0893-9659(93)90022-F.
28. Goldfarb D., Hao J., Kai S. Shortest path algorithms using dynamic breadth-first search. *Networks*. 1991. **21**, N 1. P. 29–50. DOI: 10.1002/net.3230210105.
29. Pallottino S. Shortest-path methods: complexity, interrelations and new propositions. *Networks*. 1984. **14**, N 2. P. 257–267. https://doi.org/10.1002/net.3230140206.
30. Pape U. Implementation and efficiency of Moore algorithms for the shortest root problem. *Mathematical Programming*. 1974. **7**. P. 212–222. DOI:10.1007/BF01585517.

31. Glover F., Glover R., Klingman D. Computational study of an improved shortest path algorithm. *Networks*. 1984. **14**, N 1. P. 25–36. DOI: 10.1002/net.3230140103.
32. Van Emde Boas P., Kaas R., Zijlstra E. Design and implementation of an efficient priority queue. *Math. Syst. Theory*. 1976. **10**, N 1. P. 99–127. https://doi.org/ 10.1007/BF01683268.
33. Gabow H., Tarjan R. Faster scaling algorithms for network problems. *SIAM Journal on Computing*. 1989. **18**, N 5. P. 1013–1036. DOI: 10.1137/0218069.
34. Goldberg A. Scaling algorithms for the shortest paths problem. *SIAM Journal on Computing*. 1995. **24**, N 3. P. 494–504. https://doi.org/10.1137/S0097539792231179.
35. Hung M., Divoky J. A computational study of efficient shortest path algorithms. *Computers and Operations Research*. 1988. **15**, N 6. P. 567–576.
36. Cherkassky B., Goldberg A., Radzik T. Shortest paths algorithms: theory and experimental evaluation. *Mathematical Programming*. 1996. **73**, N 2. P. 129–174. https://doi.org/10.1007/BF02592101.
37. Zhan F., Noon C. Shortest path algorithms: an evaluation using real road networks. *Transportation Science*. 1998. **32**, N 1. P. 65–73. DOI: 10.1287/trsc.32.1.65.
38. Mehlhorn K., Näher S. The LEDA platform for combinatorial and geometric computing. Cambridge University Press, 1999. P. 316–359. DOI: 10.1145/204865.204889.
39. Floyd R. Algorithm 97, shortest path. *Comm. ACM*. 1962. **5**, N 6. P. 345. https://doi.org/10.1145/367766.368168.
40. Warshall S. A theorem on Boolean matrics. *Journal of ACM*. 1962. **9**, N 1. P. 11–12. https://doi.org/10.1145/321105.321107.
41. Carre B. A matrix factorization method for finding optimal paths through networks. *In I.E.E. Conference Publication*: Computer-Aided Design, 1969. **51**. P. 388–397.
42. Carre B. An algebra for network routing problems. *Journal of Institute of Mathematics and Its Applications*. 1971. **7**. P. 273–294.
43. Pallottino S., Scutellà M. Dual algorithms for the shortest path tree problem. Networks. 1997. **29**, N 2. P. 125–133. https://doi.org/10.1002/(SICI)1097-0037(199703)29:2< 125::AID-NET7>3.0.CO;2-L.
44. Morris J. An escalator process for the solution of linear simultaneous equations. *Philos. Mag*. 1946. Series 7. **37, N** 265. P. 106–120. https://doi.org/10.1080/14786444608561331.
45. Dantzig G. All shortest routes in a graph. *In Theory of Graphs (International Symposium, Rome, 1966)*. New York : Gordon and Breach, 1967. P. 91–92.
46. Mills G. A decomposition algorithm for the shortest-route problem. *Operations Research*. 1966. **14**, N 2. P. 279–291. https://doi.org/10.1287/opre.14.2.279.
47. Hu T. A decomposition algorithm for shortest paths in a network. *Operations Research*. 1968. **16**, N 1. P. 91–102. https://www.jstor.org/stable/168405.
48. Johnson D. Efficient algorithms for shortest paths in sparse networks. *Journal of the ACM*. 1977. **24**, N 1. P. 1–13. https://doi.org/10.1145/321992.321993.
49. Nakamori M. A note on the optimality of some all-shortest-path algorithms. *Journal of the Operations Research Society of Japan*. 1972. **15**, N 4. P. 201–204.
50. Farbey B.A., Land A.H., Murchland J.D. The cascade algorithm for finding all shortest distances in a directed graph. *Management Science*. 1967. **14**, N 1 . P. 19–28. https://doi.org/10.1287/mnsc.14.1.19.
51. Hu T. Revised matrix algorithms for shortest paths. *SIAM Journal of Applied Mathematics*. 1967. **15**, N 1. P. 207–218. https://www.jstor.org/stable/2946165.
52. Land A., Stairs S. The extension of the cascade algorithm to large graphs. *Management Science*. 1967. **14**, N 1. P. 29–33. https://doi.org/10.1287/mnsc.14.1.29.
53. Shimbel A. Applications of matrix algebra to communication nets. *Bulletin of Mathematical Biophysics*. 1951. **13**. P. 165–178. https://doi.org/10.1007/BF02478225.
54. Yang L., Chen W. An extension of the revised matrix algorithm. In IEEE International Symposium on Circuits and Systems, (Portland, Oregon): IEEE, 1989. P. 1996–1999. DOI: 10.1109/ISCAS.1989.100763.
55. Aho A., Hopcroft J., Ullman J. The design and analysis of computer algorithms. Addison-Wesley, 1974. 480 p.
56. Fredman M. New bounds on the complexity of the shortest path problems. *SIAM Journal on Computing*. 1976. **5**, N 1. P. 83–89. https://doi.org/10.1137/0205006.

57. Takaoka T. A new upper bound on the complexity of the all pairs shortest path problem. *Information Processing Letters*. 1992. **43**, N 4. P. 195–199. https://doi.org/10.1016/0020-0190(92)90200-F.
58. Alon N., Galil Z., Margalit O. On the exponent of the all pairs shortest path problem. *Journal of Computer and System Sciences*. 1997. **54**, N 2. P. 255–262. https://doi.org/10.1006/jcss.1997.1388.
59. Takaoka T. Subcubic cost algorithms for the all pairs shortest path problem. *Algorithmica*. 1998. **20**, N 3. P. 309–318. https://doi.org/10.1007/PL00009198.
60. Zwick U. All pairs shortest paths in weighted directed graphs — exact and almost exact algorithms. *In Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science* (Palo Alto, California), November, 1998. P. 310–319. DOI: 10.1109/SFCS.1998.743464.
61. Galil Z., Margalit O. All pairs shortest paths for graphs with small integer length edges. *Journal of Computer and System Sciences*. 1997. **54**, N 2. P. 243–254. https://doi.org/10.1006/jcss.1997.1385.
62. Galil Z., Margalit O. All pairs shortest distances for graphs with small integer length edges. *Information and Computation*. 1997. **134**. P. 103–139.
63. Seidel R. On the all–pairs-shortest-path problem in unweighted undirected graphs. *Journal of Computer and System Sciences*. 1995. **51**, N 3. P. 400–403. https://doi.org/10.1006/jcss.1995.1078.
64. Backhouse R., Carre B. A comparison of Gaussian and Gauss-Jordan elimination in regular algebra. *International Journal of Computer Mathematics*. 1982. **10**, , N 3-4. P. 311–325. https://doi.org/10.1080/00207168208803290.
65. Goto S., Ohtsuki T., Yoshimura T. Sparse matrix techniques for the shortest path problem. IEEE *Transactions on Circuits and Systems*. 1976. **23**, N 12. P. 752–758. DOI: 10.1109/TCS.1976.1084155.
66. Dial R., Glover F., Karney D., Klingman D. A computational analysis of alternative algorithms and labeling techniques for finding shortest path trees. *Networks*. 1979. **9**, N 3. P. 215–248. https://doi.org/10.1002/net.3230090304.
67. Goldfarb D., Hao J., Kai S. Efficient shortest path simplex algorithms. *Operations Research*. 1990. **38**, N 4. P. 624–628. https://www.jstor.org/stable/171080.
68. Goldfarb D., Jin Z. An $O(nm)$-time network simplex algorithm for the shortest path problem. *Operations Research*. 1999. **47**, N 3. P. 445–448. https://doi.org/ 10.1287/opre.47.3.445.
69. Bertsekas D., Pallottino S., Scutellà M. Polynomial auction algorithms for shortest paths. *Computational Optimization and Applications*. 1995. **4**, N 2. P. 99–125. https://doi.org/10.1007/BF01302891.
70. Cerulli R., Festa P., Raiconi G. Graph collapsing in shortest path auction algorithms. *Computational Optimization and Applications.* 2001. **18**, N 3. P. 199–220. https://doi.org/10.1023/A:1011246118315.
71. Papadimitriou C., Steiglitz K. Combinatorial optimization: algorithms and complexity. Englewood Cliffs, NJ : Prentice-Hall, 1982. 496 p.
72. Burton D. On the inverse shortest path problem. Department de Mathematique, Faculty des Sciences, Faculties Universities Notre-Dame de la Paix de Namur, 1993. 126 p.
73. Larsen J., Pedersen I. Experiments with the auction algorithm for the shortest path problem. *Nordic Journal of Computing*. 1999. **6**, N 4. P. 403–421.
74. Филлер М.Ф. Один алгоритм нахождения кратчайших путей. Исследования по дискретной оптимизации. М. : Наука, 1976. С. 359–364.
75. Диниц Е.А. Экономные алгоритмы нахождения кратчайших путей в сети. *Тр. Всесоюз. НИИ системных исследований.* М. : ВИНИТИ, 1978. Вып. 4. С. 36–44.
76. Диниц Е.А., Карзанов А.В. Программа расчета кратчайших путей в сети. *Тр. Всесоюз. НИИ системных исследований.* М.: ВИНИТИ,1978. Вып. 4. С. 44–48.
77. Карзанов А.В. Справочная для выборки максимального элемента и ее приложения. Исследования по дискретной оптимизации. Под ред. А.А. Фридмана. М. : Наука, 1976. С. 340–359.
78. Итоги конкурса программ построения матрицы кратчайших расстояний «Транспорт-83». *Экономика и математические методы*. М. : ЦЭМИ РАН, 1985. **XII**, вып. 3. С. 565–567.

79. Shortest-path feasibility algorithms: An experimental evaluation. B. Cherkassky, L. Georgiadis, A. Goldberg, R. Tarjan, R. Werneck. *In ACM Journal of Experimental Algorithmics*. 2009. **14**, N 2. P. 2.7:1–2.7:37. https://doi.org/10.1145/1498698.1537602.
80. Delling D., Goldberg A., Nowatzyk A., Werneck R. PHAST: Hardware-accelerated shortest path trees. *Journal of Parallel and Distributed Computing*. 2013. **73**, N 7. P. 940–952. https://doi.org/10.1016/j.jpdc.2012.02.007.
81. Madduri K., Bader D., Berry J., Crobak J. Parallel shortest path algorithms for solving large-scale instances. In the shortest path problem: ninth DIMACS Implementation Challenge, ser. DIMACS Book, C Demetrescu, A.V. Goldberg and D.S. Johnson, eds. *American Mathematical Society*. 2009. **74**. P. 249–290. DOI: https://doi.org/10.1090/dimacs/074.
82. Garey M., Johnson D. Computers and intractability: A guide to the theory of NP completeness. San Francisco, CA: W.H. Freeman, 1979. 338 p.
83. Hansen P. Bicriteria path problems. In lecture notes in economics and mathematical systems: G. Fandel & T. Gal, eds. Berlin: Springer Verlag, 1980. **177**. P. 109–127. https://doi.org/10.1007/978-3-642-48782-8_9.
84. Warburton A. Approximation of Pareto optima in multiple-objective, shortest-path problems. *Oper. Res*. 1987. **35**, N 1. P. 70–79. https://doi.org/10.1287/opre.35.1.70.
85. Ehrgott M., Gandibleux X. Multiple Criteria optimization. State of the art. Annotated bibliographic surveys. Boston, MA : Kluwer, 2002. 496 p. DOI: 10.1007/b101915.
86. Vasyanin V.A. A two-criterion lexicographic algorithm for finding all shortest paths in networks. *Cybernetics and Systems Analysis*. 2014. **50**, N 5. P. 759–767. DOI: 10.1007/s10559-014-9666-9.
87. Trofymchuk O.M., Vasyanin V.A. Simulation of packing, distribution and routing of small-size discrete flows in a multicommodity network. *Journal of Automation and Information Sciences*. 2015. **47**, N 7. P. 15–30. DOI: 10.1615/JAutomatInfScien.v47.i7.30.
88. Trofymchuk O.M., Vasyanin V.A., Vasyanin V.A. Problem of distribution and routing of transport blocks with mixed attachments and its decomposition. *Journal of Automation and Information Sciences*. 2015. **47**, N 2. P. 56–69. JAutomatInfScien DOI: 10.1615/.zv47.i2.60.
89. Kuzmenko V.N. Complexity of one packing optimization problem. *Cybernetics and Systems Analysis*. 2016. **52**, N 1. P. 76–84. DOI: 10.1007/s10559-016-9802-9.
90. Trofymchuk O.M., Vasyanin V.A., Kuzmenko V.N. Optimization algorithms for packing of small-lot correspondence in communication networks. *Cybernetics and Systems Analysis*. 2016. **52**, N 2 P. 258–268. DOI: 10.1007/s10559-016-9822-5.

Получено 18.07.2019
После доработки 31.03.2020