

DOI: 10.21821/2309-5180-2022-14-4-615-627

## STREAMING NETWORK ROUTING BASED ON BELLMAN–FORD ALGORITHM MODIFICATION

**A. A. Chertkov, Ya. N. Kask, L. B. Ochina**

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

*The operational task of automating the construction and routing of the network model with the known coordinates of the conditional goals set for a group of vessels to achieve them in the minimum time is solved; it makes it possible to obtain the reserves of running time necessary for saving fuel and energy, taking into account the load, the cost of cargo, transportation costs, logistics characteristics, etc. It is emphasized that in stormy weather conditions and vessel management in situations related to schedule correction, flexible operational decisions of dispatching services, made on the basis of numerical optimization methods using modern computing environments, are necessary. In this regard, the method of dynamic programming, implemented using the Bellman-Ford routing algorithm, which is supplemented by a recursive step-by-step optimization procedure that removes the limitation of the algorithm in the presence of inversely oriented edges with negative weights in the graph, is discussed in the paper. In the presence of negative weights, there are conditions for the appearance of a negative cycle in the graph, in which the practical implementation of the Bellman-Ford algorithm will become impossible due to an endless cycle of relaxations (attenuation) of the vertices weights included in this cycle. Hence, at a limited period of time for weighing all vertices (passes on all edges), the algorithm can give a knowingly false result. The proposed procedure for modifying the well-known Bellman-Ford algorithm eliminates this limitation and allows it to be used not only for estimating the shortest paths in a network containing arcs with negative weights, but also to detect negative cycles in it. The modified Bellman-Ford algorithm is implemented as a program compiled in MATLAB codes, and it is demonstrated by the example of automated construction and calculation of a network model containing both positive and negative edges (flows), using a recursive procedure of step-by-step optimization. It is shown that the proposed model, unlike the known models, eliminates the limitations caused by the presence of negative cycles in the network model, which makes it possible to automate the search for the shortest paths to conditional goals by the functional means of the MATLAB environment. The constructed computer model is simple and compact. The proposed algorithm and the recursive procedure are recommended for finding energy-efficient solutions for managing mobile objects in water transport.*

*Keywords: automation, shortest route, negative cycle, algorithm, weight of graph ribs, optimization, computational procedure.*

### For citation:

Chertkov, Alexandr A., Yaroslav N. Kask, and Ludmila B. Ochina. "Streaming network routing based on Bellman–Ford algorithm modification." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admiral S. O. Makarova* 14.4 (2022): 615–627. DOI: 10.21821/2309-5180-2022-14-4-615-627.

**УДК 681.5**

## МАРШРУТИЗАЦИЯ ПОТОКОВОЙ СЕТИ НА ОСНОВЕ МОДИФИКАЦИИ АЛГОРИТМА БЕЛЛМАНА – ФОРДА

**А. А. Чертков, Я. Н. Каск, Л. Б. Очина**

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

*В работе решена операционная задача автоматизации построения и маршрутизации сетевой модели с известными координатами условных целей, поставленных группе судов для их достижения в течение минимального времени, что позволяет получить резервы ходового времени, необходимые для экономии топлива и энергии с учетом загрузки, стоимости груза, расходов на перевозки и др. Подчеркивается, что в условиях штормовой погоды и при управлении судами в ситуациях, связанных с корректировкой расписания, необходимы гибкие оперативные решения диспетчерских служб, принимаемые на основе численных*

методов оптимизации с использованием современных вычислительных сред. В связи с этим в работе рассмотрен метод динамического программирования, реализуемый с использованием алгоритма маршрутизации Беллмана – Форда, дополненный рекурсивной процедурой пошаговой оптимизации, снимающей ограничение алгоритма при наличии в графе обратно ориентированных ребер с отрицательными весами. Отмечается, что при наличии отрицательных весов возникают условия появления отрицательного цикла в графе, при котором практическая реализация алгоритма Беллмана – Форда становится невозможной из-за бесконечного цикла релаксаций (ослабления) весов вершин, входящих в этот цикл. Отсюда на ограниченном отрезке времени взвешивания всех вершин (проходов по всем ребрам) алгоритм может дать заведомо ложный результат. Предложенная процедура модификации известного алгоритма Беллмана – Форда устраняет это ограничение, позволяя применять его не только для оценок кратчайших путей в сети, содержащей дуги с отрицательными весами, но и для обнаружения в ней отрицательных циклов. Модифицированный алгоритм Беллмана – Форда реализован в виде программы, составленной в кодах MATLAB, и демонстрируется на примере автоматизированного построения и расчета сетевой модели, содержащей как положительные, так и отрицательные ребра (потoki) с применением рекурсивной процедуры пошаговой оптимизации. Показано, что предложенная модель, в отличие от известных, устраняет ограничения, обусловленные наличием отрицательных циклов в сетевой модели, что позволяет автоматизировать поиск кратчайших путей до условных целей функциональными средствами среды MATLAB. При этом построенная компьютерная модель обладает простотой и компактностью. Предложенные алгоритм и рекурсивная процедура рекомендуются для поиска энергоэффективных решений при управлении подвижными объектами на водном транспорте.

**Ключевые слова:** автоматизация, кратчайший маршрут, отрицательный цикл, алгоритм, веса ребер графа, оптимизация, вычислительная процедура.

#### Для цитирования:

Чертков А. А. Маршрутизация потоковой сети на основе модификации алгоритма Беллмана – Форда / А. А. Чертков, Я. Н. Каск, Л. Б. Очина // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2022. — Т. 14. — № 4. — С. 615–627. DOI: 10.21821/2309-5180-2022-14-4-615-627.

### Введение (Introduction)

В условиях цифровизации экономики, в стратегии которой важная роль отводится широкому использованию безэкипажных транспортных средств, особенно актуальными становятся энергоэффективные технологии организации движения, позволяющие экономить топливо и энергию при изменениях ситуации как на линии, так и в портах назначения. Энергосбережение во многом определяется возможностью оценивания резервов ходового времени как одного судна при его движении до порта назначения, так и группы судов при управлении ими в ситуациях, связанных с коррекцией расписания. Полученные таким образом резервы времени могут быть использованы для экономии топлива и энергии при прохождении шлюзованных участков пути, требующих гибких оперативных решений диспетчерских служб. Образование резервов ходового времени во многом зависит от поиска оптимальных маршрутов движения судов с учетом загрузки, стоимости груза, расходов на перевозки, характеристик логистики и других факторов.

Поиск и оценка оптимальных (наикратчайших) маршрутов движения судов могут производиться методами математического программирования с использованием исследования операций. Этот класс задач принято решать с помощью теории графов, с введением узлов и дуг с различными, в том числе отрицательными весами, что позволяет в удобной форме графически интерпретировать как объекты, так и их модели. Отрицательность весов вводится для автоматизации учета влияния изменяющихся внешних условий на планируемый маршрут движения судна. Основным ее смысл заключается в том, что под классом «отрицательных» весов ребер понимается такой класс элементов, вес каждого из которых меньше веса любого неотрицательного элемента, принятого за среднее значение.

В рассматриваемом случае отрицательный вес отдельных ребер графа обусловлен влиянием изменяющейся внешней среды (попутного ветра, течения, волнения и др.), при котором расход топлива на этих участках будет ниже заданного среднего значения. В данной работе под оптимальным маршрутом понимается маршрут подвижного объекта (судна) с оценкой наименьшей суммы отклонений расхода топлива от заданного среднего значения на всех участках пути.

Проблема *маршрутизации графов с отрицательными весами ребер* состоит в том, что при наличии в них отрицательных циклов кратчайшего пути до некоторых вершин может не существовать, поскольку он должен быть равен «минус бесконечности». Поэтому целью работы является создание такого алгоритма, который позволяет находить циклы с отрицательными весами и выделять вершины графа, связанные такими циклами.

Для моделирования и маршрутизации на графах потоковых сетей созданы многочисленные алгоритмы, фактически формирующие инструменты для поиска кратчайших путей. К основополагающим алгоритмам в этой области исследования относятся алгоритмы Дейкстры [1], линейного и целочисленного программирования [2]–[4], Беллмана – Форда [5]–[6], Флойда – Уоршелла, подробно рассмотренные в работах [7]–[8]. Основу этих алгоритмов, используемых для решения оптимизационных задач, составляет общий принцип динамического программирования, названный Р. Беллманом *принципом оптимальности*. Согласно данному принципу решение основной задачи оптимизации может являться результатом оптимальных решений ее подзадач, получаемых рекуррентно на каждом шаге вычислений.

Следует отметить, что научная и практическая значимость универсальных алгоритмов маршрутизации сетевых моделей неуклонно возрастает. Это связано с тем, что формирование маршрутов движения судов, сходящихся в узлах и на операционных пространствах, на подходах к портам, при проведении лоцманских проводок, в условиях приливов и отливов требует принятия особых мер по обеспечению безопасности и безаварийности выполнения всех технологических операций. Причем системное решение указанных задач потребует разработки оптимизационных алгоритмов на сетевых моделях и создания инструментальной базы для робототехнического безэкипажного управления судами [9]–[11]. Цифровизация и развитие интеллектуальных систем автоматизации технологических процессов на объектах водного транспорта с использованием нейронных сетей [12], генетических алгоритмов, а также систем нечеткой логики нашли применение в моделях и алгоритмах принятия решений при управлении подвижными объектами в операционных пространствах высокой размерности и отражены в работах [13]–[16].

### Методы и материалы (Methods and Materials)

В отличие от алгоритмов, рассмотренных в работах [2]–[8], предлагается решение задачи не только оценки кратчайших путей между заданными вершинами графа с обнаружением циклов с отрицательным суммарным весом звеньев, но и алгоритмического его конструирования с заданной конфигурацией и оцифровкой всех вершин и ребер. Кроме того, решение задачи о кратчайшем пути дополнено оценкой кратчайших путей между любыми двумя узлами сети, а также оценкой переменных состояния на взвешенных графах как с положительными, так и с отрицательными весами ребер, что обеспечивает возможность введения корректирующих действий по результатам вычислений с учетом сходимости решения к оптимуму.

*Формальное описание.* Прежде чем рассматривать разработанный алгоритм для решения практических задач маршрутизации, покажем возможность алгоритмического построения рассчитываемого графа с последующей его оцифровкой, используя арсенал средств MATLAB.

Построим в кодах MATLAB граф, содержащий  $n = 6$  вершин, воспользовавшись файловой функцией `gplot`, имеющейся в инструментарии MATLAB с синтаксисом

Function h = `gplot`(V,E,kind,vkind,ekind,sa),

где входными параметрами служат:

$V(n, 2)$  or  $(n, 3)$  — координаты вершин (1-й столбец —  $x$ , 2-й —  $y$ ) и, может быть, 3-й столбец — вес;  $n$  — число вершин;

$E(m, 2)$  or  $(m, 3)$  — ребра графа (стрелки диграфа) и их вес; 1-й и 2-й элементы каждой строки — это номера вершин; 3-й элемент каждой строки — вес стрелки (ребра, дуги);

$m$  — количество стрелок (ребер, дуг);

kind — вид графика при следующих условиях:

‘g’ — нарисовать граф (аргумент необязательный, реализуется по умолчанию), ‘d’ — нарисовать диграф;

vkind — вид меток для вершин (необязательно);

ekind — вид меток для ребер или стрелок (необязательно);

sa — размер стрелок (по умолчанию — 1).

Пусть элементы матрицы V примут следующие значения, соответствующие координатам вершин рассматриваемого графа (в синтаксисе MATLAB):

$$V = [0 \ 0; 1 \ 0.6; 0.8 \ -0.6; 2.5 \ 0.6; 2.7 \ -0.6; 3.5 \ 0],$$

в котором ребра и их веса задаются следующими значениями элементов матрицы E:

$$E = [1 \ 2 \ 6; 1 \ 3 \ 7; 2 \ 3 \ 8; 2 \ 4 \ 5; 2 \ 5 \ -4; 3 \ 4 \ -3;$$

$$3 \ 5 \ 9; 4 \ 2 \ -2; 4 \ 6 \ 7; 5 \ 1 \ 2; 5 \ 4 \ 7; 5 \ 6 \ 4].$$

Установив шрифт с помощью оператора set(H,'PropertyName', PropertyValue), в котором аргументы имеют конкретные значения:

set([], 'CurrentAxes', 'FontName', 'Times New Roman Cyr', 'FontSize', 11),

получим граф, подлежащий оптимальной маршрутизации.

Для построения такого графа не требуется вводить массивы данных (координат) для оцифровки вершин и весов соединительных ребер, что значительно упрощает его построение. Такой алгоритм построения выгодно отличает его от известных [8] и может использоваться для построения взвешенных графов любой сложности.

**The digraf with weighted edges**

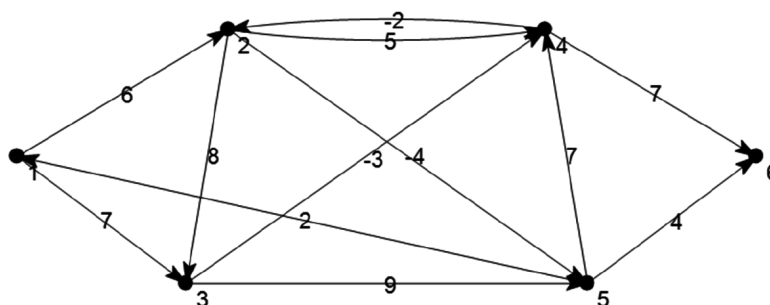


Рис. 1. Исходный взвешенный граф без отрицательных циклов

Как видно из рис. 1, исходный граф состоит из ребер как с положительными, так и с отрицательными весами. Причем на графе можно выделить два цикла между вершинами 2→3→4 и 2→5→4, для каждого из которых алгебраическая сумма весов ребер будет положительной величиной.

На первом этапе исследования для исходного сетевого графа, не содержащего отрицательных циклов, будет решаться задача поиска оптимальных маршрутов между любыми парами вершин, а также закономерности, позволяющей по полученным результатам судить об отсутствии в графе отрицательных циклов. За оптимальный принят маршрут подвижного объекта (судна) с оценкой наименьшей суммы отклонений расхода топлива от заданного среднего значения на всех участках пути. Основой рассматриваемого способа сетевой маршрутизации является алгоритм Беллмана – Форда, подробно рассмотренный в работе [5], согласно которому выполняется пошаговая маршрутизация графа.

Напомним, что алгоритм Беллмана – Форда позволяет решить задачу о кратчайшем пути из произвольной вершины до любой из остальных вершин в общем случае, когда вес каждого из ребер может быть отрицательным. Алгоритм основан на последовательном взвешивании смежных с вершиной-входом вершин, начальный вес которых задан (при инициализации графа) бесконечностью или числом, заведомо большим суммарного веса всех ребер рассчитываемого графа. Эти веса являются начальными оценками веса кратчайшего пути из истока с нулевым значением веса, которые в процессе прохода по всем ребрам графа ослабляются (улучшаются).

Приведем фрагмент программы инициализации вершин графа:

```
%Присвоение начальных весов каждой из вершин, т.е. их инициализация
for j=1:n
    L(j)=100;
end.
```

Непосредственно алгоритм Беллмана – Форда состоит из нескольких фаз, определяемых числом вершин графа. На начальной фазе задается массив расстояний  $LP[0..0]$ , который по завершению алгоритма содержит итоговый результат. В пределах каждой фазы, соответствующей отображаемой вершине, выполняется взвешивание смежных с ней вершин, или, иначе, просмотр всех ребер графа, согласно которому делается попытка релаксации (ослабления) или улучшения начальных оценок веса кратчайшего пути из вершины-истока. Релаксация вдоль ребра означает попытку улучшения значения  $LP(j)$  за счет значения  $LP(i) + w(i, j)$  и, соответственно, ослабления веса взвешиваемой вершины  $j$  с использованием ребра  $(i, j)$  весом  $w(i, j)$ .

Если в процессе взвешивания очередной вершины, т. е. суммирования весов смежной (предшествующей) ей вершины и соединяющего их ребра, полученный результат будет меньше веса взвешиваемой вершины:

$$LP(i) + w(i, j) < L(j),$$

то этот результат, являющийся ослаблением, будет служить новой оценкой веса кратчайшего пути к этой вершине:

$$L(j) = LP(i) + w(i, j); \quad LP(j) = L(j).$$

Таким образом, в этом алгоритме используется ослабление (релаксация), в результате которого величина  $LP[j]$ , 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  $v$ .

Приведем фрагмент программы, иллюстрирующий процессы ослабления (улучшения) оценок веса кратчайших путей из истока  $s$  к каждой из вершин  $v \in V$  при каждой итерации (проходов по ребрам графа):

```
n=6; %количество вершин;
%Матрица весов ребер:
w=[0 6 7 100 100 100;100 0 8 5 -4 100;100 100 0 -3 9 100;
    100 -2 100 0 100 7;2 100 100 7 0 4;100 100 100 100 100 0];
%Одномерный массив оценок кратчайших путей из истока к каждой вершине
LP=[0 0 0 0 0 0];
i=1 % Первая итерация:
while i<=n
    for j=1:n
        if L(j)>LP(i)+w(i,j) %релаксация пути:
            L(j)=LP(i)+w(i,j);
            LP(j)=L(j); %Формирование оценок кратчайших путей
            P(i,j)=i; %Формирование списка предшествующих (конечным) вершин
            F(i,j)=j; %Формирование списка конечных вершин кратчайших путей
```



```

end
end
LP
i=i+1 % Номер итерации
end.

```

В качестве пояснения к программе добавим, что в матрице весов  $w(i, j)$  для обозначения веса отсутствующих ребер между узлами  $i, j$  используются числа, которые заведомо больше длины любого потенциального пути в графе (например, число 100). Данный алгоритм в том виде, как его описали Р. Беллман и Л. Форд, прекрасно работает как с положительными, так и с отрицательными весами ребер только в ориентированном графе, в котором все ребра следуют в одном направлении (от вершин с меньшими номерами к вершинам с большими номерами). При наличии обратно ориентированных ребер (стрелок) с отрицательными весами, направленных от вершин с большими номерами к вершинам с меньшими номерами данный алгоритм не работает, т. е. дает заведомо ложный результат. Это видно на наглядном примере рассматриваемого графа. Прямой проход по ребрам графа занимает первые три итерации (просмотр ребер смежных с вершинами 1, 2, 3), для которых нет обратно ориентированных ребер.

При *первой итерации* ослабление (релаксация) оценок кратчайших путей осуществляется только для вершин 2 и 3 графа, кратчайшие пути в которые проходят по смежным вершине 1 ребрам (1, 2) и (1, 3). В результате их веса, равные оценкам кратчайшего пути до них  $LP(2)$  и  $LP(3)$ , станут равными весам соединяющим их с вершиной 1 ребер 6 и 7 соответственно:

$$LP(2) = LP(1) + w(1, 2) = 0 + 6 = 6;$$

$$LP(3) = LP(1) + w(1, 3) = 0 + 7 = 7.$$

При *второй итерации* ослабление (релаксация) оценок кратчайших путей осуществляется только для вершин 4 и 5 графа, кратчайшие пути в которые проходят по смежным вершине 2 ребрам (2, 4) и (2, 5). В результате их веса, равные оценкам кратчайших путей до них  $LP(4)$  и  $LP(5)$  с учетом веса вершины 2, станут равными 11 и 2 соответственно:

$$LP(4) = LP(2) + w(2, 4) = 6 + 5 = 11 < 100;$$

$$LP(5) = LP(2) + w(2, 5) = 6 + (-4) = 2 < 100.$$

При *третьей итерации* ослабление (релаксация) оценок кратчайших путей осуществляется только для вершины 4 графа, кратчайший путь в которую проходит по соединяющему их ребру (3, 4). В результате ослабления ее вес, равный оценке кратчайшего пути до нее  $LP(4)$  с учетом веса вершины 3, станет равными 4:

$$LP(4) = LP(3) + w(3, 4) = 7 + (-3) = 4 < 11.$$

Как видно из первых трех итераций, соответствующих прямому проходу (просмотру) ребер никаких неожиданностей и противоречий алгоритму Беллмана – Форда нет.

Основные неожиданности (неприятности) начинаются с *четвертой итерации*, где для просмотра смежного вершине 4 ребра (4, 2) совершается обратный проход по этому ребру с отрицательным весом, в результате которого происходит ослабление оценки кратчайшего пути до вершины 2 с  $LP(2) = 6$  до  $LP(2) = 2$ :

$$LP(2) = LP(4) + w(4, 2) = 4 + (-2) = 2 < 6,$$

которая уже никак не влияет в *пятой* (последней) *итерации* (согласно рассматриваемому алгоритму) на окончательную оценку кратчайшего пути до вершины 5, равную  $LP(5) = 2$ , что не соответствует, как видно из рис. 1, точной оценке кратчайшего пути до вершины 5, которая должна составлять

$$LP(5) = LP(2) + w(2, 5) = 2 + (-4) = -2 < 2.$$

Таким образом, налицо ограничение известного алгоритма Беллмана – Форда, для преодоления которого предлагается ввести в этот алгоритм дополнительную процедуру «довзвешивания». Согласно этой процедуре для вершины (младшей по номеру), в которой произошло ослабление оценки кратчайшего пути при проходе ребра из старшей вершины (как в четвертой итерации), вновь возобновляется операция взвешивания (ослабления) смежных с ней вершин, старших по номеру.

Приведем пример фрагмента программы, демонстрирующей процедуру «довзвешивания»:

```
if j<i
    for I=j:n
        f L(I)>L(j)+C(j,I)
          L(I)=L(j)+C(j,I);
          P(I,j)=j; %Формирование списка предшествующих (конечным) вершин
          p=I;
          LP(p)=L(I); %Массив оценок кратчайшего пути из истока к каждой вершине
          F(I,j)=I; %Формирование конечных вершин кратчайших путей
        end
    end
end
```

Следует отметить, что благодаря процедуре «довзвешивания» появилась возможность определения отрицательных циклов в графе с любым направлением ребер (как с прямым, так и с обратным).

### Результаты (Results)

В качестве иллюстрации работы модифицированного алгоритма Беллмана – Форда, а также подтверждения корректности процедуры «довзвешивания» приведем результаты расчета кратчайших путей на примере рассматриваемого графа (см. рис. 1) с различной ориентацией и наличием отрицательных весов его ребер:

```
>> orgrB _ F2
i =
    1
LP =
    0     6     7     0     0     0
i =
    2
LP =
    0     6     7    11     2     0
i =
    3
LP =
    0     6     7     4     2     0
i =
    4
LP =
    0     2     7     4    -2    11
i =
    5
LP =
    0     2     7     4    -2     2
P =
    1     1     1     0     0     0
    0     0     0     2     2     0
```

$$F = \begin{matrix} & \begin{matrix} 0 & 0 & 0 & 3 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 4 \\ 2 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 4 \\ 5 \\ 6 \end{matrix} \\ \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 2 \\ 0 \\ 0 \\ 2 \\ 5 \end{matrix} & \begin{matrix} 3 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 4 \\ 4 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 5 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{matrix} 0 \\ 0 \\ 0 \\ 6 \\ 6 \end{matrix} \end{matrix}$$

>>

Численные результаты решения задачи вычисления кратчайших путей на графе (см. рис. 1) из вершины 1 во все остальные вершины, соответствующие первому этапу маршрутизации (без отрицательных циклов), содержатся в массиве  $LP = [0 \ 2 \ 7 \ 4 \ -2 \ 2]$ , построенном на пятом шаге ( $i = 5$ ) алгоритма, номер которого определяется условием  $(n - 1)$ . На шестом шаге алгоритма ( $i = n$ ) ничего не происходит и результаты решения повторяются. Как видно из массива  $LP$ , оценками кратчайших путей из вершины 1 во все остальные вершины в порядке их нумерации являются значения 2, 7, 4, -2 и 2, помещенные в следующую таблицу:

**Кратчайшие расстояния и маршруты до вершин графа**

Номера вершин	1	2	3	4	5	6
Кратчайшие пути	0	2	7	4	-2	2
Маршруты	—	1-3-4-2	1-3	1-3-4	1-3-4-2-5	1-3-4-2-5-6

Для восстановления кратчайших путей на графе в алгоритм введены массив  $F$ , содержащий на каждом шаге итерации номера конечных вершин кратчайших путей, где произошло улучшение оценок, и массив  $P$  предшествующих им вершин. Причем по аналогии с массивом  $LP$  можно воспользоваться значениями массивов  $F$  и  $P$ , полученными также в  $(n-1)$ -м шаге алгоритма, т. е. на пятом шаге.

Найдем искомые ребра (звенья) потоковой сети, которые составляют кратчайший путь из вершины 1 в вершину 6 рассматриваемого графа (см. рис. 1). Как видно из массива  $F$ , последней вершиной, в которой произошло улучшение оценки, является шестая вершина, для которой предпоследней вершиной будет, как видно из массива  $P$  в той же позиции, пятая вершина. Обратившись снова к массиву  $F$ , находим в его предпоследней строке вершину 5. Этой вершине предшествует, как видно из массива  $P$  на той же позиции, вершина 2. Вершине 2, отмеченной в четвертой строке на второй позиции массива  $F$ , предшествует, как видно из массива  $P$  на той же позиции, вершина 4, которой, как видно из третьей строки массива  $F$ , в той же позиции предшествует вершина 3. По аналогии для вершины 3 предшествующей и исходной вершиной искомого кратчайшего пути будет вершина 1. Отсюда восстановленный кратчайший путь из первой в шестую вершину проходит по следующим вершинам (см. таблицу):

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 6.$$

На рис. 2 ребра графа, составляющие кратчайший маршрут из первой в шестую вершины, выделены жирными линиями.

Целью второго этапа исследования служит проверка наличия в графе отрицательных циклов на конкретном примере. Для этого рассматривается та же задача поиска кратчайших маршрутов между любыми парами вершин, но для сетевого графа (рис. 3), в котором, в отличие от исходного (см. рис. 1), вес ребра (5, 4) изменился и стал равным пяти условным единицам ( $E_{54}=5$ ), а не семи, как было в исходном графе.



The digraf with weighted edges

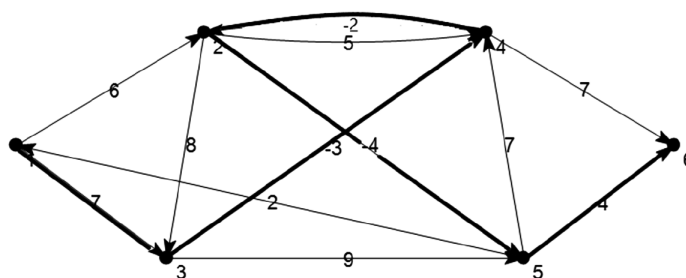


Рис. 2. Кратчайший путь на графе между парой вершин 1 и 6

The digraf with weighted edges

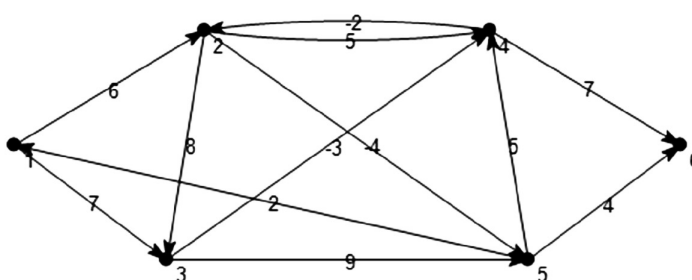


Рис. 3. Взвешенный потоковый граф с отрицательным циклом 2 – 5 – 4 – 2

В связи с этим суммарный вес ребер, образующих цикл между вершинами 2, 5, 4, 2 графа (рис. 3):

$$E_{25} + E_{54} + E_{42} = (-3) + 4 + (-2) = -1,$$

является отрицательной величиной. Таким образом, созданы условия для существования отрицательного цикла.

Выполнив по тому же алгоритму аналогичные итерации на каждом шаге, получим конечные матрицы  $P$  и  $F$ :

$$LP = \begin{matrix} & 0 & 2 & 7 & 3 & -2 & 2 \end{matrix}$$

$$P = \begin{matrix} & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 4 \\ 0 & 2 & 0 & 5 & 0 & 5 \\ 0 & 0 & 0 & 4 & 0 & 0 \end{matrix}$$

$$F = \begin{matrix} & 1 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 5 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 6 \\ 0 & 5 & 0 & 4 & 0 & 6 \\ 0 & 0 & 0 & 6 & 0 & 0 \end{matrix}$$

>>

Как видно из массива  $F$ , последней вершиной, для которой произошло улучшение оценки, является четвертая, для которой предшествующей вершиной будет, как видно из массива  $P$  в той же позиции, пятая вершина. Обратившись снова к массиву  $F$ , находим в предпоследней строке вершину 5. Этой вершине предшествует, как видно из массива  $P$  на той же позиции, вершина 2. Вершине 2, отмеченной в четвертой строке на второй позиции массива  $F$ , вновь предшествует, как видно из массива  $P$  на той же позиции, вершина 4, которой в той же позиции массива  $P$  предшествует вершина 3. Аналогично для вершины 3 предшествующей и исходной вершиной искомого кратчайшего пути будет вершина 1. Отсюда кратчайший путь из первой в шестую вершину составит

$$1 \rightarrow 3 \rightarrow 4 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 6.$$

Таким образом, на ограниченном числе итераций ( $i \leq n$ ), определяемом рассматриваемым алгоритмом, найденный кратчайший путь содержит отрицательный цикл

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

### Обсуждение (Discussion)

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 — кратчайший путь не содержит повторяющихся вершин.

Алгоритм Беллмана – Форда позволяет очень просто определить, существует ли в графе  $G$  отрицательный цикл, достижимый из вершины  $s$ . Если при исполнении очередной итерации в массиве  $F$  конечных вершин кратчайшего пути дважды повторятся одна и та же вершина, то в графе есть отрицательный цикл, достижимый из исходной вершины. С учетом этого необходимо отслеживать изменения в графе и как только они закончатся, сделать выход из цикла (дальнейшие итерации будут бессмысленны).

Анализ полученных результатов на втором этапе показывает, что проходимость графа завершается на четвертом узле, который является единственным транзитным узлом в графе. Следовательно, рассматриваемый граф должен содержать цикл с отрицательным весом, т. е. в нем отсутствуют сформулированные ранее признаки неотрицательности циклов. Отсюда следует, что данный алгоритм позволяет не только зафиксировать наличие отрицательного цикла в графе, но и указать проблемный узел, с которого нарушается проходимость графа. Таким образом, выполненная проверка подтвердила практическую применимость и эффективность модифицированного алгоритма Беллмана – Форда не только для оптимальной маршрутизации сетевого графа, но и для сигнализации о наличии отрицательного цикла в нем и места непроходимости графа.

### Выводы (Summary)

На основе проведенного исследования можно сделать следующие выводы:

1. Разработана универсальная компьютерная модель, позволяющая автоматизировать не только операции определения кратчайших маршрутов между любой парой транзитных узлов, но и конструирования самой потоковой (транспортной) сети с восстановлением всех транзитных узлов, составляющих искомый маршрут. Кроме того, модель пригодна для оперативной оценки энергоэффективности выбранных маршрутов движения судов и других подвижных объектов с учетом изменяющихся условий плавания в рейсе.

2. Предложена рекурсивная процедура «довзвешивания», позволяющая устранить ограничения алгоритма Беллмана – Форда, связанные с некорректностью его работы при наличии в графе обратно ориентированных ребер (стрелок) с отрицательными весами, направленных от вершин с большими номерами к вершинам с меньшими номерами.

3. Достоинствами предложенной модификации алгоритма Беллмана – Форда, в отличие от алгоритмов целочисленного линейного программирования, является простота программной реализации с применением процедур последовательного приближения оценок кратчайших путей к оптимальным значениям, полученным на основе динамического программирования, реализуемого с помощью средств универсального инструментария Optimization Toolbox среды MATLAB.

4. Корректность и эффективность предложенной модели и ее программной реализации подтверждена на конкретном примере автоматизации расчета кратчайших путей с полным анализом возможных вариантов на каждом шаге итерации, а также алгоритмического конструирования и оцифровки графа транспортной сети со сложной конфигурацией при наличии ребер с отрицательными весами.

## 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. DOI: 10.21821/2309-5180-2015-7-6-196-204.
2. *Чертков А. А.* Автоматизация определения критического пути в логистической системе / А. А. Чертков, А. А. Вардомская, А. А. Дмитриев // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2015. — № 5(33). — С. 194–200. DOI: 10.21821/2309-5180-2015-7-5-194-200.
3. *Сахаров В. В.* Автоматизация поиска оптимальных маршрутов и грузовых потоков в транспортных сетях средствами целочисленного линейного программирования / В. В. Сахаров, И. А. Сикарев, А. А. Чертков // Вестник государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2018. — Т. 10. — № 3. — С. 647–657. DOI: 10.21821/2309-5180-2018-10-3-647-657.
4. *Сахаров В. В.* Алгоритм трафика перевозки грузов с обеспечением минимума транспортной работы / В. В. Сахаров, А. А. Чертков, А. А. Дмитриев // Вестник государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2016. — № 1(35). — С. 180–188. DOI: 10.21821/2309-5180-2016-8-1-180-188.
5. *Bellman R.* On a Routing Problem / R. Bellman // Quarterly of Applied Mathematics. — 1958. — Vol. 16. — № 1. — С. 87–90. DOI: 10.1090/qam/102435
6. *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.
7. *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.
8. *Sakharov V. V.* Network routing method for ships and other moving objects using MATLAB / V. V. Sakharov, A. A. Chertkov, I. B. Ariefjew // Zeszyty Naukowe Akademii Morskiej w Szczecinie. — 2020. — No. 62 (134). — Рр. 61–68. DOI: 10.17402/420.
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. *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.
11. *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.

12. 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.

13. 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.

14. 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.

15. Федоренко К. В. Исследование основных параметров генетического алгоритма применительно к задаче поиска оптимального маршрута / К. В. Федоренко, А. Л. Оловянных // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2017. — Т. 9. — № 4. — С. 714–723. DOI: 10.21821/2309-5180-2017-9-4-714-723.

16. 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.

## REFERENCES

1. Chertkov, Alexandr Alexandrovich, Anna Alexandrovna Vardomskaia, 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. DOI: 10.21821/2309-5180-2015-7-6-196-204.

2. Chertkov, Alexandr Alexandrovich, Anna Alexandrovna Vardomskaia, and Alexandr Alexandrovich Dmitriev. “Automation to define critical way in logistical system”. *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 5(33) (2015): 194–200. DOI: 10.21821/2309-5180-2015-7-5-194-200.

3. Saharov, Vladimir V., Igor A. Sikarev, and Alexander A. Chertkov. “Automating search optimal routes and goods flows in transport networks means the integer linear programming.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 10.3 (2018): 647–657. DOI: 10.21821/2309-5180-2018-10-3-647-657.

4. Saharov, Vladimir Vasilevich, Alexandr Alexandrovich Chertkov, and Alexandr Alexandrovich Dmitriev. “Ensuring minimum transport work algorithm for goods traffic”. *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 1(35) (2016): 180–188. DOI: 10.21821/2309-5180-2016-8-1-180-188.

5. Bellman, Richard. “On a Routing Problem.” *Quarterly of Applied Mathematics* 16.1 (1958): 87–90. DOI: 10.1090/qam/102435.

6. Chertkov, Alexander A. “Automation selection shortcuts routes of ships on the basis of modified BellmanFord 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.

7. 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 admirala S.O. Makarova* 11.2 (2019): 230–242. DOI: 10.21821/2309-5180-2019-11-2-230-242.

8. Sakharov, Vladimir V., Alexandr A. Chertkov, and Igor B. Ariefjew. “Network routing method for ships and other moving objects using MATLAB.” *Zeszyty Naukowe Akademii Morskiej w Szczecinie* 62(134) (2020): 61–68. DOI: 10.17402/420.

9. Saharov, V.V., A.A. Chertkov, and D.S. Tormašev. “Optimal planning group communication Algorithm of robots.” *Morskoy Vestnik* 4 (2014): 119–122.

10. 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.

11. Dedkov, V. A., and M. N. Kirsanov. “The problem of control for multirobot systems.” *Innovatsionnye informatsionnye tekhnologii* 2.2 (2013): 206–213.

12. Kirsanov, Mihail Nikolaevich. “Analysis of algorithms for the selection of optimal routes the group’s vessels.” *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 2(36) (2016): 183–190. DOI: 10.21821/2309-5180-2016-8-2-183-190.

13. 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.

14. Fonseca, Carlos M., and Peter J. Fleming. "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization." *Genetic Algorithms: Proceedings of the Fifth International Conference*. Edited by S. Forrest. San Mateo, CA: Morgan Kaufmann, 1993. 416–423.

15. Fedorenko, Kirill V., and Arkadii L. Olovyanikov. "Research of the main parameters of the genetic algorithm for the problem of searching the optimal route." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 9.4 (2017): 714–723. DOI: 10.21821/2309-5180-2017-9-4-714-723.

16. 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.

#### ИНФОРМАЦИЯ ОБ АВТОРАХ

##### **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](mailto:chertkov51@mail.ru),  
[kaf\\_electricautomatic@gumrf.ru](mailto:kaf_electricautomatic@gumrf.ru)

##### **Каск Ярослав Николаевич —**

кандидат технических наук  
ФГБОУ ВО «ГУМРФ имени адмирала  
С.О. Макарова»  
198035, Российская Федерация, Санкт-Петербург,  
ул. Двинская, 5/7  
e-mail: [rgam2010@yandex.ru](mailto:rgam2010@yandex.ru), [kaf\\_electricautomatic@gumrf.ru](mailto: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: [rgam2010@yandex.ru](mailto:rgam2010@yandex.ru), [kaf\\_electricautomatic@gumrf.ru](mailto:kaf_electricautomatic@gumrf.ru)

#### INFORMATION ABOUT THE AUTHORS

##### **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](mailto:chertkov51@mail.ru),  
[kaf\\_electricautomatic@gumrf.ru](mailto:kaf_electricautomatic@gumrf.ru)

##### **Kask, Yaroslav N. —**

PhD  
Admiral Makarov State University of Maritime  
and Inland Shipping  
5/7 Dvinskaya Str., St. Petersburg, 198035,  
Russian Federation  
e-mail: [rgam2010@yandex.ru](mailto:rgam2010@yandex.ru), [kaf\\_electricautomatic@gumrf.ru](mailto: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: [rgam2010@yandex.ru](mailto:rgam2010@yandex.ru), [kaf\\_electricautomatic@gumrf.ru](mailto:kaf_electricautomatic@gumrf.ru)

Статья поступила в редакцию 12 мая 2022 г.

Received: May 12, 2022.