УДК 519.681

В. В. КАРПЕНКО

МЕТОД РЕШЕНИЯ ЗАДАЧИ МАРШРУТИЗАЦИИ В РЕАЛЬНОМ ВРЕМЕНИ

Сформульовано задачу забезпечення доставки товару від виробника до випадкової множини споживачів. Розглянуто методи знаходження найкоротших маршрутів. Встановлено, що для задач реальної розмірності ці методи не забезпечують можливості отримання швидкого рішення. Запропоновано метод знаходження найкоротшого маршруту, заснований на використанні спеціальної операції над числовими матрицями, елементи яких складаються з довжин шляхів між сусідніми пунктами на маршруті. Метод дозволяє отримати швидке наближене рішення задачі, близьке до оптимального.

Ключові слова: задача маршрутизації, управління перевезеннями, транспортна задача, лінійне програмування, пошук найкоротшого маршруту, задача комівояжера, знаходження маршруту у реальному часі, ефективність маршрутизації.

Сформулирована задача обеспечения доставки продукта от производителя к случайному множеству потребителей. Рассмотрены методы отыскания кратчайших маршрутов. Установлено, что для задачи реальной размерности эти методы не обеспечивают возможности получения быстрого решения. Предложен метод отыскания кратчайшего маршрута, основанный на использовании специальной операции над числовыми матрицами, элементы которых – длины путей между соседними пунктами на маршруте. Метод позволяет получить быстрое приближенное решение задачи, близкое к оптимальному.

Ключевые слова: задача маршрутизации, управление перевозками, транспортная задача, линейное программирование, отыскание кратчайшего маршрута, задача коммивояжера, нахождение маршрута в реальном времени, эффективность маршрутизации.

The problem of ensuring the delivery of the product from the producer to the random set of consumers. An important characteristic of the possible technologies to solve this problem is lead time, which depends on the length of the route from producer to consumer. The known methods for finding the shortest routes based on real city roads. It was found that these methods do not provide opportunities for a quick solution to the real dimension of the problem. A method for finding the shortest route based on the use of special operations on numerical matrices whose elements – the path lengths between adjacent points on the route. The method provides a quick approximate solution is close to optimal. An example explaining the essence and meaning of the proposed computational procedure for solving the problem.

Keywords: routing problem, transportation management, transportation problem, linear programming, finding the shortest route, the traveling salesman problem, finding a route in real–time routing efficiency.

Введение. В современных условиях хозяйствования важное значение приобретает задача управления перевозками. Для крупных фирм, располагающих разветвленной сетью филиалов, складов, с многоэлементной системой потребителей появляется потребность в решении транспортной задачи (по схеме «от многих к многим»). Аналогичная задача возникает и при управлении транспортировками для производственных предприятий, имеющих многономенклатурную, территориально распределенную систему разнотипных подразделений, обеспечивающих реализацию технологического процесса производства. Не менее важной для практики является задача управления перевозками в системе «от одного к многим».

Анализ основных достижений и литературы. Следует отметить важную особенность реальных задач управления перевозками от классических транспортных задач [1, 2]. Постановка традиционных транспортных задач предусматривает отыскание распределения объема перевозок от совокупности поставщиков к совокупности потребителей. При этом совершенно не рассматривается вопрос о том, по какому маршруту должен транспортироваться груз от конкретного поставщика к конкретному потребителю. Вместе с тем, во многих случаях эта проблема выходит на первый план. Именно так дело обстоит в задаче доставки какого–либо продукта массового потребления от пункта производства этого продукта к множеству его заказчиков. При этом, ввиду территориального рассредоточения множества независимых потребителей возникает необходимость решения соответствующего множества разных задач маршрутизации. Каждая конкретная задача этого типа формулируется следующим образом. Заданы пункт производства и пункт потребления, а также совокупность промежуточных пунктов, через которые может проходить маршрут от начального пункта к конечному. Кроме того, если из промежуточного пункта i возможен непосредственный переезд в промежуточный пункт j, то задано расстояние между ними r_{ij}, необязательно равное r_{ij}, $i = 1,2,...,N$, $j = 1,2,...,N$. Далее вводится набор индикаторов

$$x_{i_1 i_2} = \begin{cases} 1, & \text{если маршрут между начальным} \\ & \text{и конечным пунктами содержит} \\ & \text{участок между пунктами } i_1, i_2, \\ 0, & \text{в противном случае.} \end{cases}$$

Этот набор образует матрицу $\boldsymbol{X} = (x_{i_1 i_2})$.

Тогда общая длина маршрута будет равна

$$L(\boldsymbol{X}) = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} r_{i_1 i_2} x_{i_1 i_2}. \tag{1}$$

Задача состоит в отыскании набора, минимизирующего целевую функцию (1). Этот набор должен удовлетворять следующей системе ограничений

$$\sum_{i_2=1}^{N} x_{i_1 i_2} \le 1, \quad i_1 = 1,2,...,N, \tag{2}$$

$$\sum_{i_1=1}^{N} x_{i_1 i_2} \le 1, \quad i_2 = 1,2,...,N, \tag{3}$$

$$u_{i_1} - u_{i_2} + n x_{i_1 i_2} \le n - 1, \quad (4)$$

$$i_1 = 1, 2, ..., N, \quad i_2 = 1, 2, ..., N.$$

Система ограничений (2) обеспечивает построение маршрута, в котором выезд из каждого пункта i_1 осуществляется не более чем один раз. Ограничения (3) обеспечивают въезд в каждый из пунктов не более одного раза. Наконец, система ограничений (4) устраняет возможность появления циклов и построение связного маршрута.

Эта задача напоминает задачу коммивояжера [3, 4]. Принципиальное отличие состоит в том, что в данной задаче искомый маршрут может, но не обязательно должен проходить через все промежуточные пункты N. Полученная задача относится к классу целочисленных задач линейного программирования с булевыми переменными.

Известные методы решения булевых задач линейного программирования. В теории линейного программирования разработаны общие методы решения целочисленных задач – методы отсечения и эффективного перебора [5–7].

Методы отсечения основаны на возможности построения в пространстве переменных гиперплоскости, которая отсекает нецелочисленные решения задачи от целочисленных [5]. При этом происходит добавление к системе ограничений задачи дополнительного ограничения. Новая задача решается без учета ограничения на целочисленность. Если полученное решение оказывается нецелочисленным, то строится новая отсекающая гиперплоскость. Процесс продолжается до получения целочисленного решения. Основной недостаток метода – непрогнозируемая продолжительность времени решения задачи, которая может оказаться неприемлемо большой.

Другой подход к решению целочисленных задач использует разнообразные правила эффективного сокращения множества целочисленных планов. Число планов любой задачи линейного программирования конечно. Поэтому конечным является и число целочисленных планов, которые, в принципе, можно последовательно перебрать и выбрать наилучший. Однако, в реальных задачах число планов велико настолько, что их перебор нереализуем. Методы эффективного перебора позволяют осуществить перебор лишь некоторых перспективных вариантов, отсеивая при этом очень большое число планов, которые заведомо не являются оптимальными. Центральная проблема здесь – построение эффективного решающего правила, позволяющего на основании имеющейся к текущему шагу информации отсевать возможно большее число бесперспективных вариантов. Известны несколько методов, реализующих идею эффективного перебора применительно к задаче маршрутизации. К их числу относятся: метод последовательного конструирования, анализа и отбора вариантов [6], метод Балаша [8] и метод ветвей и границ [4]. Технология построения соответствующих алгоритмов однотипна. Множество вариантов описывается в виде дерева, каждой ветви которого соответствует конкретный план задачи. Решение начинается с выбора некоторой конкретной ветви, для которой вычисляется значение целевой функции – длина маршрута. Полученную длину называют эталонной. Затем выбирается другая ветвь и начинается движение вдоль неё, начиная от вершины. Пусть сделано несколько шагов, определяющих начальный участок маршрута, и вычислена его длина. Далее предполагается, что имеется возможность расчета оптимистической оценки длины оставшейся части маршрута и, таким образом, оптимистической оценки длины всего маршрута. Полученная оценка сравнивается с эталонной. Понятно, что в случае, если оптимистическая длина хуже, чем длина эталонной ветви, то рассматриваемый начальный участок ветви порождает куст бесперспективных ветвей и его можно не продолжать. Последовательное применение этой технологии позволяет эффективно сокращать число переборов вариантов. Перечисленные методы отличаются друг от друга способом расчета оптимистических оценок. Общий недостаток методов эффективного перебора состоит в экспоненциально быстром росте числа вариантов маршрутов в зависимости от возможного числа промежуточных пунктов. Кроме того, все эти методы плохо учитывают специфику задач маршрутизации. В связи с этим отметим, что актуальность и практическая потребность в решении задачи маршрутизации привели к разработке методов, ориентированных на существо и особенности именно этой задачи. Наиболее успешные из них используют идеи «коллективного интеллекта». Построенные на этой идее методы отыскания кратчайшего маршрута реализуют способы поиска, имитирующие поведение «коллектива» живых организмов. Системы коллективного интеллекта состоят из множества агентов, взаимодействующих между собой и внешней средой. Анализ совместной деятельности множества локальных агентов колоний позволяет построить совокупность методов эффективного поиска рационального маршрута, которые получили название «роевые алгоритмы» [9]. К их числу относятся: муравьиный алгоритм [10, 11], алгоритм роя частиц [12], метод дифференциальной эволюции [13].

Основные принципы работы таких алгоритмов рассмотрим на примере муравьиного алгоритма. Поиск оптимального маршрута происходит следующим образом. Каждый муравей движется в поисках пищи случайным образом. При этом, если из некоторого конкретного пункта имеется N путей, то вероятность выбора i–го пути вычисляется по формуле

$$P_i = \frac{l_i^q f_i^p}{\sum_{k=1}^{N} l_k^q f_k^p},$$

где l_i – длина перехода по i–му пути,

f_i – количество феромона на i–м пути,

q – величина, определяющая «жадность» алгоритма,

p – величина, определяющая «стадность» алгоритма,

$p + q = 1.$

Если этот муравей находит источник пищи, он возвращается к муравейнику, оставляя за собой феромоновый след. Ближайшие муравьи, обнаружив след, движутся по отмеченному феромоном пути, усиливая его. Таким образом, удачный маршрут подтверждается и усиливается, а неудачные ослабевают, так как феромон выветривается.

Алгоритм роя частиц – это метод численной многомерной оптимизации нулевого порядка. Для реализации алгоритма не требуется вычисление градиента оптимизируемой функции. Алгоритм работает следующим образом. Цель реализуемой в алгоритме вычислительной процедуры – найти в многомерном координатном пространстве точки, соответствующие максимальному значению целевой функции. Поиск осуществляется путем перемещения в этом пространстве множества частиц, положение каждой из которых в каждый момент времени задано своими координатами. При этом для каждой частицы известно её наилучшее положение в прошлом. Кроме того, известно наилучшее положение для всего роя частиц. Направление перемещения каждой частицы на очередном шаге определяется случайным образом, исходя из равномерного распределения, а скорость выбирается исходя из компромисса между удалениями до наилучших положений в прошлом. Для нового положения вычисляется значение целевой функции, которое запоминается, если оно лучше предыдущих. Останов процедуры происходит, если в течение некоторого заданного числа шагов не произошло удушения значения целевой функции.

Наконец, метод дифференциальной эволюции организован в виде пошаговой процедуры поиска точки в многомерном пространстве с наилучшим значением целевой функции и имеет много общего с алгоритмом роя частиц. В процессе работы метода формируется некоторое множество точек со своими координатами. На очередном шаге формируется новое поколение точек. При этом для каждой точки из старого поколения случайным образом выбирается ещё три разные точки из старого поколения, линейная комбинация которых определяет так называемую мутантную точку. Эта мутантная точка скрещивается (путем обмена некоторых координат) с исходной точкой. Полученная точка называется пробной. Если значение целевой функции в пробной точке лучше, чем её значение в исходной точке, то пробная точка заменяет исходную. Процедура останавливается, если в течение некоторого числа итераций не произошло улучшения целевой функции. Из приведенного описания метода дифференциальной диагностики ясно, что он является одной из модификаций генетического алгоритма.

Алгоритмы этого типа могут успешно применяться при решении многих задач маршрутизации в случаях, когда задача состоит в построении рационального маршрута в целях его дальнейшего многократного использования. Однако они совершенно не пригодны для быстрого решения множества задач маршрутизации с разными пунктами назначения. В связи с этим поставим задачу построения метода отыскания маршрута в реальном времени между известным начальным пунктом и случайным конечным пунктом.

Цель исследования. Постановка задачи. Цель исследования состоит в разработке метода, обеспечивающего построение кратчайшего маршрута между заданными начальным и конечным пунктами, с учетом реальных транспортных магистралей. Положение всех пунктов задано их координатами в декартовой системе координат. В рассматриваемой задаче доставки изготавливаемого продукта от производителя к конкретному потребителю по его требованию задача маршрутизации должна решаться немедленно и максимально быстро, поскольку малое время доставки продукта – одно из основных требований к системе. В связи с этим целесообразно отказаться от поиска алгоритма, обеспечивающего отыскание кратчайшего маршрута. В практических целях поставим задачу нахождения короткого маршрута, близкого к оптимальному, но зато предельно быстро.

Метод отыскания короткого маршрута в реальном времени. Пусть заданы n пунктов, для каждой пары из которых нужно найти кратчайший маршрут. Пункты соединены между собой дугами, обозначающими возможность перехода из одного в другой. Каждой дуге приписаны два числа, определяющие длину дуги придвижении в прямом и обратном направлениях. Таким образом, получен ориентированный граф. Технологию решения задачи удобно иллюстрировать на конкретном примере.

Пусть ориентированный граф имеет вид, представленный на рис. 1. Информация, приведенная на рисунке, также отображена в приводимой ниже матрице длин одношаговых переходов между пунктами $\boldsymbol{M}_1$.

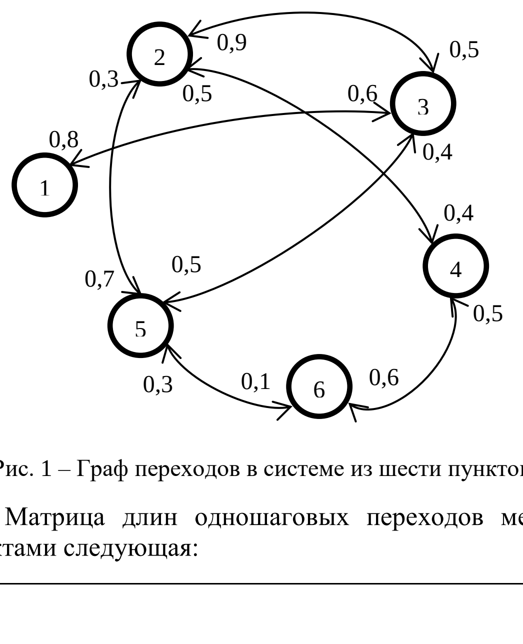

Рис. 1 – Граф переходов в системе из шести пунктов

Матрица длин одношаговых переходов между пунктами следующая:

ISSN 2079-0023 (print)
ISSN 2410-2857 (online)

$\boldsymbol{M}_1 =$

	1	2	3	4	5	6
1	∞	G	0,6	G	G	G
2	G	∞	0,5	0,4	0,7	G
3	0,8	0,9	∞	G	0,5	G
4	G	0,5	G	∞	G	0,6
5	G	0,3	0,4	G	∞	0,1
6	G	G	G	0,5	0,3	∞

.

В матрице $\boldsymbol{M}_1$ символом G для элемента (i, j) обозначена ситуация, когда между пунктами i и j одношаговый переход отсутствует.

Определим теперь длины кратчайших двухшаговых переходов между пунктами. С этой целью введем специальную операцию компиляции матриц. Для двух квадратных матриц $\boldsymbol{A} = (a_{ij})$ и $\boldsymbol{B} = (b_{ij})$ введем матрицу $\boldsymbol{C} = (c_{ij})$, элементы которой рассчитываются по правилу

$$c_{ij} = \min_k\{(a_{ik} + b_{kj})\}, \tag{5}$$

$$\dim \boldsymbol{A} = \dim \boldsymbol{B} = \dim \boldsymbol{C} = n \times n,$$

$$i = 1,2,\ldots,n, \quad j = 1,2,\ldots,n.$$

В матричной форме эту операцию запишем формулой $\boldsymbol{C} = \boldsymbol{A} \oplus \boldsymbol{B}$.

С использованием операции (5) осуществим расчет матрицы $\boldsymbol{M}_2 = \boldsymbol{M}_1 \oplus \boldsymbol{M}_1$:

$\boldsymbol{M}_2 = \boldsymbol{M}_1 \oplus \boldsymbol{M}_1 =$

	1	2	3	4	5	6
1	∞	G	0,6	G	G	G
2	G	∞	0,5	0,4	0,7	G
3	0,8	0,9	∞	G	0,5	G
4	G	0,5	G	∞	G	0,6
5	G	0,3	0,4	G	∞	0,1
6	G	G	G	0,5	0,3	∞

$\oplus$

	1	2	3	4	5	6
1	∞	G	0,6	G	G	G
2	G	∞	0,5	0,4	0,7	G
3	0,8	0,9	∞	G	0,5	G
4	G	0,5	G	∞	G	0,6
5	G	0,3	0,4	G	∞	0,1
6	G	G	G	0,5	0,3	∞

$=$

	1	2	3	4	5	6
1	G	1,5	0,6	G	1,1	G
2	1,3	G	0,5	0,4	0,7	0,8
3	0,8	0,8	G	1,3	0,5	0,6
4	G	0,5	1	G	0,9	0,6
5	1,2	0,3	0,4	0,6	G	0,1
6	G	0,6	0,6	0,5	0,3	G

.

Ниже приведены несколько примеров расчета длины двухшаговых путей:

$$S_{26} = \min\{(S_{24} + S_{46}), (S_{25} + S_{56})\} =$$

$$= \min\{(0{,}4 + 0{,}6), (0{,}7 + 0{,}1)\} = 0{,}8,$$

$$S_{32} = \min\{S_{32}, (S_{35} + S_{52})\} =$$

$$= \min\{0{,}9, (0{,}5 + 0{,}3)\} = 0{,}8,$$

$$S_{45} = \min\{(S_{46} + S_{65}), (S_{42} + S_{25})\} =$$

$$= \min\{(0{,}6 + 0{,}3), (0{,}5 + 0{,}7)\} = 0{,}9;$$

$$S_{54} = \min\{(S_{52} + S_{24}), (S_{56} + S_{64})\} =$$

$$= \min\{(0{,}3 + 0{,}4), (0{,}1 + 0{,}5)\} = 0{,}6;$$

$$S_{62} = \min\{(S_{64} + S_{42}), (S_{65} + S_{52})\} =$$

$$= \min\{(0{,}5 + 0{,}5), (0{,}3 + 0{,}3)\} = 0{,}6.$$

Отметим, что в матрице $\boldsymbol{M}_2$ меньше элементов, отмеченных символом G. Это означает, что предложенная процедура (5) выявляет кратчайшие двухшаговые пути, соединяющие пункты, которые непосредственно не связаны между собой. Однако, поскольку такие элементы в $\boldsymbol{M}_2$ все же присутствуют, сделаем ещё один шаг расчетов, результаты которых определят наличие и эффективность трехшаговых путей:

$$\boldsymbol{M}_3 = \boldsymbol{M}_2 \oplus \boldsymbol{M}_1 = (S_{ij}^3) =$$

$=$

	1	2	3	4	5	6
1	G	1,5	0,6	G	1,1	G
2	1,3	G	0,5	0,4	0,7	0,8
3	0,8	0,8	G	1,3	0,5	0,6
4	G	0,5	1	G	0,9	0,6
5	1,2	0,3	0,4	0,6	G	0,1
6	G	0,6	0,6	0,5	0,3	G

$\oplus$

	1	2	3	4	5	6
1	∞	G	0,6	G	G	G
2	G	∞	0,5	0,4	0,7	G
3	0,8	0,9	∞	G	0,5	G
4	G	0,5	G	∞	G	0,6
5	G	0,3	0,4	G	∞	0,1
6	G	G	G	0,5	0,3	∞

$=$

	1	2	3	4	5	6
1	G	1,1	0,6	1,8	1,1	1,2
2	1,3	G	0,5	0,4	0,7	0,8
3	0,8	0,9	G	1,1	0,5	0,6
4	1,8	0,5	1	G	0,9	0,6
5	1,2	0,3	0,4	0,6	G	0,1
6	1,5	0,6	0,7	0,5	0,3	G

.

В матрице $\boldsymbol{M}_3$ уже не осталось элементов, помеченных символом G. Это значит, что в приведенной системе пунктов любой из них достижим из любого другого не более, чем за три перехода. Поэтому дальнейшие расчёты уже не нужны. Завершающая часть алгоритма отыскания матрицы кратчайших путей состоит в следующем.

Итоговая матрица $\boldsymbol{M}_3$ содержит значения кратчайших путей от каждого из пунктов к каждому. Для практического использования этой матрицы её необходимо дополнить ещё одной матрицей, содержащей для каждой пары «начальный пункт – конечный пункт» информацию о номерах промежуточных пунктов оптимального маршрута. Эта матрица для данной задачи имеет вид, приведенный в табл. 1.

Таблица 1– Перечень номеров пунктов маршрута

	1	2	3	4	5	6
1	–	1, 3, 2	1, 3	1, 3, 2, 4	1, 3, 5	1, 3, 5, 6
2	2, 3, 1	–	2, 3	2, 4	2, 5	2, 5, 6
3	3, 1	3, 2	–	5, 6, 4	3, 5	3, 5, 6
4	4, 2, 3, 1	4, 2	4, 2, 3	–	4, 6, 5	4, 6
5	5, 3, 1	5, 2	5, 3	5, 6, 4	–	5, 6
6	6, 5, 3, 1	6, 5, 2	6, 5, 3	6, 4	6, 5	–

Следует отметить высокую скорость описанного метода отыскания кратчайших путей. Для системы содержащей n пунктов необходимо не более n раз выполнить операцию компиляции, реализация которой требует $\approx n^3$ элементарных операций. Тогда общее число операций при решении задачи имеет порядок $N = n^4$. Если $n = 32$, то $N \cong (2^5)^4 = 2^{20} \cong 10^6$.

Рассмотрим теперь возможность применения предложенной технологии для построения кратчайших маршрутов в задаче организации поставок в системе «производитель – множество потребителей», произвольным образом распределенных в районах города. На карте города построим прямоугольник, охватывающий все его районы. Этот прямоугольник разобьём на квадраты, со стороной, равной l км. Пусть в результате разбиения получено N квадратов, которые перенумеруем по порядку, $i = 1,2,...,N$. На рис. 2 приведен пример такого разбиения.

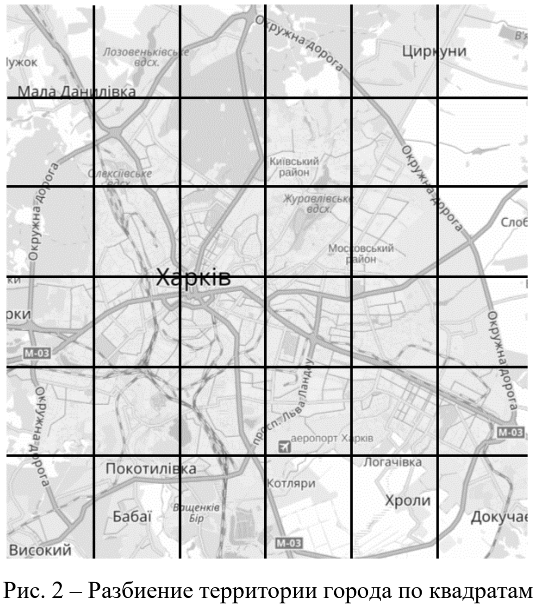

Рис. 2 – Разбиение территории города по квадратам

Одновременно формируется аналог табл. 1, определяющей последовательность прохождения квадратов. В реальной ситуации для отыскания кратчайшего маршрута от производителя до конкретного потребителя выполняются следующие операции. Сначала по координатам заказчика отыскивается номер квадрата, в котором он находится. Затем, используя матрицу $\boldsymbol{M}_3$, а также соответствующую ей табл. 1, отыскивается рациональная последовательность прохождения квадратов, принадлежащих искомому маршруту. После попадания в последний квадрат, накрывающий место расположения потребителя, завещающий участок маршрута находится непосредственно. Получаемый в результате маршрут не является гарантированно кратчайшим, однако многочисленные эксперименты показали, что этот маршрут близок к нему.

Задача существенно усложняется, если в качестве критерия эффективности маршрутизации использовать не длину маршрута, а продолжительность его прохождения. Возникающая в этой задаче неопределенность в отношении времени прохождения каждого участка маршрута связана с влиянием на это время множества трудно учитываемых факторов (время года, время суток, погодные условия и т. д.). Решение задачи в этой ситуации возможно с использованием теоретико-вероятностных моделей [14–17] или с привлечением моделей нечеткой математики [18, 19].

Выводы. Проведенный анализ известных методов отыскания кратчайшего маршрута показал, что они не могут быть реализованы для решения таких задач в реальном времени.

Предложена простая вычислительная процедура, которая обеспечивает быстрое нахождение кратчайшего маршрута с приемлемой точностью.

Список литературы

1. *Юдин Д. Б.* Задачи и методы линейного программирования / *Д. Б. Юдин, Е. Г. Гольштейн*. – М. : Сов. радио, 1961. – 494 с.
2. *Раскин Л. Г.* Многомерные задачи линейного программирования / *Л. Г. Раскин*. – М. : Радио и связь, 1982. – 240 с.
3. *Саати Т.* Целочисленные методы оптимизации и связанные с ними экстремальные проблемы / *Т. Саати*. – М. : Мир, 1973. – 304 с.
4. *Литтл Дж.* Алгоритм для решения задачи о коммивояжере / *Дж. Литтл, К. Мурти. Д Суини.* // Экономика и математические методы. – 1965. – Т. 1, Вып. 1 – С. 94–107 с.
5. *Гомори Р.* Численные методы оптимального планирования / *Р. Гомори, У. Бомоль*. – Новосибирск : СО АН СССР, 1962. – С. 58-72.
6. *Михалевич В. С.* Последовательные алгоритмы оптимизации и их применение. / *В. С. Михалевич* // Кибернетика. – 1965. – № 1. – С. 16–28.
7. *Корбут А. А.* Дискретное программирование / *А А. Корбут, Ю. Ю. Финкельштейн*. – М. : Наука, 1969. – 284 с.
8. *Balas E.* Solution of Large Scale Transportation Problems Through Aggregation / *E. Balas* // Operations Research. – 1965. – Vol. 13. – pp. 82–93.
9. *Миллер П.* Роевой интеллект: Муравьи, пчелы и птицы способны многому нас научить// *П. Миллер, /* National Geographic Россия. – 2007. – № 8. – С. 88–107.
10. *Ignatyev A. L.* Solving the Travelling Salesman Problem on Shared and Distributed Memory Multiprocessor Systems / *A. L. Ignatyev, M. A. Posypkin, I. Kh. Sigal //* Proceeding International conference "Optimization and applications". – Petrovac, Montenegro. – 2009. – P. 36–39.
11. *Kravets P.* The control agent with fuzzy logic / *P. Kravets* // Perspective Technologies and Methods in MEMS Design, MEMSTECH'2010 – Proceedings of the 6th International Conference. – Lviv, – 2010. – P. 40–41.
12. *Bel J. E.* Ant colony optimization techniques for the vehicle routing problem / *J. E. Bell, P. R. McMullen* // Advanced Engineering Informatics. – 2004.– Vol. 18. – P. 41–48.

13. *Dorigo M.* Ant algorithms for discrete optimization / *Dorigo M., Di Caro G., Gambardella L. M.* //Artificial Life. – 1999. – Vol. 5. – P. 137–172.
14. *Раскин Л. Г.* Анализ сложных систем и элементы теории управления / *Л. Г. Раскин*. – М. : Сов. Радио, 1976. – 344 с.
15. *Пигнастый О. М.* Статистическая теория производственных систем / *О. М. Пигнастый*. – Х. : ХНУ им. Каразина, 2007. – 388 с.
16. *Пигнастый О. М.* Стохастическое описание экономико–производственных систем с массовым выпуском продукции / *В. П. Демуцкий, В. С. Пигнастая, О. М. Пигнастый* // Доповіді Нац. академії наук. – 2005. – № 7. – С. 66–71.
17. *Демуцкий В. П.* Теория предприятия: Устойчивость функционирования массового производства и продвижения продукции на рынок / *В. П. Демуцкий, В. С. Пигнастая, О. М. Пигнастый*. – Х. : ХНУ им. Каразина, 2003. – 272 с.
18. *Серая О. В.* Многомерные модели логистики в условиях неопределенности / *О. В. Серая.* – Х. : ФОП Стеценко, 2010. – 512 с.
19. *Серая О. В.* Модели и информационные технологии оценки и прогнозирования состояния многомерных динамических объектов в условиях нечетких исходных данных : дис. … канд. техн. наук: 05.13.06 : утв. 17.01.02 / *О. В. Серая*. – Х. : 2001. – 252 с.

References (transliterated)

1. Yudyn D. B., Hol'shteyn E. H. *Zadachy y metodi lyneynoho prohrammyrovanyya* [Objectives and methods of linear programming]. Moscow, Sov. radyo Publ., 1961. 494 p.
2. Raskyn L. H. *Mnohomernie zadachy lyneynoho prohrammyrovanyya* [Multivariate linear programming problem]. Moscow, Radyo y svyaz' Publ., 1982. 240 p.
3. Saaty T. *Tselochyslennie metodi optymyzatsyy y svyazannie s nymy ekstremal'nie problemi* [Integer optimization techniques and associated extreme problems]. Moscow, Myr Publ., 1973. 304 p.
4. Lyttl Dzh., Murty K., Suyny D. Alhorytm dlya reshenyya zadachy o kommyvoyazhere [The algorithm for solving the traveling salesman problem]. *Ekonomyka y matematycheskye metodi*. 1965, vol. 1, no. 1, pp. 94–107.
5. Homory R., Bomol' U. *Chyslennie metodi optymal'noho planyrovanyya* [Numerical methods of optimal planning]. Novosybyrsk, SO AN SSSR Publ., 1962, pp. 58–72.
6. Mykhalevych V. S. Alhorytm dlya reshenyya zadachy o kommyvoyazhere [Posledovatel'nie alhorytmi optymyzatsyy y ykh prymenenye]. *Kybernetyka*. 1965, vol. 1, pp. 16–28.
7. Korbut A. A., Fynkel'shteyn Yu. Yu. *Dyskretnoe prohrammyrovanye* [Discrete programming]. Moscow, Nauka Publ., 1969. 284 p.
8. Balas E. Solution of Large Scale Transportation Problems Through Aggregation. *Operations Research*. 1965, vol. 13, pp. 82–93.
9. Myller P. Roevoy yntellekt: Murav'y, pcheli y ptytsi sposobni mnohomu nas nauchyt' [Swarm intelligence: Ants, bees, and birds can teach us a lot]. *National Geographic Rossyya*. 2007, no. 8, pp. 88–107.
10. Ignatyev A. L. Solving the Travelling Salesman Problem on Shared and Distributed Memory Multiprocessor Systems. *Proceeding International conference "Optimization and applications"*. Petrovac, Montenegro, 2009, pp. 36–39.
11. Kravets P. The control agent with fuzzy logic. *Perspective Technologies and Methods in MEMS Design "MEMSTECH'2010", Proc.eedings of the 6th International Conference.* Lviv, 2010, pp. 40–41.
12. Bel J. E. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*. 2004, vol. 18, pp. 41–48.
13. Dorigo M., Di Caro G., Gambardella L. M. Ant algorithms for discrete optimization. *Artificial Life*. 1999, vol. 5, pp. 137–172.
14. Ruskin L. G. *Analiz slozhnyh sistem i jelementy teorii upravlenij* [The analysis of complex systems and controls theory]. Moscow, Sov. Radyo Publ., 1976, 344 p.
15. Pyhnastiy O. M. *Statystycheskaya teoryya proyzvodstvenikh system* [Statistical theory of production systems]. Kharkov, KhNU im. Karazyna Publ., 2007, 388 p.
16. Demutskyy V. P., Pyhnastaya V. S., Pyhnastiy O. M. Stokhastycheskoe opysanie ekonomyko–proyzvodstvennikh system s massovim vipuskom produktsyy [Stochastic description of the economic and production systems to mass production]. *Dopovidi Nats. akademiyi nauk*. 2005, no. 7, pp. 66–71.
17. Demutskyy V. P., Pyhnastaya V. S., Pyhnastiy O. M. *Teoryya predpryyatyya: Ustoychyvost funktsyonyrovanyya massovoho proyzvodstva y prodvyzhenyya produktsyy na rinok* [Enterprise theory: Stability of functioning of mass production and promotion of products on the market]. Kharkov, KhNU im. Karazyna Publ., 2003, 272 p.
18. Seraya O. V. *Mnohomernie modely lohystyky v uslovyyakh neopredelennosty* [Multivariate logistic models under uncertainty]. Kharkiv, FOP Stetsenko Publ., 2010, 512 p.
19. Seraya O. V. *Modeli i informatsionnyie tehnologii otsenki i prognozirovaniya sostoyaniya mnogomernyih dinamicheskih ob'ektov v usloviyah nechetkih ishodnyih dannyih*. *dis. ... kand. tekhn. nauk* 05.13.06 [Models and information technology assessment and forecasting of multivariate dynamic objects in a fuzzy initial data. Candidate eng. sci. diss. (Ph. D.)]. Kharkiv, 2001, 252 p.

Поступила (received) 06.12.2016

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Метод вирішення задачі маршрутизації у реальному часі / В. В. Карпенко // Вісник НТУ «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Х. : НТУ «ХПІ», 2016. – № 45 (1217). – С. 59–64. – Бібліогр.: 19 назв. – ISSN 2079–0023.

Метод решения задачи маршрутизации в реальном времени / В. В. Карпенко // Вісник НТУ «ХПІ». Серія: Системний аналіз, управління та інформаційні технології. – Харків : НТУ «ХПІ», 2016. – № 45 (1217). – С. 59–64. – Библиогр.: 19 назв. – ISSN 2079–0023.

The method of solving the problem of routing in real time / V. V. Karpenko // Bulletin of NTU "KhPI". Series: System analysis, control and information technology. – Kharkov : NTU "KhPI", 2016. – No. 45 (1217). – P. 59–64. – Bibliogr.: 19. – ISSN 2079–0023.

Відомості про авторів / Сведения об авторах / About the Authors

Карпенко Вячеслав Васильович – Національний технічний університет «Харківський політехнічний інститут», старший викладач кафедри Комп'ютерного моніторингу та логістики; тел.: (093) 643–19–39; e–mail: Karpenko@kml.kh.ua.

Карпенко Вячеслав Васильевич – Национальный технический университет «Харьковский политехнический институт», старший преподаватель кафедры Компьютерного мониторинга и логистики; тел.: (093) 643–19–39; e–mail: Karpenko@kml.kh.ua.

Karpenko Vyacheslav Vasilevich – National Technical University "Kharkiv Polytechnic Institute", senior lecturer Department of Computer Monitoring and logistics; tel.: (093) 643–19–39; e–mail: Karpenko@kml.kh.ua.