ВЕСТНИК
ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
МОРСКОГО И РЕЧНОГО ФЛОТА ИМЕНИ АДМИРАЛА С. О. МАКАРОВА

УДК 621.396

А. А. Чертков,
А. А. Вардомская,
А. А. Дмитриев

РЕКУРСИВНЫЙ МЕТОД ОПТИМИЗАЦИИ ЛОГИСТИЧЕСКИХ ПУТЕЙ СРЕДСТВАМИ MATLAB

Методы и средства оптимизации логистических систем и их элементов в условиях развития институтов рынка играют определяющую роль в создании новых механизмов повышения эффективности и качества функционирования объектов водного транспорта и транспортной инфраструктуры в целом. Современная теория и численные методы оптимизации являются инструментом, позволяющим принимать научно обоснованные решения при управлении технологическими системами и транспортными средствами с использованием моделей и компьютерных технологий, быстро и с малыми затратами ресурсов и времени адаптироваться при изменениях объемов и конъюнктуры рынка в соответствующих секторах предоставления транспортных услуг. В работе рассматривается рекурсивный метод оптимизации трафика логистических систем с использованием компьютерных технологий, реализованных средствами MatLab с целью повышения уровня и организации управления транспортно-логистическими системами с коррекцией по состоянию. Рекурсивный метод, в отличие от существующих решений, позволяет автоматизировать выбор кратчайшего пути доставки грузов в транспортной сети со сложной топологией, осуществляемый на основе взвешенного графа, с учетом заданного критерия качества. В работе предложены алгоритм и рекурсивная процедура оптимизации для реализации метода и сокращения времени выполнения транспортной работы. Рассмотрен конкретный пример определения кратчайших путей для подтверждения корректности предложенных технических решений.

Ключевые слова: автоматизация, алгоритм, логистика, ориентированный граф, критический путь, целевой функционал, модель, критерий качества.

РЫНОЧНЫЕ экономические условия требуют повышения эффективности функционирования всех субъектов хозяйственной деятельности, в том числе предприятий, работающих в области дистрибуции товаров и осуществляющих доставку заказов своим клиентам. Как показывают проводимые исследования, совокупные логистические издержки на доставку заказов составляют до 50 % от общей суммы затрат таких предприятий [1]. Одним из путей сокращения логистических затрат является формирование оптимальных транспортных маршрутов.

При оперативном управлении процессом доставки грузов важен показатель времени, минимизация которого связана с поиском маршрута, имеющего самую короткую длину из всех возможных. Причем, для удобства построения маршрутов перевозок необходимо предусмотреть возможность составления программой не одного, а нескольких маршрутов доставки на случай, если отдельные маршруты будут перегружены. Аналитические методы решения неприменимы к решению задачи оптимального формирования маршрутов в силу ее большой размерности и высокой вычислительной сложности. В то же время, математический аппарат динамической оптимизации, основанный на методологии пошаговой оптимизации, может быть использован при нахождении кратчайших расстояний, например, на географической карте, представленной в виде сети.

Решение задачи по определению кратчайших расстояний между пунктами отправления и пунктами получения продукции по существующей транспортной сети является исходным этапом при решении таких экономических задач как оптимальное закрепление потребителей за поставщиками, повышение производительности транспорта за счет сокращения непроизводительного пробега и др. [1] – [4]. Основной моделью большинства логистических схем доставки грузов является взвешенный граф (как ориентированный, так и неориентированный), с помощью которого определяется кратчайший путь как наиболее эффективный с точки зрения заданного критерия качества.

Теория графов началась с задачи про семь кенигсбергских мостов. Вопрос задачи звучит следующим образом: можно ли пройти по всем мостам города Кенигсберга и при этом вернуться в исходную точку так, чтобы по каждому мосту пройти только один раз. Знаменитый петербургский математик швейцарского происхождения Леонард Эйлер блестяще решил эту задачу при помощи теории графов.

В середине 1950-х гг. в теории графов сформировалась потоковая задача поиска искомого пути на сетке. Соответствующая методология получила название «Сетевое планирование и управление» (СПУ) [1], [5] – [6]. В терминологии теории графов взвешенным графом является конечный связной граф (как ориентированный, так и неориентированный) без контуров, в котором каждому его ребру присвоено некоторое число, называемое длинной ребра.

Наиболее эффективный алгоритм решения задачи о кратчайшем пути первоначально дал Дейкстра [5], [7] – [9]. Алгоритм голландского ученого Эдсгера Дейкстры находит все кратчайшие пути из одной изначально заданной вершины графа до всех остальных. С его помощью, при наличии всей необходимой информации, можно, например, узнать какую последовательность дорог лучше использовать, чтобы добраться из одного города до каждого из многих других, или в какие страны выгодней экспортировать нефть и т. п. Минусом данного метода является невозможность обработки графов, в которых имеются ребра с отрицательным весом, т. е. если, например, некая система предусматривает убыточные для фирмы маршруты, то для работы с ней следует воспользоваться отличным от алгоритма Дейкстры методом.

В общем случае этот метод основан на приписывании вершинам временных пометок, причем пометка вершины дает верхнюю границу длины пути от некоторой вершины s к рассматриваемой вершине. Эти пометки постепенно уменьшаются с помощью некоторой итерационной процедуры, и на каждом шаге итерации только одна из временных пометок становится постоянной. Последнее указывает на то, что пометка уже не является верхней границей, а дает точную длину кратчайшего пути от t к рассматриваемой вершине. Рассмотрим подробнее этот алгоритм.

Для программной реализации алгоритма понадобится два массива: логический $\boldsymbol{L}$ — для хранения текущей информации о пройденных вершинах и численный $\boldsymbol{LP}$, в который будут заноситься найденные кратчайшие пути. Итак, имеется граф $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{C})$ со взвешенными дугами, в котором $\boldsymbol{X} = \{x_1, x_2, \ldots, x_n\}$ — множество вершин, а C — матрица весов дуг (ребер). Каждая из вершин, входящих в множество $\boldsymbol{X}$, изначально отмечена как непосещенная, т. е. элементам массива $\boldsymbol{X}$ присвоено значение *zero*.

Пусть $\boldsymbol{L}(\boldsymbol{X})$ представляет собой вектор, координатами которого являются текущие расстояния $L(x_i)$ от стартовой вершины до каждой из вершин x_i. Поскольку самые выгодные пути только предстоит найти, в каждый элемент $L(x_i)$ вектора записывается такое число, которое заведомо больше любого потенциального пути. В качестве исходного пункта выбирается вершина s и ей приписывается нулевой путь $L(s) = 0$, так как нет ребра из s в s (метод не предусматривает петель). Далее находятся все соседние с s вершины, в которые входит ребро из s (пусть таковыми будут t и u) и поочередно вычисляются длины маршрута из s в каждую из них:

– $L(t) = L[s] + c(s, t)$ ребра;

– $L(u) = L[s] + c(s, u)$ ребра.

Однако вполне вероятно, что в ту или иную вершину из s существует несколько путей, поэтому цену пути в такую вершину в массиве $L(t)$ придется пересматривать, тогда наибольшее (неоптимальное) значение игнорируется, а наименьшее ставится в соответствие вершине.

После обработки смежных с s вершин она помечается как посещенная, и активной становится та вершина, путь из s в которую минимален. Допустим, путь из s в u короче, чем из s в t, следовательно, вершина u становится активной, и описанным образом исследуются ее соседи, за исключением вершины s. Далее, u помечается как пройденная, активной становится вершина t, и вся процедура повторяется для нее. Алгоритм продолжается до тех пор, пока все доступные из s вершины не будут исследованы.

Теперь на конкретном графе проследим работу алгоритма, найдем все кратчайшие пути между истоковой и всеми остальными вершинами. Размер (количество ребер) изображенного далее графа равен 5 (|E| = 5), а порядок (количество вершин) — 4 (|V| = 4). Это взвешенный граф, каждому из его ребер поставлено в соответствие некоторое числовое значение, поэтому ценность маршрута необязательно определяется числом ребер, лежащих между парой вершин.

Математическую модель сетевой задачи построим на примере сетевого графика (рис. 1), заданного в виде ориентированного графа, который состоит из 9 узлов (событий) и 13 дуг (операций). Дан граф $\boldsymbol{G} = (\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{C})$ с взвешенными дугами, пример которого показан на рис. 1. Обозначим $L(x_i)$ пометку вершины x_i. Веса дуг (или ребер) даны матрицей весов (табл. 1).

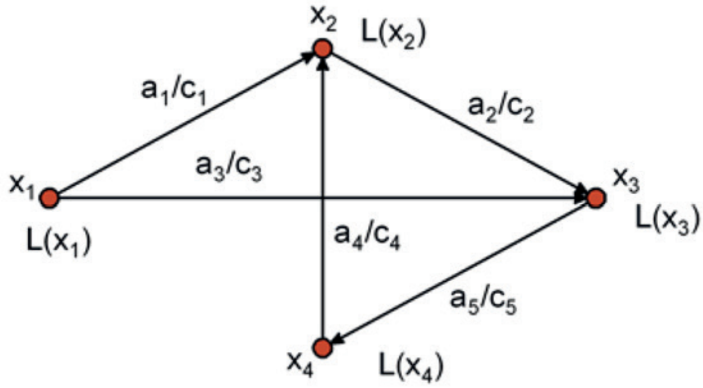

Рис. 1. Граф со взвешенными дугами

Таблица 1

Матрица весов расстояний

	x_1	x_2	x_3	x_4
x_1		c_1	c_3	
x_2			c_2	
x_3				c_5
x_4		c_4		

Рассмотрим алгоритм нахождения кратчайшего пути от вершины s к вершине t графа и более общий случай: от вершины s ко всем вершинам графа.

Шаг 1. Присвоение начальных значений вершинам графа. Положить $L(s) = 0$ и считать эту пометку постоянной. Для всех вершин $x_i \neq s$ положить $L(x_i) = \infty$ и считать эти пометки временными. За текущую рассматриваемую вершину с постоянной пометкой взять вершину p, т. е. положить $p = s$.

Шаг 2. Обновление пометок. Для вершин, входящих в прямое отображение вершины p, т. е. для всех x_i, принадлежащих $\Gamma(p)$, пометки которых временные, изменить пометки в соответствии с выражением

$$L(x_i) \leftarrow \min[L(x_i), L(p) + c(p, x_i)]. \quad (1)$$

Шаг 3. Превращение временной пометки в постоянную. Среди всех вершин с временными пометками найти такую, для которой

$$L(x^*_i) = \min[L(x_i)]. \quad (2)$$

Считать пометку вершины x^*_i постоянной и положить $p = x^*_i$.

Шаг 4. Обновление пометок для новой отображаемой вершины. В качестве таковой принять вершину x^*_i, т. е. $p = x^*_i$, а $L(p)$ является длиной кратчайшего пути от s к x^*_i.

Шаг 5. Просмотр вершин графа на наличие у них постоянных пометок:

– если все вершины отмечены постоянными метками, эти метки дают длины кратчайших путей;

– если некоторые метки являются временными, следует перейти к шагу 2.

Как только длины кратчайших путей от вершины s будут найдены, сами пути можно получить с помощью рекурсивной процедуры. Так как вершина x^*_i непосредственно предшествует вершине x_i в кратчайшем пути от s к x_i, то для любой вершины x_i соответствующую вершину x^*_i можно найти как одну из оставшихся вершин, для которой

$$L(x^*_i) + c(x^*_i, x_i) = L(x_i). \qquad (3)$$

Если кратчайший путь от s до любой вершины x_i является единственным, то дуги (x^*_i, x_i) этого кратчайшего пути образуют ориентированное дерево с корнем s. Если существует несколько кратчайших путей от s к какой-либо другой вершине, то при некоторой фиксированной вершине x^*_i соотношение (1) будет выполняться для более, чем одной вершины x_i. В этом случае, выбор может быть либо произвольным (если нужен какой-то один кратчайший путь между s и x_i), либо таким, что рассматриваются все дуги (x^*_2, x_2), входящие в какой-либо из кратчайших путей. При этом совокупность всех таких дуг образует не ориентированное дерево, а общий граф, называемый базой относительно s [10] – [12].

Пример. Рассмотрим граф смешанного типа, изображенный на рис. 2, где каждое неориентированное ребро рассматривается как пара противоположно направленных дуг равного веса.

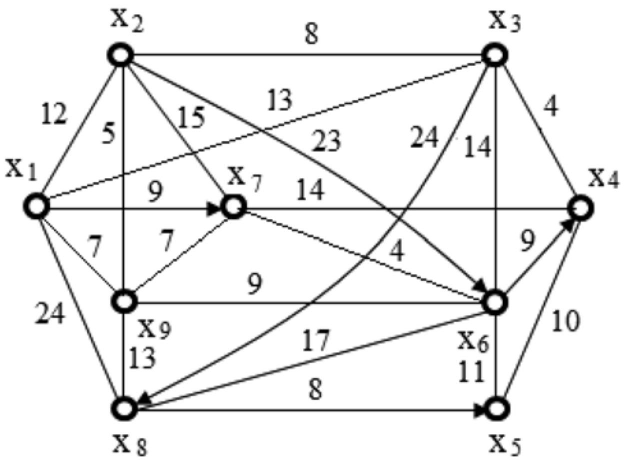

Рис. 2. Исходный взвешенный граф смешанного типа

Матрица весов приведена в табл. 2. Требуется найти все кратчайшие пути от вершины x_1 ко всем остальным вершинам.

Таблица 2

Матрица весов дуг исходного графа

$C =$

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_1		12	13				9	24	7
x_2	12		8			23	15		5
x_3	13	8		4		14		24	
x_4			4		10		14		
x_5				10		11			
x_6			14	9	11		4	17	9
x_7		15		14		4			7
x_8	24				8	17			13
x_9	7	5				9	7	13	

Постоянные пометки будем помечать знаком +.

Присвоим $L(x_1) = 0$, $L(x_i) = 100$ для всех x_i, кроме x_1. Положим $p = x_1$. Эта вершина является первой отображаемой вершиной, с которой в соответствии с выражениями (1) – (2) согласно описанной процедуре производится обновление временных и постоянных пометок для каждой из оставшихся вершин графа (табл. 3).

Таблица 3

Итерационная процедура по обновлению пометок вершин графа

№ итерации	Отображаемая вершина $\mathbf{\Gamma}(x_i^*) = \mathbf{\Gamma}(p)$	Массивы временных пометок вершин графа	Кратчайший путь $L(p)$
1	$\mathbf{\Gamma}(x_1) = \{x_2, x_3, x_7, x_8, x_9\}$	$L(x_2) = \min[L(x_2), L(x_1) + c(x_1, x_2)] = \min[100, 0 + 12] = 12$ $L(x_3) = \min[L(x_3), L(x_1) + c(x_1, x_3)] = \min[100, 0 + 13] = 13$ $L(x_4) = 100$ $L(x_5) = 100$ $L(x_6) = 100$ $L(x_7) = \min[L(x_7), L(x_1) + c(x_1, x_7)] = \min[100, 0 + 9] = 9$ $L(x_8) = \min[L(x_8), L(x_1) + c(x_1, x_8)] = \min[100, 0 + 24] = 24$ $L(x_9) = \min[L(x_9), L(x_1) + c(x_1, x_9)] = \min[100, 0 + 7] = 7$	$L(x_9) = 7^+$
2	$\mathbf{\Gamma}(x_9) = \{x_1, x_2, x_6, x_7, x_8\}$	$L(x_2) = \min\ [12{,}7 + 5] = 12$ $L(x_3) = 13$ $L(x_4) = 100$ $L(x_5) = 100$ $L(x_6) = \min\ [100, 7 + 9] = 16$ $L(x_7) = \min\ [9, 7 + 7] = 9$ $L(x_8) = \min[24, 7 + 13] = 20$	$L(x_7) = 9^+$
3	$\mathbf{\Gamma}(x_7) = \{x_2, x_4, x_6, x_9\}$	$L(x_2) = \min[12, 9 + 15] = 12$ $L(x_3) = 13$ $L(x_4) = \min[100, 9 + 14] = 23$ $L(x_5) = 100$ $L(x_6) = \min[16, 9 + 4] =13$ $L(x_8) = 20$	$L(x_2) = 12^+$
4	$\mathbf{\Gamma}(x_2) = \{x_1, x_3, x_6, x_7, x_9\}$	$L(x_3) = \min[13, 12 + 8] = 13$ $L(x_4) = 23$ $L(x_5) = 100$ $L(x_6) = \min\ [13, 12 + 23] = 13$ $L(x_8) = 20$	$L(x_3) = 13^+$
5	$\mathbf{\Gamma}(x_3) = \{x_1, x_2, x_4, x_{6,} x_8\}$	$L(x_4) = \min[23, 13 + 4] = 17$ $L(x_5) = 100$ $L(x_6) = \min\ [13, 13 + 14] = 13$ $L(x_8) = \min[20, 13 + 24] = 20$	$L(x_6) = 13^+$
6	$\mathbf{\Gamma}(x_6) = \{x_3, x_4, x_5, x_7, x_{8,} x_9\}$	$L(x_4) = [17, 13 + 9] = 17$ $L(x_5) = \min[100, 13 + 11] = 24$ $L(x_8) = \min\ [20, 13 + 17] = 20$	$L(x_4) = 17^+$
7	$\mathbf{\Gamma}(x_4) = \{\mathrm{x}_3, x_{5,} x_6, x_7\}$	$L(x_5) = \min\ [24, 17 + 10] = 24$ $L(x_8) = 20$	$L(x_8) = 20^+$
8	$\mathbf{\Gamma}(x_8) = \{x_1, x_5, x_6, x_7, x_9\}$	$L(x_5) = \min[24, 20 + 8] = 24$	$L(x_5) = 24^+$

По окончании итерационной процедуры все вершины графа имеют постоянные отметки (рис. 3). Таким образом, алгоритм завершен.

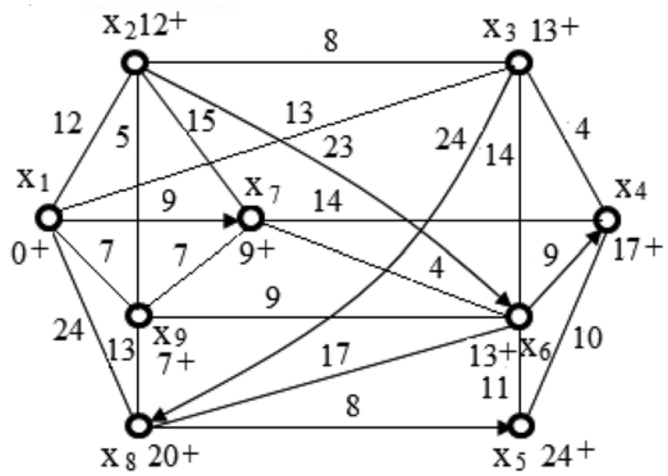

Рис. 3. Граф с постоянными отметками во всех вершинах

Для нахождения кратчайшего пути между вершинами, например x_2 и начальной x_1, последовательно используем соотношение (3)

$$L(x_2') + c\,(x_2', x_2) = L\,(x_2) = 12,$$

где вершина x_2' — это вершина, непосредственно предшествующая x_2 в кратчайшем пути от x_1 к x_2 .

Одной из таких вершин является вершина x_9. Далее соотношение (3) применяем второй раз

$$L(x_7') + \mathrm{c}(x_7', x_7) = L(x_7) = 7.$$

Непосредственно предшествующей вершине x_9 является вершина x_1. Поэтому кратчайший путь от x_1 к x_2 есть (x_1, x_9, x_2). Вершина x_1, называемая базой и дающая все кратчайшие пути от x_1, представляет дерево.

Реализация в среде MatLab. Для проведения расчетов кратчайших расстояний от стартовой вершины x_1 до каждой из вершин $x_i \neq x_1$ (i =1, …, n) по оптимизационной схеме на основе рассмотренного алгоритма из восьми итераций разработана программа в кодах MatLAB (License Attributes: version R2010b,license number:358625), позволяющая получить численные значения кратчайших путей в исходном графе.

В итерационной процедуре используются следующие обозначения:

C — матрица весов дуг (ребер), значения элементов в которой выбраны из табл. 2. Отсутствующие значения в элементах матрицы заполнены числами 100, заведомо большими любого потенциального пути;

L — одномерный массив временных пометок каждой из вершин графа;

LP — одномерный массив кратчайших расстояний (постоянных пометок) каждой из вершин графа;

P — одномерный массив номеров вершин графа, отмеченных постоянными пометками;

I — номер итерации.

При реализации алгоритма разработана программа в кодах MatLab и получены следующие результаты (табл. 4).

Таблица 4

Основные результаты реализации рекурсивной процедуры

№ итерации (I)	Массивы временных пометок (***L***)	Массивы номеров вершин (***P***)	Массивы постоянных пометок (***LP***)
1	100 12 13 100 100 100 9 24 7	1 0 0 0 0 0 0 0 9	0 0 0 0 0 0 0 0 7
2	100 12 13 100 100 16 9 20 100	1 0 0 0 0 0 7 0 9	0 0 0 0 0 0 9 0 7
3	100 12 13 23 100 13 100 20 100	1 2 0 0 0 0 7 0 9	0 12 0 0 0 0 9 0 7

Таблица 4
(*Окончание*)

4	100 100 13 23 100 13 100 20 100	1 2 3 0 0 070 9	0 12 13 0 0 0 9 07
5	100 100 100 17 100 13 100 20 100	1 2 3 0 0 6 7 0 9	0 12 130 0 13 9 0 7
6	100 100 100 17 24 100 100 20 100	1 2 3 4 0 6 7 0 9	0 12 13 17 0 139 0 7
7	100 100 100 100 24 100 100 20 100	1 2 3 4 0 6 7 8 9	0 1213 170 13 9 20 7
8	100 100 100 100 24 100 100 100 100	12 3 4 5 6 7 8 9	0 12 13 17 24 13 9 20 7

Анализ результата. Полученные программным способом значения кратчайшихпутей от стартовой (первой) вершины до всех остальных вершин взвешенного графа представлены в табл. 5.

Таблица 5

Кратчайшие расстояния до вершин графа

Номера вершин	1	2	3	4	5	6	7	8	9
Кратчайшие расстояния	0	12	13	17	24	13	9	20	7

Выводы

1. Разработана универсальная компьютерная модель, позволяющая автоматизировать операции определения кратчайших маршрутов движения материальных грузопотоков в логистических системах с двусторонним движением (как в прямом, так и в обратном направлении) грузопотоков.

2. Предложена итерационная модель на основе рекурсивной процедуры определения последовательностей ребер, из которых состоит рассматриваемый кратчайший путь. Для выполнения технологических расчетов на основе рекурсивных уравнений вводятся переменные состояния в виде временных и постоянных отметок вершин графа, что позволило использовать для расчетов функции инструментарий Optimization Toolbox среды MatLab.

3. Корректность и эффективность предложенной процедуры подтверждена на конкретном примере автоматизации расчета кратчайших путей с полным анализом возможных вариантов с размерностью вектора состояния $n = 9$.

4. Приведенная модель расчета, реализованная в кодах MatLab, может быть использована не только для автоматизации логистических систем на водном транспорте, но и в телекоммуникационных системах пакетной передачи данных [14], в системах оптимизации маршрутных карт российских регионов [13], а также в других системах автоматизации и управления высокой размерности [14], [15] различного назначения.

СПИСОК ЛИТЕРАТУРЫ

1. *Dijkstra E. W.* A note on two problems in connexion with graphs / E. W. Dijkstra // Numerische Mathematik. — 1959. — Vol. 1. — Pp. 269–271.

2. *Сахаров В. В.* Модели и алгоритмы оптимизации технологических процессов на объектах водного транспорта в среде MatLab: монография / В. В. Сахаров, А. А. Кузьмин, А. А. Чертков. — СПб.: Изд-во ГУМРФ им. адм. С. О. Макарова, 2015. — 436 с.

3. *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.

4. *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.

5. *Дейкстра Э.* Дисциплина программирования / Э. Дейкстра. — М.: Мир, 1978. — 275 с.

6. Алгоритмы: построение и анализ: пер. с англ. / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн. — 2-е изд. — М.: ИД «Вильямс», 2010. — 1296 с.

7. *Левитин А. В.* Алгоритмы: введение в разработку и анализ / А. В. Левитин. — М.: ИД «Вильямс», 2006. — 565 с.

8. *Russell J.* Алгоритм Дейкстры / J. Russell, R. Cohn. — Изд-во VSD, 2012. — 112 с.

9. *Охорзин В. А.* Оптимизация экономических систем / В. А. Охорзин. — М.: Финансы и статистика, 2005. — 144 с.

10. *Романовский И. В.* Дискретный анализ / И. В. Романовский. — 4-е изд., испр. и доп. — СПб.: Невский Диалект, БХВ-Петербург, 2008. — 336 с.

11. Mathematical programming techniques in water network optimization / C. D'Ambrosio, A. Lodi, S. Wiese, C. Bragalli // European Journal of Operational Research. — 2015. — Vol. 243. — № 3. — Pp. 774–788. DOI:10.1016/J.EJOR.2014.12.039.

12. *Reich D.* A linear programming approach for linear programs with probabilistic constraints / D. Reich // European Journal of Operational Research. — 2013. — Vol. 230. — № 3. — Pp. 487–494. DOI:10.1016/J.EJOR.2013.04.049.

13. *Соколов А. В.* Разработка технологических дорожных карт / А. В. Соколов, О. И. Карасев // Российские нанотехнологии. — 2009. — Т. 4. — № 3–4. — С. 16–17.

14. *Оре О.* Теория графов / О. Оре. — М.: Изд-во Либроком, 2009. — 354 с.

15. *Бояринцева Т. И.* Теория графов: методические указания к выполнению домашнего задания по курсу «Дискретная математика» / Т. И. Бояринцева, А. А. Мастихина. — М.: Изд-во МГТУ им. Н. Э. Баумана, 2014. — 40 с.

A RECURSIVE METHOD OF OPTIMIZATION OF THE LOGISTIC WAYS BY MEANS OF MATLAB

Methods and tools for optimization the logistic systems and their elements in terms of market institutions development , play the main role in the creation of new mechanisms to improve the efficiency and quality of the objects of water transport and transport infrastructure in general. Modern theory and numerical methods are means to make scientifically sound decisions in the management of technological systems and vehicles using models and computer technologies, fast and low cost resources and time to adjust when changes in the volume and market conditions in the relevant sectors, the provision of transport services. Is a recursive method to optimize traffic logistics systems with the use of computer technologies by means of MATLAB in order to increase the level of organization and management of transportation and logistics systems with correction by status. A recursive method, unlike the existing, you can automate the shortest path is the delivery of the goods in the transport network with complex topology, implemented through a weighted graph, given the quality criterion As a result of the proposed algorithm and recursive procedure for method realization optimization and reducing the execution time of the transport operation. Concrete example for determine the shortest path to confirm the correctness of the proposed technical solution is considered.

Keywords: automation, algorithm, logistic, directed graph, critical path, the target functionality, model, quality criterion.

REFERENCES

1. Dijkstra, E. W. “A note on two problems in connexion with graphs.” *Numerische Mathematik* 1 (1959): 269-271.

2. Saharov, V. V., A. A. Kuzmin, and A. A. Chertkov. *Modeli i algoritmy optimizacii tehnologicheskih processov na obektah vodnogo transporta v srede MATLAB: monografija*. — SPb.: GUMRF imeni admirala S.O. Makarova, 2015.

3. Chertkov, A. A., A. A. Vardomskaya, and A. A. 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.

4. 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.

5. Dejkstra, Je. *Disciplina programmirovanija*. M.: Mir, 1978.

6. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. *Algoritmy: postroenie i analiz.* – 2-e izd. Per. s angl. M.: Izdatelskij dom «Vilijams», 2010.

7. Levitin, A. V. *Algoritmy: vvedenie v razrabotku i analiz*. M.: Izdatelskij dom «Viljams», 2006.

8. Russell, Jesse, and Ronald Cohn. *Algoritm Dejkstry*. Izd-vo VSD, 2012.

9. Ohorzin, V. A. *Optimizacija jekonomicheskih sistem*. M.: Finansy i statistika, 2005.

10. Romanovsky, I. V. *Diskretnyj analiz. 4-e izd., ispr. i dop.* SPb.: Nevskij Dialekt, BHV-Peterburg, 2008.

11. D'Ambrosio, C., A. Lodi, S. Wiese, and C. Bragalli. "Mathematical programming techniques in water network optimization." *European Journal of Operational Research* 243.3 (2015): 774-788. DOI:10.1016/J.EJOR.2014.12.039.

12. Reich, Daniel. "A linear programming approach for linear programs with probabilistic constraints." *European Journal of Operational Research* 230.3 (2013): 487-494. DOI:10.1016/J.EJOR.2013.04.049.

13. Sokolov, A. V., and O. I. Karasev. "Razrabotka tehnologicheskih dorozhnyh kart." *Rossijskie nanotehnologii* 4.3-4 (2009): 16-17.

14. Ore, O. *Teorija grafov*. Izdatelstvo: Librokom, 2009.

15. Bojarinceva, T. I., and A. A. Mastihina. *Teorija grafov: metodicheskie ukazanija k vypolneniju domashnego zadanija po kursu «Diskretnaja matematika»*. M.: MGTU im. N. Je. Baumana, 2014.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Чертков Александр Александрович —
кандидат технических наук, доцент.
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова».
chertkov51@mail.ru
Вардомская Анна Александровна — аспирант.
Научный руководитель:
Барышников Сергей Олегович —
доктор технических наук, профессор, ректор
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
kaf_osnipr@gumrf.ru
Дмитриев Александр Александрович — аспирант.
Научный руководитель:
Сахаров Владимир Васильевич —
доктор технических наук, профессор.
ФГБОУ ВО «ГУМРФ имени
адмирала С. О. Макарова»
kaf_osnipr@gumrf.ru

INFORMATION ABOUT THE AUTHORS

Chertkov Alexandr Alexandrovich —
PhD, associate professor.
Admiral Makarov State University
of Maritime and Inland Shipping
chertkov51@mail.ru
Vardomskaya Anna Alexandrovna — postgraduate.
Supervisor:
Baryshnikov Sergej Olegovich —
Dr. of Technical science, professor, rector
Admiral Makarov State University
of Maritime and Inland Shipping
kaf_osnipr@gumrf.ru
Dmitriev Alexandr Alexandrovich — postgraduate.
Supervisor:
Saharov Vladimir Vasilievich —
Dr. of Technical science, professor
Admiral Makarov State University
of Maritime and Inland Shipping
kaf_osnipr@gumrf.ru

Статья поступила в редакцию 7 октября 2015.

УДК 681.3.015

Н. Ю. Барышникова

РАЗРАБОТКА МЕХАНИЗМА ПРЕОБРАЗОВАНИЯ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ В ИНФОРМАЦИОННОЙ СИСТЕМЕ

Современные достижения в области информационных технологий проявляются в создании и развитии интеллектуальных информационных систем, обеспечивающих хранение и изменение необходимых пользователю данных, а также их получение по требованию. В статье описан метод реализации формального механизма преобразования запроса к данным на проблемно-ориентированном подмножестве естественного языка в информационной системе. За основу взята система электронного тестирования для подготовки морских специалистов. Разработан алгоритм функционирования механизма, позво-