

ROUTING THE NETWORKS WITH NEGATIVE WEIGHTS OF LINKS IN THE MATLAB OPTIMIZATION PACKAGE

V. V. Saharov, A.A. Chertkov, L.B. Ochina

Admiral Makarov State University of Maritime and Inland Shipping,
St. Petersburg, Russian Federation

The subject of research in the field of unmanned maritime and river navigation vessels is the new technological principles and models of unmanned navigation with high reliability and accuracy in the autonomous regimes on the specified path, with the use of satellite navigation, electronic charts and ground-based radio navigation systems. Non-crew vessels management from the shore requires the implementation of group management by the mobile objects, taking into account the update of the local maps of the shipping situation. When selecting the vessels routes, it is necessary to solve the operational problem of determining the shortest vessel route (Autonomous mobile object) at the specific navigation conditions, which directly defines the energy efficiency of the vessel in the voyage. The solution of the operational problem of selecting the shortest paths (routes) for a group of vessels moving in the direction of the specified targets with known coordinates located in a limited space is presented in the paper. A model and an optimization algorithm allowing, in comparison with existing solutions, to significantly expand the class of tasks by introducing the negative weights of individual sections of the path (graph arcs) and make the transition to solving high-dimensional problems are proposed. The key issue in the problem of automation and formation of the routes is the selection of the mathematical apparatus not only for the calculation of the shortest routes, but also for their recovery. For this purpose, a recursive optimization method based on the integer linear programming is proposed for high-dimensional tasks. The practical implementation of the proposed algorithm is demonstrated by the example of calculating a network model with a complex topology, using an iterative procedure for the program compiled in the MATLAB codes. It is shown that the computer model realized on its basis has properties of convergence of calculations and convenience of practical use. In contrast to the existing ones, the proposed model allows to remove the restrictions associated with the presence of negative weights and cycles at the network, to automate the calculations of the shortest paths in the places of branching by means of digital technologies. The correctness of the obtained solutions is confirmed, which makes it possible to use the model and algorithm as a component of the tasks tool designed to control the transport process.

Keywords: automation, algorithm, route, shortest paths, network model, network links, negative weight links, optimization, recursive procedure.

For citation:

Saharov, Vladimir V., Alexander A. Chertkov, and Ludmila B. Ochina. "Routing the networks with negative weights of links in the MATLAB optimization package." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admiral S.O. Makarova* 11.2 (2019): 230–242. DOI: 10.21821/2309-5180-2019-11-2-230-242.

УДК 681.5

МАРШРУТИЗАЦИЯ СЕТЕЙ С ОТРИЦАТЕЛЬНЫМИ ВЕСАМИ ЗВЕНЬЕВ В ПАКЕТЕ ОПТИМИЗАЦИИ MATLAB

В. В. Сахаров, А. А. Чертков, Л. Б. Очина

ФГБОУ ВО «ГУМРФ имени адмирала С. О. Макарова»,
Санкт-Петербург, Российская Федерация

В рамках исследований новых технологических принципов и моделей беспилотного судовождения с использованием спутниковых средств навигации, электронных карт и наземных радионавигационных систем решается задача управления безэкипажными судами, функционирующими в автономных режимах при следовании по заданному пути с целью осуществления группового управления подвижными объектами с учетом обновления локальных карт судоходной обстановки. При выборе маршрутов движения судов требуется решать операционную задачу определения наикратчайшего пути следования

судна (автономного подвижного объекта) в конкретных навигационных условиях, что непосредственно определяет энергоэффективность судна в рейсе. В работе представлено решение операционной задачи выбора наикратчайших путей (маршрутов) для группы судов, движущихся в направлении заданных целей с известными координатами, расположенными в ограниченном пространстве. Предложен алгоритм оптимизации, позволивший, в сравнении с существующими алгоритмами, значительно расширить класс решаемых задач путем введения отрицательных весов отдельных участков пути (дуг графа) и осуществить переход к решению задач высокой размерности. Ключевым вопросом в проблеме автоматизации и формирования маршрутов является выбор математического аппарата не только для расчета кратчайших маршрутов, но и для их восстановления. С этой целью для задач большой размерности в работе предложен рекурсивный метод оптимизации, базирующийся на целочисленном линейном программировании. Практическая реализация предложенного алгоритма демонстрируется на примере расчета сетевой модели со сложной топологией с применением итерационной процедуры по программе, составленной в кодах MATLAB. Показано, что реализованная на ее основе компьютерная модель обладает свойствами сходимости вычислений и удобством практического использования. В отличие от существующих предложенная модель позволяет снять ограничения, связанные с наличием отрицательных весов и циклов на сети, и автоматизировать расчеты кратчайших путей в местах разветвлений средствами цифровых технологий. Подтверждена корректность полученных решений, что позволяет использовать модель и алгоритм как составляющую инструмента задач, предназначенных для управления транспортным процессом.

Ключевые слова: автоматизация, алгоритм, маршрут, кратчайшие пути, сетевая модель, звенья сети, отрицательные веса звеньев, оптимизация, рекурсивная процедура.

Для цитирования:

Сахаров В. В. Маршрутизация сетей с отрицательными весами звеньев в пакете оптимизации MATLAB / В. В. Сахаров, А. А. Чертков, Л. Б. Очина // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2019. — Т. 11. — № 2. — С. 230–242. DOI: 10.21821/2309-5180-2019-11-2-230-242.

Введение (Introduction)

Вопросы выбора маршрутов движения судов в условиях применения безэкипажных технологий организации движения становятся исключительно актуальными, требующими оперативных решений при изменениях ситуации как на линии, так и в портах назначения. Так, например, для экономии топлива и энергии на линии необходимо выполнять оценки резерва ходового времени судна до порта назначения и использовать его без изменений объемов транспортной работы, совершаемой судном. При движении судов в условиях штормовой погоды и групповом управлении судами возможны решения, связанные с коррекцией расписания, также приводящей к образованию резервов ходового времени. Резервы экономии топлива и энергии могут быть также оценены и использованы при прохождении шлюзованных участков пути, что требует гибких оперативных решений диспетчерских служб. Выбор наикратчайших либо наиболее эффективных маршрутов движения судов с учетом загрузки, стоимости груза, расходов на перевозки, характеристик логистики и других факторов — это тот класс задач, который требует для принятия оптимальных решений создания новых операционных технологий, моделей и алгоритмов, базирующихся на инструментариях компьютерных систем. Оценка и выбор оптимальных (наикратчайших) маршрутов для организации движения судов могут производиться методами организации движения с использованием исследования операций. Этот класс задач принято решать с помощью теории графов, с введением узлов и дуг с различными весами и другими способами, что позволяет в удобной форме графически интерпретировать как объекты, так и их модели.

Для моделирования на графах стохастических и детерминированных систем созданы многочисленные алгоритмы, фактически формирующие инструменты для решения задач данного класса. К основополагающим работам в этой предметной области следует отнести алгоритмы Дейкстры [1], Беллмана – Форда [2], модифицированный алгоритм Беллмана – Форда, подробно рассмотренные в работах [3]–[7]. С развитием интеллектуальных методов и систем, нейронных сетей, генетических алгоритмов, а также Fuzzy Systems, моделей ANFIS и других появились

алгоритмы для принятия решений в системах высокой размерности, представленные в работах [8]–[10], в которых определены новые операционные пространства для координации действий по управлению подвижными объектами.

Оптимизационные задачи на операционных пространствах приобретают направленность на научное обоснование и решение практических задач выбора маршрутов движения судов, использования мобильных роботов, видеороботов в системах обслуживания и оказания транспортных услуг в условиях рыночного спроса [11]–[14]. Научная и практическая значимость сетевых моделей существенно расширяется. Их применение для формирования маршрутов движения судов, сходящихся в узлах и на операционных подпространствах на подходах к портам, при лоцманских проводках, в условиях приливов и отливов требует принятия особых мер по обеспечению безопасности и безаварийности выполнения всех технологических операций. Системное решение этих задач требует разработки и создания алгоритмической и программной инструментальной базы для безэкипажного управления судами с берега.

Методы и материалы (Methods and Materials)

Рассмотрим оценку и выбор кратчайших маршрутов движения судов на выбранных операционных подпространствах на примере разветвленной сети с системой целей, моделирующих реальную ситуацию. Будем считать, что известными являются координаты расположения целей на подпространствах. Задача состоит в выборе маршрутов, движение по которым к цели потребует минимальных временных затрат. Рабочее время при управлении группой судов оценим по времени $T_{\max}$, необходимому для прибытия к наиболее удаленной цели. Скорость движения судов, согласно принятой терминологии в маршрутизации сетей — «агентов», выберем одинаковой для всех объектов. Временем «обработки» судов при достижении целей пренебрегаем. Принятые условия позволяют принять скорость «агента», равной единице. Тогда время выполнения операции условно можно принять равным длине маршрута движения судна.

В процессе решения задачи используем рекуррентный способ последовательных операций, который применяют для параметрической оценки маршрута каждого «агента». На нулевой итерации может быть выбран произвольно любой «агент», для которого далее находится самая близко расположенная к нему цель. Далее фиксируется расположение «агента» в новом положении, и решение повторяется до момента прохождения этим «агентом» всех заданных ему целей. Согласно рекурсии, эти цели исключаются из перечня (списка) целей, выбирается следующий по очередности «агент», для которого задача решается аналогично. Вычислительный алгоритм можно отнести к классу локально оптимальных (или «жадных») алгоритмов, что связано с возможностью попадания решения в локальный минимум.

В предлагаемой работе приведен алгоритм, который, в отличие от алгоритмов Дейкстры и Беллмана – Форда, позволяющих также решать задачу о кратчайшем пути, обеспечивает одно-временную оценку кратчайших путей между любыми двумя узлами сети. Этот алгоритм позволяет производить оценку переменных состояния на взвешенных графах как с положительными, так и отрицательными весами ребер, обеспечивая возможность введения корректирующих действий по результатам вычислений с учетом сходимости решения к оптимуму. В известном алгоритме Беллмана – Форда эти операции не выполняются, а применение динамического программирования для оценки кратчайших путей является эффективным только при решении задач невысокой размерности. При этом, согласно предложенному алгоритму, все подзадачи решаются в инверсном времени с сохранением данных вычислений на каждом шаге.

Формальное описание. Предположим, что требуется найти кратчайшие пути между любыми двумя узлами взвешенного графа. При этом дуги графа с указанием расстояний между узлами могут принимать как положительные, так и отрицательные значения (при отсутствии циклов с отрицательной длиной). Теоретически отсутствующей дуге в графе приписывается вес, равный бесконечности. Однако на практике ограничиваются весом, равным конечному (достаточно большому) числу, которое могло быть больше расстояния любого пути в этом графе.

Поясним сначала на простом графе содержательную составляющую метода. Пусть имеются три узла: i, j и k , а также заданы расстояния между ними (рис. 1). При выполнении условия $d_{ij} > d_{ik} + d_{kj}$ целесообразно произвести замену пути $i \rightarrow j$ по схеме $i \rightarrow k \rightarrow j$. Эта замена обозначает *применение треугольного оператора*. Ее следует выполнять в процессе реализации алгоритма.

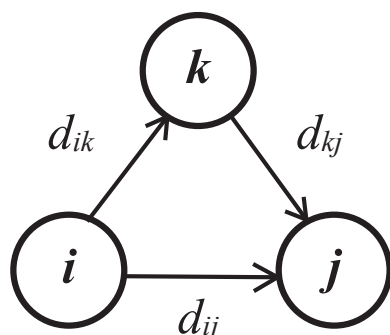


Рис. 1. Определение треугольного оператора

Алгоритм требует выполнения приведенной далее последовательности действий.

Начальный шаг. Формируем исходную матрицу расстояний D_0 и матрицу последовательности прохождения узлов S_0 . Помечаем знаком «—» диагональные элементы обеих матриц (табл. 1 и 2), поскольку эти элементы в вычислительном алгоритме не используются. Примем $k = 1$ и перейдем к основному шагу.

Таблица 1

Начальная матрица расстояний

	1	2	...	j	...	n
1	—	d_{12}	...	d_{1j}	...	d_{1n}
2	d_{21}	—	...	d_{2j}	...	d_{2n}
□	...	...	...	...	...	...
i	d_{i1}	d_{i2}		d_{ij}	...	d_{in}
□	...	...	...	...	...	...
n	d_{n1}	d_{n2}	...	d_{nj}	...	—

Таблица 2

Матрица последовательности узлов

	1	2	...	j	...	n
1	—	2	...	J	...	n
2	1	—	...	j	...	n
□	...	...	...	...	...	...
i	1	2		J	...	n
□	...	...	...	...	...	...
n	1	2	...	j	...	—

Основной шаг k . Зададим для выбора элемента матрицы строку k и столбец k как ведущие компоненты вычислительного алгоритма. Применим треугольный оператор к элементам d_{ij} матрицы D_{k-1} .

Если соблюдаются условия, определенные неравенством

$$d_{ik} + d_{kj} < d_{ij} (i \neq k, j \neq k \text{ и } i \neq j),$$

то выполняем следующие действия:

- формируем матрицу Dk , заменяя в матрице $Dk-1$ элемент d_{ij} суммой $d_{ik} + d_{kj}$;
- заменяем в матрице $Sk-1$ элементы $s_{i,j}$ на k и образуем Sk . Затем полагаем $k = k + 1$ и повторяем операцию, предусмотренную шагом k .

Ведущими строкой и столбцом на шаге $k = n$ полагаем последнюю строку и последний столбец матрицы D_{n-1} . Аналогично строим матрицы D_n и S_n . Далее никаких действий не производим, что соответствует завершению вычислений.

Пример. Определим для сети, приведенной на рис. 2, кратчайшие пути между произвольно выбранными двумя узлами.

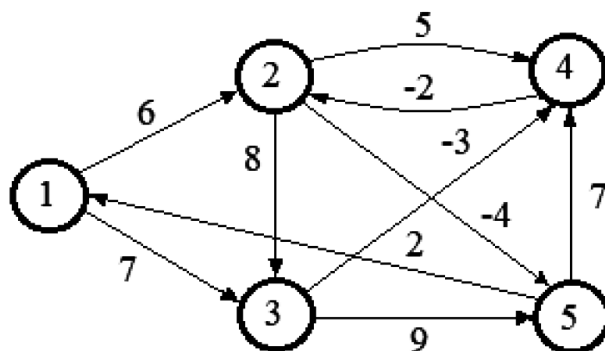


Рис. 2. Исходный взвешенный граф

Рассмотрим подробно работу алгоритма в пошаговом режиме с использованием фрагментов программы в кодах MATLAB (MatLab 2017b (9.3.0.713579), License Number: 123456).

Шаг 0. Строятся начальные матрицы расстояний (весов) D_0 (табл. 3) и последовательности узлов S_0 (табл. 4) непосредственно по заданной структуре сети.

Таблица 3

Начальная матрица весов ребер взвешенного графа

	1	2	3	4	5
1	0	6	7	100	100
2	100	0	8	5	-4
3	100	100	0	-3	9
4	100	-2	100	0	100
5	2	100	100	7	0

Таблица 4

Начальная матрица последовательности узлов S_0

	1	2	3	4	5
1	0	2	3	4	5
2	1	0	3	4	5
3	1	2	0	4	5
4	1	2	3	0	5
5	1	2	3	4	0

Для формирования матрицы S_0 инициализируются вершины графа согласно алгоритму.

Приведем фрагмент программы инициализации вершин графа:

```
for J=1:n
    if J==I
        S(I,J)=0;
    else
        S(I,J)=J;
    end
end
```

Шаг 1. На этом шаге $k = 1$. Это означает, что ведущими строкой и столбцом станут первая строка и первый столбец, выделенные на матрице D_0 светло-серым фоном. Рамкой и более темным фоном выделены следующие элементы пятой строки, значения которых можно улучшить с помощью *треугольного оператора*:

- элемент d_{52} , так как $d_{52} > d_{51} + d_{12}$;
- элемент d_{53} , так как $d_{53} > d_{51} + d_{13}$.

На первом шаге вычислений в цикле формируем матрицы D_1 и S_1 на основе матриц D_0 и S_0 в указанной последовательности:

1) заменим значение $d_{52} = 100$ (из матрицы D_0) суммой значений $d_{51} + d_{12} = 2 + 6 = 8$ и установим новое значение элемента $s_{52} = 1$ с шагом $k = 1$;

2) заменим значение $d_{53} = 100$ суммой значений $d_{51} + d_{13} = 2 + 7 = 9$, а в матрице S_1 установим новое значение элемента s_{53} , равное единице. В результате матрицы D_1 и S_1 примут вид, приведенный в табл. 5 и 6.

Таблица 5

Матрица D_1

	1	2	3	4	5
1	0	6	7	100	100
2	100	0	8	5	-4
3	100	100	0	-3	9
4	100	-2	100	0	100
5	2	8	9	7	0

Таблица 6

Матрица S_1

	1	2	3	4	5
1	0	2	3	4	5
2	1	0	3	4	5
3	1	2	0	4	5
4	1	2	3	0	5
5	1	1	1	4	0

Шаг 2. На этом шаге $k = 2$. Ведущими строкой и столбцом станут вторая строка и второй столбец, выделенные на матрице D_1 светло-серым фоном. Рамкой и более темным фоном выделены следующие элементы, значения которых также можно улучшить с помощью *треугольного оператора*:

- в первой строке — элементы d_{14} и d_{15} , так как $d_{14} > d_{12} + d_{24}$ и $d_{15} > d_{12} + d_{25}$;
- в четвертой строке — элементы d_{41} , d_{43} и d_{45} , так как $d_{41} > d_{42} + d_{21}$, $d_{43} > d_{42} + d_{23}$ и $d_{45} > d_{42} + d_{25}$.

На втором шаге вычислений в цикле формируем значения матриц D_2 и S_2 на основе матриц D_1 и S_1 в следующем порядке:

1) заменим значения $d_{14} = 100$ и $d_{15} = 100$ суммами значений, соответственно, $d_{12} + d_{24} = 6 + 5 = 11$, $d_{12} + d_{25} = 6 + (-4) = 2$ и установим значения элементов $s_{14} = s_{15} = 2$;

2) заменим значения $d_{41} = 100$, $d_{43} = 100$, $d_{45} = 100$ суммами значений:

$$d_{42} + d_{21} = (-2) + 100 = 98, d_{42} + d_{23} = (-2) + 8 = 6, d_{42} + d_{25} = (-2) + (-4) = -6$$

и установим значения элементов $s_{41} = s_{43} = s_{45} = 2$.

В итоге матрицы D_2 и S_2 примут вид, приведенный в табл. 7 и 8.

Таблица 7

Матрица D_2

	1	2	3	4	5
1	0	6	7	11	2
2	100	0	8	5	-4
3	100	100	0	-3	9
4	98	-2	6	0	-6
5	2	8	9	7	0

Таблица 8

Матрица S_2

	1	2	3	4	5
1	0	2	3	2	2
2	1	0	3	4	5
3	1	2	0	4	5
4	2	2	2	0	2
5	1	1	1	4	0

Шаг 3. На этом шаге $k = 3$. Ведущими строкой и столбцом становятся третья строка и третий столбец, выделенные на матрице D_2 светло-серым фоном. Рамкой и более темным фоном выделены элементы, значения которых можно улучшить с помощью треугольного оператора:

- в первой строке — элемент d_{14} , так как $d_{14} > d_{13} + d_{34}$;
- в пятой строке — элемент d_{54} , так как $d_{54} > d_{53} + d_{34}$.

На третьем шаге вычислений в цикле формируем значения матриц D_3 и S_3 на основе матриц D_2 и S_2 путем замены значений $d_{14} = 11$, $d_{54} = 7$ суммами значений: $d_{13} + d_{34} = 7 + (-3) = 4$, $d_{53} + d_{34} = 9 + (-3) = 6$ и устанавливаем значения элементов $s_{14} = s_{54} = 3$.

Таким образом, матрицы D_3 и S_3 примут вид, приведенный в табл. 9 и 10.

Таблица 9

Матрица D_3

	1	2	3	4	5
1	0	6	7	4	2
2	100	0	8	5	-4
3	100	100	0	-3	9
4	98	-2	6	0	-6
5	2	8	9	6	0

Таблица 10

Матрица S_3

	1	2	3	4	5
1	0	2	3	3	2
2	1	0	3	4	5
3	1	2	0	4	5
4	1	2	2	0	2
5	1	1	1	3	0

Шаг 4. На этом шаге $k = 4$. Ведущими строкой и столбцом станут четвертая строка и четвертый столбец, которые выделены на матрице D_2 светло-серым фоном. Рамкой и более темным фоном выделены элементы, значения которых можно улучшить с помощью *треугольного оператора*:

- в первой строке — элементы d_{12}, d_{15} , так как $d_{12} > d_{14} + d_{42}$ и $d_{15} > d_{14} + d_{45}$;
- в третьей строке — элементы d_{31}, d_{32}, d_{35} , так как $d_{31} > d_{34} + d_{41}$, $d_{32} > d_{34} + d_{42}$ и $d_{35} > d_{34} + d_{45}$;
- в пятой строке — элемент d_{52} , так как $d_{52} > d_{54} + d_{42}$.

На четвертом шаге вычислений формируем в цикле значения матриц D_4 и S_4 на основе матриц D_3 и S_3 в следующем порядке:

1) заменим значения $d_{12} = 6$ и $d_{15} = 2$ суммами значений, соответственно, $d_{14} + d_{42} = 4 + (-2) = -2$, $d_{14} + d_{45} = 4 + (-6) = -2$ и устанавливаем значения элементов $s_{12} = s_{15} = 4$;

2) заменим значения $d_{31} = 100$, $d_{32} = 100$, $d_{35} = 9$ суммами значений, соответственно:

$$d_{34} + d_{41} = (-3) + 98 = 95, d_{34} + d_{42} = (-3) + (-2) = -5, d_{34} + d_{45} = (-3) + (-6) = -9$$

и установим значения элементов $s_{31} = s_{32} = s_{35} = 4$;

3) заменим значение $d_{52} = 8$ суммой значений $d_{54} + d_{42} = 6 + (-2) = 4$ и установим значение элемента $s_{52} = 4$. Таким образом, матрицы D_4 и S_4 примут вид, приведенный в табл. 11 и 12.

Таблица 11

Матрица D_4

	1	2	3	4	5
1	0	2	7	4	-2
2	100	0	8	5	-4
3	95	-5	0	-3	-9
4	98	-2	6	0	-6
5	2	4	9	6	0

Таблица 12

Матрица S_4

	1	2	3	4	5
1	0	4	3	3	4
2	1	0	3	4	5
3	4	4	0	4	4
4	2	2	2	0	2
5	1	4	1	3	0

Шаг 5. На этом шаге $k = 5$. Ведущими строкой и столбцом станут пятая строка и пятой столбец, выделенные более светлым фоном на матрице D_4 . Рамкой и более темным фоном выделены следующие элементы, значения которых можно улучшить с помощью *треугольного оператора*:

– во второй строке — элементы d_{21}, d_{23}, d_{24} , так как $d_{21} > d_{25} + d_{51}$, $d_{23} > d_{25} + d_{53}$ и $d_{24} > d_{25} + d_{54}$;

– в третьей строке — элемент d_{31} , так как $d_{31} > d_{35} + d_{51}$;

– в четвертой строке — элементы d_{41}, d_{43} , так как $d_{41} > d_{45} + d_{51}$ и $d_{43} > d_{45} + d_{53}$.

На пятом шаге вычислений в цикле формируем значения матриц D_5 и S_5 на основе матриц D_4 и S_4 в следующем порядке:

1) заменим значение $d_{21} = 100$, $d_{23} = 8$ и $d_{24} = 5$ суммами значений соответственно:

$$d_{25} + d_{51} = (-4) + 2 = -2, d_{25} + d_{53} = (-4) + 9 = 5, d_{25} + d_{54} = (-4) + 6 = 2$$

и установим значение элемента $s_{21} = s_{23} = s_{24} = 5$;

2) заменим значение $d_{31} = 95$ суммой значений $d_{35} + d_{51} = (-9) + 2 = -7$ и установим значение элементов $s_{31} = 5$;

3) заменим значения $d_{41} = 98$, $d_{43} = 6$ суммами значений, соответственно, $d_{45} + d_{51} = (-6) + 2 = -4$, $d_{45} + d_{53} = (-6) + 9 = 3$ и установим значения элементов $s_{41} = 5$, $s_{43} = 5$.

В итоге матрицы D_5 и S_5 примут вид, приведенный в табл. 13 и 14.

Таблица 13

Матрица D_5

	1	2	3	4	5
1	0	2	7	4	-2
2	-2	0	5	2	-4
3	-7	-5	0	-3	-9
4	-4	-2	3	0	-6
5	2	4	9	6	0

Таблица 14

Матрица S_5

	1	2	3	4	5
1	0	4	3	3	4
2	5	0	5	5	5
3	5	4	0	4	4
4	5	2	5	0	2
5	1	4	1	3	0

Фрагмент программы применения треугольного оператора, составленной в кодах MATLAB. Итерации с применением треугольного оператора:

```
while k <= n
for I = 1:n
for J = 1:n
if J ~ I && D(I, J) > D(I, k) + D(k, J)
D(I, J) = D(I, k) + D(k, J);
S(I, J) = k;
end
end
end
end
```

На пятой итерации работа алгоритма завершается.

Результаты (Results)

Конечные матрицы D_5 и S_5 (см. табл. 13 и 14) содержат всю необходимую информацию для определения кратчайших путей между любыми двумя узлами. Например, на графе кратчайшее расстояние между узлами 1 и 5 равно -2.

Для получения последовательности прохождения узлов на выбранных маршрутах следует иметь в виду, что сегмент маршрута (i, j) должен состоять из ребра (i, j) только тогда, когда $s_{ij} = j$. В иных случаях узлы i и j связаны, по крайней мере, через один промежуточный узел. Покажем это на примере.

Найдем последовательность узлов для кратчайшего пути из первой в пятую вершину, т. е. маршрута $1 \rightarrow 5$ с расстоянием $d_{15} = -2$ (см. табл. 13). Из табл. 14 видно, что $s_{1,5} = 4$ ($s_{1,5} \neq 5$). Следовательно, узлы 1 и 5 не связываются одним ребром. Находим один из промежуточных узлов. Из данных табл. 14 следует, что второй узел может быть первым промежуточным узлом, так как для этого узла $s_{2,5} = 5$. Таким образом, кратчайший маршрут между узлами 1 и 5 можно представить последовательностью

$$1 \rightarrow 2 \rightarrow 5.$$

Проверим, будет ли этот маршрут окончательным. Узлы 1 и 2 не связаны одним ребром, так как $s_{1,2} \neq 2$. Следовательно, маршрут $1 \rightarrow 2 \rightarrow 5$ не будет окончательным. Далее находим второй

из промежуточных узлов. Как видно из табл. 14, вторым промежуточным узлом может служить четвертый узел, так как для него $s_{42} = 2$. Таким образом, следующий кратчайший маршрут между узлами 1 и 5 будет иметь вид

$$1 \rightarrow 4 \rightarrow 2 \rightarrow 5.$$

Проверим, будет ли и этот маршрут окончательным. Узлы 1 и 4 также не связаны одним ребром, так как $s_{14} \neq 4$. Следовательно, маршрут $1 \rightarrow 4 \rightarrow 2 \rightarrow 5$ не будет окончательным. Далее находим третий промежуточный узел. Как видно из табл. 14, третьим промежуточным узлом может оказаться третий узел, так как для него $s_{34} = 4$. Таким образом, последующий кратчайший маршрут между узлами 1 и 5 будет иметь вид

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5.$$

Выполним проверку. Узлы 1 и 3 связаны одним ребром, так как $s_{13} = 3$. Следовательно, кратчайший путь с длиной $d_{15} = -2$, соответствующий маршруту

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5,$$

будет единственным и окончательным (рис. 3).

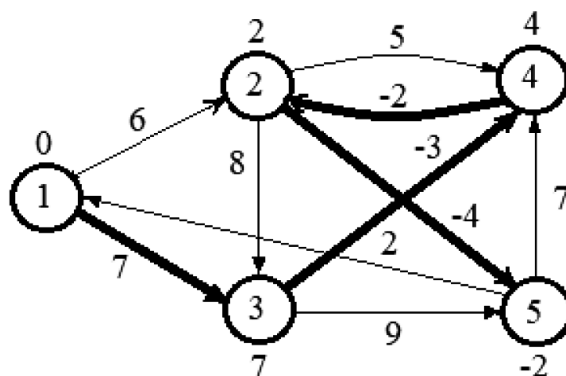


Рис. 3. Реконструированный кратчайший маршрут между узлами 1 и 5

Обсуждение (Discussion)

Из пошагового выполнения вычислений по представленному алгоритму следует, что для корректной оценки расстояний кратчайших путей в графе с отрицательными весами ребер требуется всего n итераций. При наличии в графе циклов с отрицательным весом алгоритм Флойда – Уоршелла применять некорректно. В этом случае безошибочно сработает предложенный алгоритм, поскольку после обработки графа в диагонали матрицы кратчайших путей возникнут отрицательные числа. Появление отрицательных чисел свидетельствует о том, что кратчайшее расстояние от вершины в данном цикле до нее самой меньше нуля. Это означает, что было совершено прохождение по отрицательному циклу.

Выводы (Summary)

1. Полученные результаты оптимальной маршрутизации по алгоритму на рассмотренном примере сетевого графа (см. рис. 3) полностью совпадают с результатами поиска кратчайшего маршрута, полученными в [3] на том же примере с использованием модифицированного алгоритма Беллмана – Форда.

2. На основе алгоритма Флойда – Уоршелла создана универсальная компьютерная модель, предназначенная для оценки и формирования кратчайших маршрутов между выбранной парой транзитных узлов в транспортных и телекоммуникационных сетях и, тем самым, способная обеспечить такой выбор маршрутов движения судов, наземного и воздушного транспорта, а также любых других подвижных объектов, отличающихся эффективностью и экономичностью.

3. Предложена рекурсивная процедура для выбора последовательности ребер, служащих для формирования кратчайшего пути. Вычисления выполнены с использованием инструментов OptimizationToolbox компьютерной среды MATLAB.

4. Корректность работы модели и предложенного алгоритма подтверждена на конкретном примере автоматизации расчета кратчайших расстояний между любыми парами вершин на графе транспортной сети со сложной конфигурацией.

5. Достоинства предложенных решений состоят в простоте и устойчивости вычислительного процесса при программной реализации алгоритмов средствами вычислительной среды MATLAB.

6. Областью применения алгоритма является оптимизация транспортных, телекоммуникационных и сетей, базирующихся на механизмах цифровой экономики в статических и динамических режимах.

СПИСОК ЛИТЕРАТУРЫ

1. *Dijkstra E.W.* A note on two problems in connexion with graphs / E.W. Dijkstra // *Numerische mathematik*. — 1959. — Vol. 1. — Is. 1. — Pp. 269–271. DOI: 10.1007/BF01386390.

2. *Bellman R.* On a Routing Problem / R. Bellman // *Quarterly of Applied Mathematics*. — 1958. — Vol. 16. — № 1. — С. 87–90. DOI: 10.1090/qam/102435.

3. *Чертков А. А.* Автоматизация выбора кратчайших маршрутов судов на основе модифицированного алгоритма Беллмана – Форда / А. А. Чертков // *Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова*. — 2017. — Т. 9. — № 5. — С. 1113–1122. DOI: 10.21821/2309-5180-2017-9-5-1113-1122.

4. *Warshall S.* A theorem on Boolean matrices / S. Warshall // *Journal of the ACM*. — 1962. — No 1. — Pp. 11–12.

5. *Floyd R.W.* Algorithm 97: Shortest Path / R.W. Floyd // *Communications of the ACM*. — 1962. — Vol. 5. — Is. 6. — Pp. 345. DOI: 10.1145/367766.368168.

6. *Abraham I.* Hub-Based Labeling Algorithm for Shortest Paths on Road Networks / I. Abraham, D. Delling, A.V. Goldberg, R.F. Werneck // *International Symposium on Experimental Algorithms*. — Springer, Berlin, Heidelberg, 2011. — Pp. 230–241. DOI: 10.1007/978-3-642-20662-7_20.

7. *Kröger M.* Shortest multiple disconnected path for the analysis of entanglements in two-and three-dimensional polymeric systems // *Computer physics communications*. — 2005. — Vol. 168. — Is. 3. — Pp. 209–232. DOI: 10.1016/j.cpc.2005.01.020.

8. *Сазонов А. Е.* Прогнозирование траектории движения судна при помощи нейронной сети / А. Е. Сазонов, В. В. Дерябин // *Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова*. — 2013. — № 3 (22). — С. 6–13.

9. *Fonseca C. M.* Genetic algorithms for multiobjective optimization: formulation, discussion and generalization / C.M. Fonseca, P.J. Fleming // *Genetic Algorithms: Proceedings of the Fifth International Conference* (S. Forrest, ed.). — San Mateo, CA: Morgan Kaufmann, 1993. — Pp. 416–423.

10. *Castillo O.* Multiple Objective Genetic Algorithms for Path-planning Optimization in Autonomous Mobile Robots / O. Castillo, L. Trujillo, P. Melin // *Soft Computing-A Fusion of Foundations, Methodologies and Applications*. — 2007. — Vol. 11. — Is. 3. — Pp. 269–279. DOI: 10.1007/s00500-006-0068-4.

11. *Чертков А. А.* Итерационный алгоритм выбора оптимальной стратегии группового взаимодействия подвижных объектов / А. А. Чертков // *Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова*. — 2015. — № 4 (32). — С. 207–215.

12. *Сахаров В. В.* Алгоритм оптимального планирования группового взаимодействия роботов / В. В. Сахаров, А. А. Чертков, Д. С. Тормашев // *Морской Вестник*. — 2014. — № 4. — С. 119–122.

13. *Dedkov V. A.* The problem of control for multirobot systems / V. A. Dedkov, M. N. Kirsanov // *Инновационные информационные технологии*. — 2013. — Т. 2. — № 2. — С. 206–213.

14. *Kirsanov A.* Path planning for the autonomous underwater vehicle / A. Kirsanov, S. Anavatti, T. Ray // *International Conference on Swarm, Evolutionary, and Memetic Computing*. — Springer, Cham, 2013. — Pp. 476–486. DOI: 10.1007/978-3-319-03756-1_43.

REFERENCES

1. Dijkstra, Edsger W. "A note on two problems in connexion with graphs." *Numerische mathematik* 1.1 (1959): 269–271. DOI: 10.1007/BF01386390.
2. Bellman, Richard. "On a Routing Problem." *Quarterly of Applied Mathematics* 16.1 (1958): 87–90. DOI: 10.1090/qam/102435.
3. Chertkov, Alexander A. "Automation selection shortcuts routes of ships on the basis of modified Bellman-Ford Algorithm." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S. O. Makarova* 9.5 (2017): 1113–1122. DOI: 10.21821/2309-5180-2017-9-5-1113-1122.
4. Warshall, Stephen. "A theorem on Boolean matrices." *Journal of the ACM* 1 (1962): 11–12.
5. Floyd, Robert W. "Algorithm 97: shortest path." *Communications of the ACM* 5.6 (1962): 345. DOI: 10.1145/367766.368168.
6. Abraham, Ittai, Daniel Delling, Andrew V. Goldberg, and Renato F. Werneck. "A Hub-Based Labeling Algorithm for Shortest Paths on Road Networks." *International Symposium on Experimental Algorithms*. Springer, Berlin, Heidelberg, 2011. 230–241. DOI: 10.1007/978-3-642-20662-7_20.
7. Kröger, Martin. "Shortest multiple disconnected path for the analysis of entanglements in two-and three-dimensional polymeric systems." *Computer physics communications* 168.3 (2005): 209–232. DOI: 10.1016/j.cpc.2005.01.020.
8. Sazonov, A. E., and V. V. Derjabin. "Forecasting to paths of the motion ship with the help of neyronnoy network." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 3(22) (2013): 6–13.
9. Fonseca, Carlos M., and Peter J. Fleming. "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization." *Genetic Algorithms: Proceedings of the Fifth International Conference (S. Forrest, ed.)*. San Mateo, CA: Morgan Kaufmann, 1993. 416–423.
10. Castillo, Oscar, Leonardo Trujillo, and Patricia Melin. "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots." *Soft Computing-A Fusion of Foundations, Methodologies and Applications* 11.3 (2007): 269–279. DOI: 10.1007/s00500-006-0068-4.
11. Chertkov, A. A. "An iterative algorithm for choosing the optimal strategy of group interaction for moving objects." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 4(32) (2015): 207–215.
12. Saharov, V. V., A. A. Chertkov, and D. S. Tormašev. "Optimal planning group communication Algorithm of robots." *Morskoj Vestnik* 4 (2014): 119–122.
13. Dedkov, V. A., and M. N. Kirsanov. "The problem of control for multirobot systems." *Innovative information technology* 2.2 (2013): 206–213.
14. Kirsanov, Andrey, Sreenatha G. Anavatti, and Tapabrata Ray. "Path planning for the autonomous underwater vehicle." *International Conference on Swarm, Evolutionary, and Memetic Computing*. Springer, Cham, 2013: 476–486. DOI: 10.1007/978-3-319-03756-1_43.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Сахаров Владимир Васильевич —
доктор технических наук, профессор
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
198035, Российская Федерация,
Санкт-Петербург, ул. Двинская 5/7
e-mail: _saharov_@rambler.ru,
SaharovVV@gumrf.ru

Чертков Александр Александрович —
доктор технических наук, доцент
ФГБОУ ВО «ГУМРФ имени
адмирала С.О. Макарова»
198035, Российская Федерация,
Санкт-Петербург, ул. Двинская 5/7
e-mail: chertkov51@mail.ru,
kaf_electricautomatic@gumrf.ru

INFORMATION ABOUT THE AUTHORS

Sakharov, Vladimir V. —
Dr. of Technical Sciences, professor
Admiral Makarov State University
of Maritime and Inland Shipping
5/7 Dvinskaya Str., St.Petersburg 198035,
Russian Federation
e-mail: _saharov_@rambler.ru,
SaharovVV@gumrf.ru

Chertkov, Alexandr A. —
Dr. of Technical Sciences, associate professor
Admiral Makarov State University
of Maritime and Inland Shipping
5/7 Dvinskaya Str., St. Petersburg, 198035,
Russian Federation
e-mail: chertkov51@mail.ru,
kaf_electricautomatic@gumrf.ru

Очина Людмила Борисовна —
кандидат технических наук, доцент
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
198035, Российская Федерация,
Санкт-Петербург, ул. Двинская, 5/7
e-mail: kaf_electricautomatic@gumrf.ru

Ochina, Ludmila B. —
PhD, associate professor
Admiral Makarov State University
of Maritime and Inland Shipping
5/7 Dvinskaya Str., St. Petersburg, 198035,
Russian Federation
e-mail: kaf_electricautomatic@gumrf.ru

*Статья поступила в редакцию 3 февраля 2019 г.
Received: February 3, 2019.*