

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И АВТОМАТИЗАЦИЯ НА ТРАНСПОРТЕ

DOI: 10.21821/2309-5180-2017-9-5-1113-1122

AUTOMATION SELECTION SHORTCUTS ROUTES OF SHIPS ON THE BASIS OF MODIFIED BELLMAN-FORD ALGORITHM

A. A. Chertkov

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

Task automation of operations determine shortest routes for groups of vessels which have received job achieve conditional purposes with known coordinates, located in a confined space. When the operational management of the release of the vessels in a given area is important indicator of time, minimization of which is connected with the search for a route that has the shortest length of all possible routes. A key issue in the problem of automation generating optimal routes is the choice of mathematical apparatus not only to calculate the lengths of shortest routes, but also to restore shortcuts themselves. For this purpose in the work detail recursive dynamic optimization method based on modified Bellman-Ford algorithm to avoid large dimension and computational complexity in dealing with this task. The proposed method, in contrast to other methods of dynamic optimization, allows you to automate the determination of shortest path in a network model with complex topology if there it arcs with negative weights.

The practical implementation of the Bellman-Ford algorithm modified is shown in the example, calculating network model with complex topology using an iterative procedure for program prepared in MATLAB codes. Implemented on the basis of the computer model is compact, simple and, unlike existing models, allows to eliminate the restrictions associated with the presence of negative weights and cycles on the network model, automate calculations in shortcuts extensive area of fishing crafts sites with the use of computer technologies by means of MATLAB, and specified quality criteria. For example, calculate the confirmed the correctness of the obtained research results the author. The proposed algorithm and recursive procedure, as well as the conclusions of their analysis may be applied to the movement of sea, air and land transport.

Keywords: route, purpose, algorithm, path, optimization, procedure, control center, transportation.

For citation:

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.

УДК 681.5

АВТОМАТИЗАЦИЯ ВЫБОРА КРАТЧАЙШИХ МАРШРУТОВ СУДОВ НА ОСНОВЕ МОДИФИЦИРОВАННОГО АЛГОРИТМА БЕЛЛМАНА-ФОРДА

А. А. Чертков

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

Решается задача автоматизации операций определения кратчайших маршрутов для группы судов, получивших задание достижения условных целей с известными координатами, расположенными в ограниченном пространстве. При оперативном управлении процессом выхода судов в заданный район важен показатель времени, минимизация которого связана с поиском маршрута, имеющего самую короткую длину из всех возможных маршрутов. Ключевым вопросом в проблеме автоматизации формирования оптимальных маршрутов является выбор математического аппарата не только для расчета длин кратчайших маршрутов, но и для восстановления самих кратчайших путей. С этой целью в работе подробно рассматривается рекурсивный метод динамической оптимизации на основе модифицированного алгоритма Беллмана-Форда, позволяющего избежать большой размерности и вычислительной сложности в решении данной задачи. Предложенный ме-

тод, в отличие от других методов динамической оптимизации, позволяет автоматизировать определение кратчайшего пути в сетевой модели со сложной топологией при наличии в ней дуг с отрицательными весами.

Практическая реализация модифицированного алгоритма Беллмана–Форда показана на примере расчета сетевой модели со сложной топологией с применением итерационной процедуры по программе, составленной в кодах MATLAB. Реализованная на ее основе компьютерная модель компактна, проста и, в отличие от существующих моделей, позволяет устранить ограничения, связанные с наличием отрицательных весов и циклов на сетевой модели, автоматизировать расчеты кратчайших путей в разветвленном районе мест рыболовецких промыслов с использованием компьютерных технологий, реализованных средствами MATLAB, и заданного критерия качества. На примере расчета подтверждается корректность полученных результатов исследований автора. Предложенные алгоритм и рекурсивная процедура, а также выводы из их анализа могут найти применение в организации движения морского, воздушного и наземного транспорта.

Ключевые слова: маршрут, цель, алгоритм, путь, оптимизация, процедура, центр управления, транспорт.

Для цитирования:

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.

Введение

Выбор кратчайшего маршрута для судна, летательного аппарата или наземного транспорта определяет его эффективность и экономичность. Такие задачи для отдельных судов, или в терминологии робототехники — агентов, решаются методами исследования операций на графах, в частности на основе известных алгоритмов Дейкстры, подробно рассмотренных автором в [1], Беллмана–Форда, Флойда и др. [2]. Не менее эффективны методы искусственного интеллекта на основе нейронных сетей [3], позволяющие рассчитывать меняющиеся со временем условия [4], [5] и генетические алгоритмы [6], [7]. Более сложные задачи для группы агентов, в том числе задачи о коллективном покрытии операционных пространств, решаются с использованием вероятностных методов и координации действий агентов на заранее выделенных подпространствах [8] – [13]. Область применения решений подобных оптимизационных задач весьма широка — от выбора маршрутов судов [14] и мобильных роботов в системе обслуживания и военном деле [15] до трехмерных задач планирования путей подводных аппаратов [16], [17].

В рамках теории исследования операций рассматривается большое количество практических задач, которые можно сформулировать как задачи нахождения кратчайшего пути и решить с помощью сетевых моделей, в частности:

- формирования безопасных маршрутов в зонах разделения судопотоков различных направлений, сходящихся в небольшом месте или в узловой точке, которой могут служить подходы к портам, места приема лоцманов, места установки подходов буйев или плавучих маяков, входы в фарватеры, каналы, эстуарии и т. д.;
- проектирования трубопровода, соединяющего буровые скважины морского базирования с находящейся на берегу коллекторной станцией;
- соединения всех узлов сети (кабельной, дорожной, трубопроводной, коммуникационной) с помощью путей наименьшей длины;
- проектирования самых надежных маршрутов движения в сети дорог (водных путей);
- формирование маршрутов судов, оптимальных по времени перехода;
- замены оборудования при обновлении флота за определенный период;
- определения максимальной пропускной способности трубопровода;
- определения схемы транспортировки нефти и нефтепродуктов от пунктов добычи к нефтеперерабатывающим заводам и потребителям с минимальной стоимостью транспортировки;
- составления временного графика строительных работ (определение дат начала и завершения отдельных этапов работы);

– календарного планирования, использующие разностные ограничения.

Решение приведенных задач требует применения различных сетевых оптимизационных алгоритмов.

Методы и материалы

Из множества задач нахождения кратчайших путей рассмотрим задачу планирования маршрутов движения судов в разветвленной системе целей, моделирующих, например, реальную ситуацию с распределением областей промыслов в рыболовецкой компании между отдельными судами. Предполагается, что места расположения целей (координаты мест промыслов, богатых добычей) разведаны и известны центральному управлению. Ставится задача выбора таких маршрутов, для которых время в пути было бы минимальным. Время работы определяется по времени T_{\max} достижения самой удаленной цели. Скорость движения агентов предполагается одинаковой, временем «обработки» целей пренебрегаем, поэтому, если скорость агента принять за единицу, время выполнения задачи можно условно считать равным длине всего маршрута.

Задача о минимальном маршруте решается последовательно для каждого агента. Выбирается произвольный агент, и определяется наиболее близкая к нему цель. Затем агент размещается в новом положении, и задача опять повторяется до тех пор, пока агент не пройдет заданные ему N целей. Затем эти цели вычеркиваются из списка целей, и задача решается для другого агента. Очевидно, этот алгоритм относится к локально оптимальным (или «жадным») алгоритмам и имеет тенденцию к попаданию в локальный минимум. Если не прибегать к нейросетевым или генетическим алгоритмам, то единственная возможность избежать попадания в локальный минимум — это осуществить полный перебор всех вариантов последовательностей агентов.

В данной работе представлен алгоритм Беллмана–Форда, позволяющий решить задачу о кратчайшем пути из фиксированного истока в общем случае, когда вес любого ребра может быть отрицательным. Этот алгоритм отличается своей простотой. К его достоинствам также относится то, что он определяет, содержится ли в графе цикл с отрицательным весом, достижимый из истока.

Алгоритм Беллмана–Форда решает ту же задачу, что и алгоритм Дейкстры, но работает с отрицательными весами ребер (дуг) графа, позволяя обнаруживать отрицательные циклы. Алгоритм носит имя двух американских учёных: Ричарда Беллмана (Richard Bellman) и Лестера Форда (Lester Ford). Форд фактически изобрёл этот алгоритм в 1956 г. при решении другой математической задачи, подзадача которой свелась к поиску кратчайшего пути в графе, и Форд дал набросок решающего эту задачу алгоритма. Беллман в 1958 г. опубликовал статью [18], посвящённую задаче нахождения кратчайшего пути, и в этой статье чётко сформулировал алгоритм в том виде, в котором он известен нам сейчас. В алгоритме Беллмана–Форда используется ослабление, в результате которого величина $d[v]$, представляющая собой оценку веса кратчайшего пути из истока s к каждой из вершин $v \in V$, уменьшается до тех пор, пока она не станет равной фактическому весу кратчайшего пути из s в v . Значение TRUE возвращается алгоритмом тогда и только тогда, когда граф не содержит циклов с отрицательным весом, достижимых из истока.

Формальное описание. Предположим, что требуется найти длины кратчайших путей от вершины-входа до каждой другой вершины графа. Для этого алгоритма дуговые расстояния могут быть как положительными, так и отрицательными, но не должно быть циклов отрицательной длины. Предположим, что если в графе отсутствует та или иная дуга, то её вес равен бесконечно большому числу. Основная идея алгоритма Беллмана–Форда состоит в том, чтобы сначала найти длины кратчайших путей, при условии, что пути содержат не более двух дуг, не более трех дуг и т. д. Кратчайший путь до вершины v , при условии, что он содержит не более h дуг, будем называть *кратчайшим путем в графе*. Согласно идее Беллмана–Форда, Р. Прим предложил алгоритм определения длин кратчайших путей, ведущих из произвольной вершины графа во все остальные его вершины (в которые есть пути из вершины-входа), состоящий в присвоении вершинам некоторых потенциалов ϕ_v , служащих оценкой веса кратчайшего пути

из вершины-входа до вершины $v_i \in V$. Признаком окончания работы алгоритма является остановка процесса изменения потенциалов.

Алгоритм, предложенный Р. Примом, состоит в выполнении следующих операций.

1. Вершине-входу присваиваем потенциал $\varphi_1 = 0$, остальным вершинам — потенциал $\varphi_i = \infty$, $i = 1 \dots n$, где n — число вершин в рассматриваемом графе. На практике вместо теоретического предела $\varphi_i = \infty$ используют численные значения φ_j , заведомо большие значений длины любого потенциального пути до любой из вершин графа. Таким образом, результатом инициализации вершин графа, исключая вершину-исток, станет массив оценок кратчайших путей L_0 .

2. На i -м шаге для каждой вершины j , смежной с вершиной i , находим $\varphi_i + l(i, j)$ и проверяем неравенство $\varphi_i + l(i, j) < \varphi_j$. Если неравенство выполняется, то $\varphi_j = \varphi_i + l(i, j)$. Новое значение потенциала φ_j вносим в массив L оценок кратчайших путей на место j -й координаты, а саму j -ю вершину заносим в массив P .

3. Проверяем массив $L_i \neq L_{i-1}$. Если неравенство выполняется, осуществляем переход к п. 2 алгоритма.

4. Конец алгоритма нахождения кратчайших путей.

Рассмотрим работу алгоритма на примере.

Пример. Пусть имеется взвешенный граф, представленный на рис. 1, в котором есть дуги с отрицательными весами.

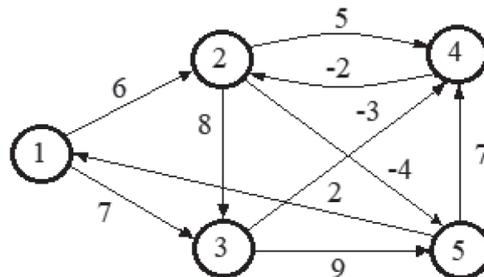


Рис. 1. Исходный взвешенный граф

Матрица весов приведена в табл. 1. Требуется найти все кратчайшие пути от вершины 1 ко всем остальным вершинам.

Таблица 1

Элементы матрицы весов ребер исходного взвешенного графа

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

Рассмотрим подробно работу алгоритма в пошаговом режиме с помощью фрагментов программы, составленной в кодах MATLAB, и табл. 2, в которой представлены результаты работы алгоритма по всем шагам вычислений.

Таблица 2

Результаты работы алгоритма по шагам вычислений

Шаг вычислений	Массив отображаемых вершин P	Оценки кратчайших путей L до вершин графа					Массив предшествующих вершин V
		1	2	3	4	5	
0	[]	0	100	100	100	100	—
1	2, 3	0	6	7	100	100	1, 1
2	4, 5	0	6	7	11	2	2, 2
3	4	0	6	7	4	2	3
4	2, 5	0	2	7	4	-2	4, 2
5	1, 4	0	2	7	4	-2	5, 5

На нулевом шаге производится инициализация согласно п. 1 алгоритма. Фрагмент программы инициализации вершин графа, составленный в кодах MATLAB:

```
L(1)=0;
for j=2:n % инициализация вершин
    L(j)=100;
end
```

Далее, на первом шаге вычислений в цикле производится поиск вершин, смежных с вершиной входа (под номером 1). Всем вершинам, смежным с вершиной-входом, присваиваются потенциалы, равные длинам (весам) дуг, соединяющих их с вершиной входа, если они существуют, в противном случае — потенциал вершины принимается равным числу, заведомо большему длины любого потенциального пути до нее, например, числу 100. Таким образом, на первом шаге, как видно из рис. 1, изменяются только потенциалы вершин 2 и 3:

- 1) потенциал вершины 2 изменяется с $\varphi_2 = 100$ до $\varphi_2 = 6$;
- 2) потенциал вершины 3 изменяется с $\varphi_3 = 100$ до $\varphi_3 = 7$.

Значения потенциалов вершин и выделенные ребра до смежных вершин графа на первом шаге показаны на рис. 2.

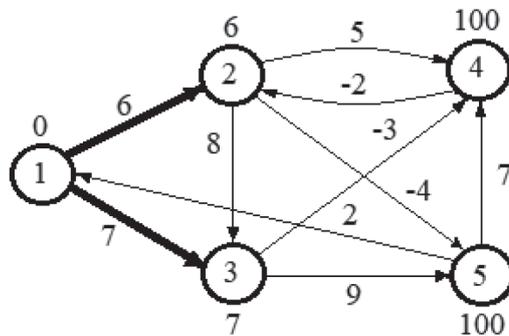


Рис. 2. Проход по ребрам графа на первом шаге

На втором шаге производится поиск вершин, смежных со второй вершиной, которая соединена с ними прямыми дугами. Всем смежным с ней вершинам, согласно п. 2 алгоритма, присваивается новое значение потенциала, если оно оказывается меньше значения потенциала на предыдущем шаге. Потенциалы не смежных с ней вершин не изменяются. Таким образом, смежные с ней вершины получают следующие значения потенциалов:

- 1) потенциал вершины 3 не изменяется, так как новое значение $\varphi_3 = \varphi_2 + l_{23} = 6 + 8 = 14$ больше значения $\varphi_3 = 9$, полученного на предыдущем шаге;
- 2) потенциал вершины 4 изменяется с $\varphi_4 = 100$ до $\varphi_4 = \varphi_2 + l_{24} = 6 + 5 = 11$;
- 3) потенциал вершины 5 изменяется с $\varphi_5 = 100$ до $\varphi_5 = \varphi_2 + l_{25} = 6 + (-4) = 2$.

Значения потенциалов вершин и выделенные ребра кратчайших путей на втором шаге показаны на рис. 3.

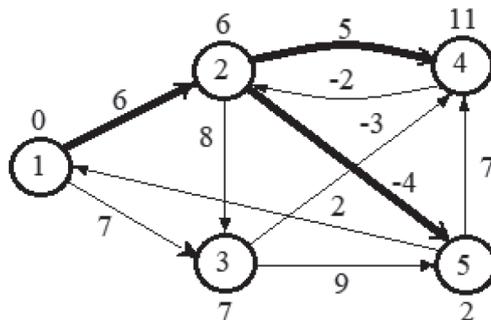


Рис. 3. Проход по ребрам графа на втором шаге

Фрагмент программы автоматизации операций взвешивания, составленной в кодах MATLAB:

```
for j=1:n
    if L(j)>LP(i)+C(i,j) %релаксация пути
        L(j)=LP(i)+C(i,j);
        V(i,j)=i; %Формирование списка предпоследних вершин в кратчайшем %пути
        p=j;
        LP(p)=L(j); %Формирование массива оценок кратчайших путей
        P(i,j)=j; %Формирование массива взвешиваемых вершин
```

На *третьем шаге*, аналогично второму шагу, производится взвешивание вершин, смежных с третьей вершиной. В результате взвешивания потенциалы несмежных с ней вершин не изменятся (см. табл. 1), а потенциалы смежных ней четвертой и пятой вершин будут следующими:

1) потенциал вершины 4 изменяется с $\varphi_4 = 11$ до $\varphi_4 = \varphi_3 + l_{34} = 7 + (-3) = 4$;

2) потенциал вершины 5 не изменяется, так как новое значение $\varphi_5 = \varphi_3 + l_{35} = 7 + 9 = 16$ больше значения $\varphi_5 = 2$, полученного на предыдущем шаге.

Значения потенциалов вершин и выделенные ребра кратчайших путей на третьем шаге показаны на рис. 4.

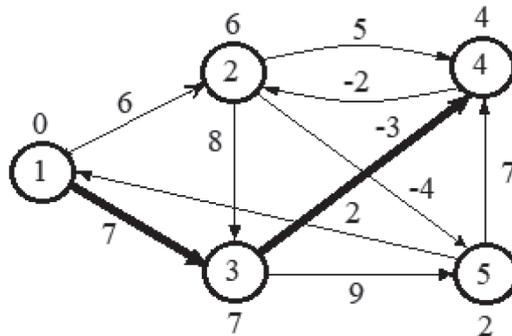


Рис. 4. Проход по ребрам графа на третьем шаге

Аналогично на *четвертом шаге* осуществляется поиск и взвешивание вершин, смежных с четвертой вершиной. В результате получаем:

1) потенциал вершины 2 изменяется с $\varphi_2 = 6$ до $\varphi_2 = \varphi_4 + l_{42} = 4 + (-2) = 2$. Поскольку изменился потенциал меньшей по номеру четвертой вершины, выполняются операции «довзвешивания» сначала вершин, смежных со второй, затем с третьей и, наконец, с четвертой вершинами;

2) потенциал вершины 3 не изменяется, так как новое значение $\varphi_3 = \varphi_2 + l_{23} = 2 + 8 = 10$ больше значения $\varphi_3 = 7$, полученного на первом шаге;

3) потенциал вершины 4 не изменяется, так как новое значение $\varphi_4 = \varphi_2 + l_{24} = 2 + 5 = 7$ больше значения $\varphi_4 = 4$, полученного на предыдущем шаге;

4) потенциал вершины 5 изменяется с $\varphi_5 = 2$ до $\varphi_5 = \varphi_2 + l_{25} = 2 + (-4) = -2$.

Значения потенциалов вершин и выделенные ребра кратчайших путей на третьем шаге показаны на рис. 5.

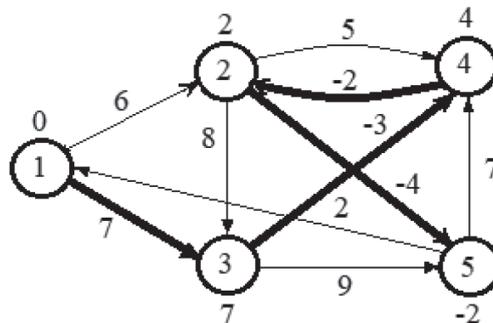


Рис. 5. Проход по ребрам графа на четвертом шаге

Фрагмент программы, обеспечивающей автоматизацию операций «довзвешивания»:

```

if j<i
  for I=j:n
    if L(I)>L(j)+C(j,I)
      L(I)=L(j)+C(j,I);
      V(I,j)=j;
      p=I;
      LP(p)=L(I);
      P(I,j)=I;
    
```

На пятом шаге вычислений не произошло изменений ни в одном из потенциалов вершин графа, что позволяет сделать вывод об окончании работы алгоритма.

Восстановим кратчайший путь от вершины 1 к вершине 5. С целью автоматизации операции восстановления кратчайших путей в алгоритм программы введен массив V , в котором для каждой вершины будет храниться ее «предок», т. е. предпоследняя вершина в кратчайшем пути, ведущим в нее. Это следует из логики работы алгоритма Беллмана–Форда, согласно которой предполагается, что если кратчайшее расстояние до одной вершины рассчитано, то можно улучшить оценку кратчайшего расстояния до другой вершины. Отсюда в момент улучшения оценки кратчайшего пути до этой вершины необходимо просто запоминать в отдельном массиве ту вершину, из которой это улучшение произошло.

Результаты

Результатами решения задачи автоматизации вычисления и восстановления кратчайших путей на графе (см. рис. 1) от вершины 1 ко всем остальным вершинам будут массивы $LP = [\varphi_1, \varphi_2, \dots, \varphi_n]$, V и P соответственно минимальных оценок кратчайших путей, предпоследних вершин в кратчайшем пути и вершин, в которых произошло улучшение оценок. Причем значения массивов V и P отражены на каждом шаге представленного алгоритма:

LP =

0	2	7	4	-2
---	---	---	---	----

V =

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

P =

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

Как видно из массива P , последней вершиной, для которой произошло улучшение оценки, является пятая вершина, для которой предпоследней вершиной будет, как видно из массива V в той же позиции, вторая вершина. Обращаемся снова к массиву P и находим в предпоследней строке вершину 2. Этой вершине предшествует, как видно из массива V на той же позиции, вершина 4. Вершине 4, отмеченной в третьей строке на четвертой позиции массива P , предшествует, как видно из массива V на той же позиции, вершина 3, которой, как видно из первых строк массивов P и V , предшествует вершина 1. Таким образом, на данном графе кратчайшие расстояния от вершины 1 ко всем остальным вершинам будут проходить по следующим маршрутам (табл. 3).

Кратчайшие расстояния и маршруты до вершин графа

Номера вершин	1	2	3	4	5
Кратчайшие расстояния	0	2	7	4	-2
Маршруты	-	1-3-4-2	1-3	1-3-4	1-3-4-2-5

Обсуждение

Как видно из пошаговой реализации алгоритма, для корректного подсчета длин всех кратчайших путей в графе требуется всего $(n - 1)$ итераций (шагов). После этого ни одна релаксация не может гарантированно завершиться улучшением оценки расстояния до какой-либо вершины. Данное утверждение справедливо в случае отсутствия отрицательного цикла. При его наличии алгоритм Беллмана–Форда сможет бесконечно делать релаксации среди вершин этого цикла и вершин, достижимых из него. Следовательно, для обнаружения достижимого цикла отрицательного веса необходимо выполнить еще одну итерацию и если на ней произойдет хотя бы одна релаксация, то граф содержит цикл отрицательного веса, в противном случае такого цикла нет. В данном примере отсутствие отрицательного цикла подтверждается результатами программной реализации на пятом шаге — результаты оценок кратчайших расстояний не меняются.

Заключение

1. Разработана универсальная компьютерная модель, позволяющая автоматизировать операции определения кратчайших расстояний для группы судов, получивших задание достижения условных целей с известными координатами, расположенных в ограниченном пространстве.

2. Предложена рекурсивная процедура определения последовательности ребер, из которых состоит рассматриваемый кратчайший путь с использованием функций инструментария Optimization Toolbox среды MATLAB.

3. Корректность и эффективность компьютерной модели модифицированного алгоритма Беллмана–Форда и ее программной реализации подтверждена конкретным примером автоматизации расчета кратчайших расстояний при наличии отрицательных ребер на графе.

4. Достоинствами предложенной процедуры автоматизации расчета кратчайших путей на основе модифицированного алгоритма Беллмана–Форда является простота программной реализации и возможность использования универсального инструментария, предоставляемого вычислительной средой MATLAB.

СПИСОК ЛИТЕРАТУРЫ

1. *Чертков А. А.* Рекурсивный метод оптимизации логистических путей средствами MATLAB / А. А. Чертков, А. А. Вардомская, А. А. Дмитриев // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2015. — № 6 (34). — С. 196–204.

2. *Касьянов В. Н.* Графы в программировании: визуализация и применение / В. Н. Касьянов, В. А. Евстигнеев. — СПб.: БХВ-Петербург, 2003. — 1104 с.

3. *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.

4. *Gao S.* Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks / S. Gao, H. Huang // Transportation Research Part C: Emerging Technologies. — 2012. — Vol. 21. — Is. 1. — Pp. 196–213. DOI: 10.1016/j.trc.2011.09.007.

5. *Fu L.* An adaptive routing algorithm for in-vehicle route guidance systems with real-time information / L. Fu // Transportation Research Part B: Methodological. — 2001. — Vol. 35. — Is. 8. — Pp. 749–765. DOI: 10.1016/S0191-2615(00)00019-9.

6. Сахаров В. В. Модели и алгоритмы оптимизации технологических процессов на объектах водного транспорта в среде MatLab / В. В. Сахаров, А. А. Кузьмин, А. А. Чертков. — СПб.: Изд-во ГУМРФ им. адм. С. О. Макарова. — 2015. — 436 с.
7. 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.
8. 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.
9. 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.
10. Каляев И. А. Модели и алгоритмы коллективного управления в группах роботов / И. А. Каляев, А. Р. Гайдук, С. Г. Капустян. — М: Физматлит, 2009. — 280 с.
11. Kalyaev I. A. UAV group control in task of order forming / I. A. Kalyaev, A. R. Gaiduk, S. G. Kapustyan // *Робототехника и техническая кибернетика*. — 2014. — № 4 (5). — С. 28–39.
12. Александров В. А. Коллективный алгоритм выделения операционных подпространств для группы роботов при решении задачи покрытия территории / В. А. Александров, А. И. Кобрин // *Известия высших учебных заведений*. — Серия: Машиностроение. — 2011. — № 9. — С. 65–69.
13. Александров В. А. Аппаратно-программный комплекс для моделирования задач группового управления мобильными роботами / В. А. Александров, А. И. Кобрин // *Вестник Московского энергетического института*. — 2011. — № 3. — С. 88–95.
14. Кирсанов М. Н. Анализ алгоритмов выбора оптимальных маршрутов группы судов / М. Н. Кирсанов // *Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова*. — 2016. — № 2 (36). — С. 183–190. DOI: 10.21821/2309-5180-2016-8-2-183-190.
15. 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.
16. 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.
17. Zeng Z. A survey on path planning for persistent autonomy of autonomous underwater vehicles / Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, A. Lammas // *Ocean Engineering*. — 2015. — Vol. 110. — Part A. — Pp. 303–313. DOI: 10.1016/j.oceaneng.2015.10.007.
18. Bellman R. On a Routing Problem / R. Bellman // *Quarterly of Applied Mathematics*. — 1958. — Vol. 16. — № 1. — С. 87–90. DOI: 10.1090/qam/102435.

REFERENCES

1. Chertkov, Alexandr Alexandrovich, Anna Alexandrovna Vardomsкая, and Alexandr Alexandrovich Dmitriev. “A Recursive method of optimization of the logistic ways by means of MATLAB.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 6(34) (2015): 196–204.
2. Kasyanov, V. N., and V. A. Evstigneev. *Grafy v programmirovanii: vizualizacija i primenenie*. SPb.: BHV-Peterburg, 2003.
3. Sazonov, A. E., and V. V. Derjabin. “Forecasting to paths of the motion ship with the help of neyronny network.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 3(22) (2013): 6–13.
4. Gao, Song, and He Huang. “Real-time traveler information for optimal adaptive routing in stochastic time-dependent networks.” *Transportation Research Part C: Emerging Technologies* 21.1 (2012): 196–213. DOI: 10.1016/j.trc.2011.09.007.
5. Fu, Liping. “An adaptive routing algorithm for in-vehicle route guidance systems with real-time information.” *Transportation Research Part B: Methodological* 35.8 (2001): 749–765. DOI: 10.1016/S0191-2615(00)00019-9.
6. Saharov, V. V., A. A. Kuzmin, and A. A. Chertkov. *Modeli i algoritmy optimizacii tehnologicheskikh processov na obectah vodnogo transporta v srede MATLAB*: monografija. SPb.: GUMRF imeni adm. S.O. Makarova, 2015.

7. 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.
8. 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 admiral S.O. Makarova* 4(32) (2015): 207–215.
9. Saharov, V. V., A. A. Chertkov, and D. S. Tormashev. "Algorithm optimal'nogo planirovaniya gruppovogo vzaimodeistviya robotov." *Morskoj Vestnik* 4 (2014):119–122.
10. Kalyaev, I. A., A. R. Gaiduk, and S. G. Kapustyan. *Modeli i algoritmy kollektivnogo upravleniya v grup-pah robotov*. M: Fizmatlit, 2009.
11. Kalyaev, I. A., A. R. Gaiduk, and S. G. Kapustyan. "UAV group control in task of order forming." *Robot-ics and Technical Cybernetics* 4(5) (2014): 28–39.
12. Aleksandrov, V. A., and A. I. Kobrin. "Collective operational environment subspace dividing algorithm for the robotic group in the coverage area problem." *Proceedings of Higher Educational Institutions. Machine Build-ing* 9 (2011): 65–69.
13. Aleksandrov, V. A., and A. I. Kobrin. "The hardware-software complex for modeling of grouped control problems for mobile robots." *MPEI Vestnik* 3 (2011): 88–95.
14. Kirsanov, Mihail Nikolaevich. "Analysis of algorithms for the selection of optimal routes the group's vessels." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admiral S.O. Makarova* 2(36) (2016): 183–190. DOI: 10.21821/2309-5180-2016-8-2-183-190.
15. Dedkov, V. A., and M. N. Kirsanov. "The problem of control for multirobot systems." *Инновационные информационные технологии* 2.2 (2013): 206–213.
16. Kirsanov, Andrey, Sreenatha G. Anavatti, and Tapabrata Ray. "Path planning for the autonomous under-water vehicle." *International Conference on Swarm, Evolutionary, and Memetic Computing*. Springer, Cham, 2013: 476–486. DOI: 10.1007/978-3-319-03756-1_43.
17. Zeng, Zheng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas. "A survey on path planning for per-sistent autonomy of autonomous underwater vehicles." *Ocean Engineering* 110 (2015): 303–313. DOI: 10.1016/j.oceaneng.2015.10.007.
18. Bellman, Richard. "On a Routing Problem." *Quarterly of Applied Mathematics* 16.1 (1958): 87–90. DOI: 10.1090/qam/102435.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Чертков Александр Александрович —
кандидат 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 AUTHOR

Chertkov, Alexandr A. —
PhD, 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

Статья поступила в редакцию 15 сентября 2017 г.
Received: September 15, 2017.