УДК 004.021:338.47

ПРИМЕНЕНИЕ ИНСТРУМЕНТАЛЬНОГО СРЕДСТВА НА ОСНОВЕ ТРЁХМЕРНЫХ СТРУКТУР ДАННЫХ В ТРАНСПОРТНОМ АСПЕКТЕ СИСТЕМЫ ОБРАЩЕНИЯ С КОММУНАЛЬНЫМИ ОТХОДАМИ

Н. Н. Леонович[1], Т. Ф. Старовойтова[2], Т. Г. Хомицкая[3]

[1] Аспирантка кафедры управления информационными ресурсами Академии управления при Президенте Республики Беларусь, г. Минск, Республика Беларусь, e-mail: nnleonovich@g.bstu.by
[2] К. э. н., доцент, доцент кафедры управления информационными ресурсами Академии управления при Президенте Республики Беларусь, г. Минск, Республика Беларусь, e-mail: Tan.star00@gmail.com
[3] Старший преподаватель кафедры информатики и прикладной математики Брестского государственного технического университета, г. Брест, Республика Беларусь, e-mail: tgh@g.bstu.by

Реферат

Базовым способом решения транспортной задачи коммивояжёра является метод ветвей и границ, в основе которого лежит последовательное разбиение множества допустимых решений на подмножества. При этом, на каждом шаге метода, подмножества проверяются на оптимальность, посредством вычисления оценки снизу для целевой функции.

В данной статье рассматривается способ применения метода ветвей и границ в виде алгоритма Литтла для поиска кратчайшего маршрута движения по заданной матрице расстояний. Авторами статьи метод автоматизирован в виде процедур и функций, созданных в Visual Basic for Application приложения Microsoft Excel с использованием трёхмерных массивов.

Ключевые слова: задача коммивояжёра, метод ветвей и границ, алгоритм Литтла, автоматизация, Visual Basic for Application.

TOOL USE BASED ON THREE-DIMENSIONAL DATA STRUCTURES IN THE TRANSPORT ASPECT OF THE SYSTEM MUNICIPAL WASTE MANAGEMENT

N. N. Leonovich, T. F. Starovoitova, T. G. Khomitskaya

Abstract

The basic way to solve the transport traveling salesman problem is the branch and bound method, which is based on the sequential partitioning of the set of feasible solutions into subsets. At the same time, at each step of the method, subsets are checked for optimality by calculating a lower estimate for the objective function.

This article discusses how to apply the branch and bound method in the form of Little's algorithm to find the shortest route for a given distance matrix. The authors of the article automated the method in the form of procedures and functions created in the Visual Basic for Application of Microsoft Excel using three-dimensional arrays.

Keywords: traveling salesman problem, branch and bound method, Little's algorithm, automation, Visual Basic for Application.

Введение

Математическое моделирование играет важную роль в решении различных задач, позволяя оценить структуру для целостного анализа, и в связи с этим является неотъемлемой частью любого исследования в области экономики. Ввиду сложности экономических систем, для их структурного описания используются различные подходы, одним из которых является линейное программирование. В общем случае задача, в которой необходимо найти самый дешевый способ удовлетворить спрос в n пунктах назначения, таким образом уменьшив транспортные издержки предприятия, делая его конкурентоспособным, является транспортной задачей с возможностью исследования при помощи методов линейного программирования. Частным случаем транспортной задачи является задача коммивояжёра.

Задача коммивояжёра (от англ. Travelling salesman problem (TSP)) заключается в поиске самого выгодного маршрута, проходящего через указанные точки хотя бы по одному разу с последующим возвратом в первоначальную точку. В условиях задачи указываются критерий выгодности маршрута (кратчайший, самый дешевый или др.) и соответствующие матрицы (расстояний, стоимости и др.). Как правило, маршрут должен проходить через каждый пункт только один раз и условный коммивояжёр должен вернуться в исходный пункт отправления, другими словами, во взвешенном графе требуется найти гамильтонов контур минимального веса. Сложность данной задачи заключается в том, что при увеличении количества городов, при полном переборе всех вариантов, количество возможных маршрутов будет очень большим: при составлении маршрута для 10 пунктов – число перестановок составит n!=3 628 800, а для 15 пунктов – уже более одного триллиона (1 307 674 368 000).

Задача коммивояжёра находит применение на практике в различных областях экономики и человеческой деятельности: финансы (оптимизация денежных потоков), туризм (расчет маршрутов экскурсий и туров), шоу-бизнес (организация турне музыкальных артистов), телекоммуникации и связь (проектирование телекоммуникационных сетей, управление спутниками), энергетика и коммунальное хозяйство (соединение населенных пунктов линиями электропередач, газоснабжения, схем вывоза мусора), электроника (проектирование топологий микросхем, оптимальное вырезание чипов лазером) и мн. др.

Для поиска решения задачи коммивояжёра кроме эвристических способов решения (муравьиный, генетический алгоритмы) существуют комбинаторные (метод северо-западного угла, симплекс-метод), среди которых следует выделить метод ветвей и границ [1]. В отличии от глобального рассмотрения всех возможных вариантов (полного перебора), в методе ветвей и границ происходит отсев подмножеств допустимых решений, заведомо не содержащих оптимальных. Основу метода составляют две процедуры: ветвления и нахождения оценок (границ). Процедура ветвления состоит в разбиении множества допустимых решений на подмножества (подобласти) меньших размеров. Полученные подмножества образуют бинарное дерево поиска (в каждом узле порождаются 2 ветви), например, содержит маршрут выбранную дугу перехода или не содержит. Процедура нахождения оценок заключается в поиске верхних и нижних границ для решения задачи на подмножестве допустимых решений. Суть заключается в следующей идее: если нижняя граница значений функции на подобласти P больше, чем верхняя граница на какой-либо ранее рассмотренной подобласти, то P может быть отсеяна из дальнейшего рассмотрения. Если нижняя граница для узла

doi.org/10.36773/1818-1112-2022-128-2-168-172

дерева совпадает с верхней границей, то это значение является минимумом функции на соответствующей подобласти.

Метод ветвей и границ имеет широкое применение во многих экономических практических задачах: календарное планирование, упорядочение работ в системе конвейерного типа, сетевое планирование с ограниченными ресурсами.

Целью данного исследования является создание инструментального средства в виде программной реализации метода ветвей и границ для автоматизации формирования оптимального (кратчайшего) маршрута движения автомобиля.

Математическая модель транспортной задачи коммивояжёра

Рассматривается n пунктов назначения, связанных дорожной сетью. С каждой дугой (i, j) связано значение c_{ij}, интерпретируемое как расстояние от пункта i до пункта j. Маршрут или путь – это произвольная последовательность дуг, соединяющих данные пункты. Пусть булевы переменные x_{ij} принимают значение 1 (истина), если коммивояжёр переезжает из i-го пункта в j-й пункт и 0 (ложь), если j-й пункт не посещается после i-го. В таком случае решением задачи будет определение минимума целевой функции – пройденного расстояния:

$$F(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min, \quad (1)$$

при ограничениях:

$$\sum_{i=1}^{n} x_{ij} = 1, \ j = \overline{1, n} \ ; \quad (2)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \ i = \overline{1, n} \ ; \quad (3)$$

$$x_{ij} \in \{0,1\}, \ i, j = \overline{1, n}, i \neq j. \quad (4)$$

Здесь функция (1) определяет расстояние выбранного маршрута. Условие (2) в силу ограничения (4) указывает на то, что после пункта i может следовать только один пункт, а условие (3) – перед пунктом j может быть также только один пункт.

Для того чтобы обеспечить замкнутость маршрута (выйти из первого и вернутся, после обхода всех пунктов только один раз, опять в первый) и избежать замкнутых подциклов (несвязанных между собой), необходимы дополнительные ограничения, связывающие переменные x_{ij}. Не допускается расщепление замкнутого из n+1 звеньев маршрут условного коммивояжёра на несколько замкнутых маршрутов меньшего числа звеньев [2]:

$$u_i - u_j + n x_{ij} \leq n - 1, \ x_{ij}, i, j = \overline{2, n}, \ i \neq j. \quad (2)$$

Таким образом, условие (5) определяет построение гамильтонового цикла из n вершин. Следует отметить, что в сформулированной задаче в качестве неизвестных, кроме переменной x_{ij}, выступают переменные $u_i, \ i = \overline{2, n}$.

Алгоритм решения задачи коммивояжёра посредством метода ветвей и границ был предложен в 1963 году группой учёных Дж. Литтлом, К. Мурти, Д. Суини, К. Кэролом и в дальнейшем получил название алгоритм Литтла [3].

Пример решения транспортной задачи при помощи алгоритма Литтла

В качестве примера рассмотрим процесс поиска оптимального (с минимальной длиной) маршрута движения автомобиля для сбора твердых коммунальных отходов из пяти контейнерных площадок Брестского района, где первая точка маршрута – мусороперерабатывающий завод, из которого начинается и которым должен завершатся маршрут, итого шесть пунктов маршрута.

Расстояния между пунктами расположения контейнерных площадок взяты из геоинформационной системы Яндекс.Карты [4] (рисунок 1), округлены до целых (для наглядности расчетов) и представлены в исходной матрице W_0 (рисунок 2). Всем элементам матрицы расстояний, расположенным на главной диагонали, присвоены достаточно большие, по сравнению с остальными, значения, так как соответствующие им дуги – это петли, которые не должны включаться в маршрут.

Рисунок 1 – Поиск расстояния по карте между двумя пунктами в геоинформационной системе Яндекс.Карты [4]

1. *Получение приведённой матрицы* W_0^*. Для этого необходимо выписать справа от каждой строки исходной матрицы W_0 минимальный элемент, а затем вычесть из каждой строки ее минимальный элемент; далее необходимо выписать под каждым столбцом полученной матрицы минимальный элемент этого столбца и затем вычесть из каждого столбца его минимальный элемент (нулевые минимальные элементы не учитываются, так как их вычитание ничего не изменит). Таким образом получится приведённая матрица W_0^*, у которой в каждой строке и каждом столбце содержится нуль.

W_0	1	2	3	4	5	6	
1	1000	11	9	12	9	7	7
2	6	1000	16	12	10	4	4
3	3	13	1000	6	10	2	2
4	3	1	2	1000	4	18	1
5	5	3	1	2	1000	11	1
6	4	5	8	7	9	1000	4

→

W_0	1	2	3	4	5	6
1	993	4	2	5	2	0
2	2	996	12	8	6	0
3	1	11	998	4	8	0
4	2	0	1	999	3	17
5	4	2	0	1	999	10
6	0	1	4	3	5	996
				1	2	

→

W_0^*	1	2	3	4	5	6
1	993	4	2	4	0	0
2	2	996	12	7	4	0
3	1	11	998	3	6	0
4	2	0	1	998	1	17
5	4	2	0	0	997	10
6	0	1	4	2	3	996

Рисунок 2 – Получение приведенной матрицы W_0^* из исходной матрицы расстояний

2. *Нахождения предварительной нижней оценки* C_0 *общей длинны маршрута.* Для этого достаточно просуммировать все вычтенные числа: $C_0 = 7 + 4 + 2 + 1 + 1 + 4 + 1 + 2 = 22$. Число C_0 указывает, что построить маршрут меньшей длины нельзя.
3. *Вычисление оценок (степеней) всех нулевых элементов приведенной матрицы.* Для этого для каждого нуля вычисляется сумма минимальных элементов строки и столбца, на пересечении которых он располагается без учета рассматриваемого нуля.
4. *Выбор нуля с наибольшей степенью (если таких несколько, то выбирается любой)*. Альтернативным является первый нуль с наибольшей степенью 2 (рисунок 3).
5. *Разбиение исходного множества на два*. Так как выбранный нуль стоит на пересечении 2-й строки и 6-го столбца, рассматриваются два варианта:

- проезд из 2-го пункта в 6-й (2→6);
- исключение проезда из 2-го пункта в 6-й (2↛6).

Записываются две матрицы W_{11} и W_{12}, соответствующие каждому из двух случаев. Для получения матрицы W_{11} (для случая 2→6) из приведённой матрицы удаляется строка и столбец, на пересечении которых находится нуль с наибольшей степенью. Дополнительно ставится достаточно большое значение в 6-й строке и 2-м столбце, запрещая построение неполного цикла переходов (см. рисунок 3).

Для получения матрицы W_{12} (для случая 2↛6) в приведённой матрице вместо нуля с наибольшей степенью, для запрета перехода, ставится достаточно большое значение.

Далее для каждой из матриц W_{11}, W_{12} определяются числа, которые необходимо вычесть из строк и столбцов, чтобы получить из нее приведённую матрицу (рисунок 3).

doi.org/10.36773/1818-1112-2022-128-2-168-172

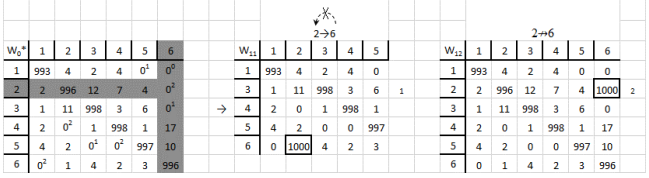

Рисунок 3 – Вычисление нулевых степеней и две матрицы W_{11} и W_{12} для случаев посещения или непосещения пункта 6 маршрута

6. *Уточнение оценок для каждой матрицы.* Для получения уточнённой оценки увеличивается предыдущая оценка (C_0) на сумму вычтенных чисел:

- $C_{11} = 22 + 1 = 23$
- $C_{12} = 22 + 2 = 24$

Строится 1-е ветвление на дереве решений (рисунок 4)

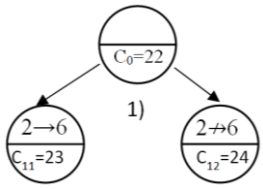

Рисунок 4 – 1-е ветвление на дереве решений

Следует заметить, что оценка множества с запрещением перехода есть сумма предыдущей оценки C_0 и оценки выбранного нуля.

7. *Из всех открытых на данный момент ветвей выбирается ветвь с наименьшей оценкой* (C_{11} = 23) *и в дальнейшем рассматривается соответствующая ей матрица* (W_{11}).

Далее в ходе решения для выбранной матрицы выполняются действия, аналогичные описанным выше пунктам 3–7 [5], до тех пор, пока порядок матрицы не станет равен двум.

Для выбранной матрицы W_{11} строится приведённая W_{11}^*. В приведённой матрице W_{11}^* вычисляются степени нулей и выбирается нуль с наибольшей степенью, записываются две матрицы W_{21} (для случая 1⟶5) и W_{22} (для случая 1↛5) (рисунок 5).

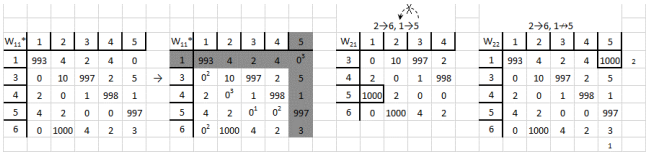

Рисунок 5 – Приведённая матрица W_{11}*, вычисление нулевых степеней и две матрицы W_{21} и W_{22} для случаев посещения или непосещения пункта 5 маршрута

Уточняются соответствующие оценки:

- $C_{21} = 23 + 0 = 23$
- $C_{22} = 23 + 2 + 1 = 26$

Строится 2-е ветвление на дереве решений (рисунок 6).

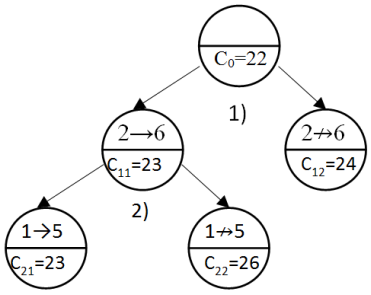

Рисунок 6 – 2-е ветвление на дереве решений

Из всех открытых на данный момент ветвей выбирается ветвь с наименьшей оценкой C_{21} и рассматривается соответствующая ей матрица W_{21}, для неё строится приведённая W_{21}^*, в которой вычисляются степени нулей и выбирается нуль с наибольшей степенью (рисунок 7).

Для нуля с максимальной степенью 3 соответствуют два случая: маршрут из пункта 4 продолжается в пункт 2 (4⟶2) или маршрут из пункта 4 не продолжается в пункт 2 (4↛2). В построенных соответствующих матрицах W_{31} и W_{32} в первом случае ставится запрет на посещение 4 пункта после 6-го (неполный цикл), а во втором – достаточно большие числа на пересечении 4-го и 2-го пунктов.

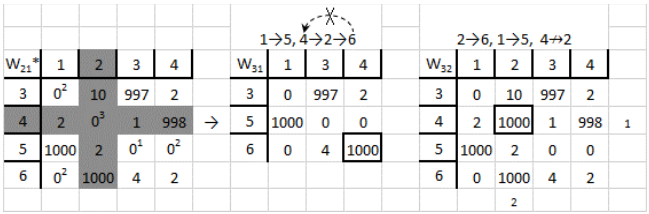

Рисунок 7 – Приведённая матрица W_{21}^*, вычисление нулевых степеней и две матрицы W_{31} и W_{32} для случаев посещения или непосещения пункта 2 маршрута

Уточненные оценки будут равны следующим значениям:

- $C_{31} = 23 + 0 = 23$
- $C_{32} = 23 + 1 + 2 = 26$

Строиться 3-е ветвление на дереве решений (рис. 8).

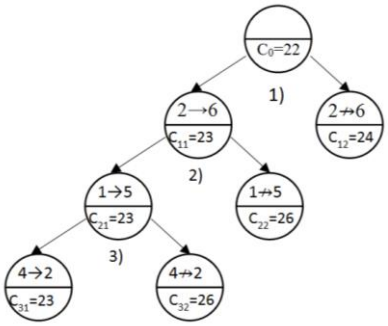

Рисунок 8 – 3-е ветвление на дереве решений

Из всех открытых на данный момент ветвей выбирается ветвь с наименьшей оценкой и соответствующая ей матрица. Для выбранной матрицы W_{31} с наименьшей оценкой C_{31} = 23 строится приведённая матрица W_{31}*, в которой вычисляются степени нулей и выбирается нуль с наибольшей степенью (рисунок 9). В связи с тем, что получившаяся матрица, после удаления соответствующей строки и столбца, на которых располагается нуль с наибольшей оценкой, имеет размерность 2 – дальнейшее построение ветвления на дереве решений не требуется.

При посещении 3-го пункта после 5-го необходимо поставить достаточно большое значение в строке, соответствующей 3-му пункту и столбцу, соответствующему 1-му пункту, запрещая построение неполного цикла переходов. В данном случае остается только два возможных перехода: из пункта 3 в пункт 4 и из пункта 6 в пункт 1.

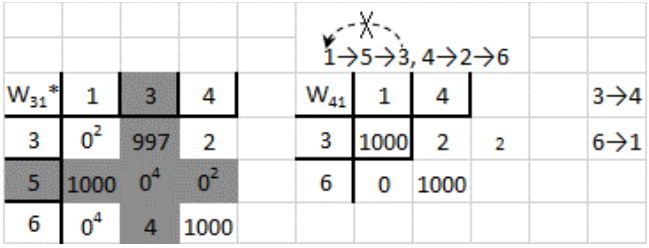

Рисунок 9 – Приведённая матрица W_{31}^*, вычисление нулевых степеней, конечная матрица размерности 2 для случая перемещения из пункта 5 в пункт 3

Уточненная оценка составит C_{41} = 23 + 2 = 25.

Таким образом, исходя из всего вышесказанного, оптимальный маршрут движения автомобиля, состоящий из шести пунктов посещения и последующего возврата в исходную точку, для данного примера

doi.org/10.36773/1818-1112-2022-128-2-168-172

будет выглядеть следующим образом: 1→5→3→4→2→6→1. Длинна пути составит 25 условных единиц.

Реализация автоматизированного поиска оптимального маршрута

Алгоритм Литтла, реализующий метод ветвей и границ, закодирован авторами статьи в среде программирования Visual Basic for Application (VBA) приложения Microsoft Excel (ME) в виде процедуры Algorithm_Littla(), где расстояния между пунктами назначения, которые необходимо посетить, представлены таблицей в виде матрицы.

При разработке указанной выше подпрограммы созданы для использования три ключевые функции: для приведения матрицы расстояний (One(w(), n, ww(), t, s, l)), для определения оценок нулей (Two(w(), n, smax, imax, jmax)) и для удаления строки и столбца при выборе пунктов отправления и прибытия с наименьшей оценкой (Three(w(), n, ww(), cc(), t, smax, imax, jmax)). Для каждой из функций входными аргументами (значениями) являются: w() – матрица расстояний, n – количество пунктов, ww() – 3D-матрица всех матриц расстояний. Дополнительные входные переменные: t – срез 3D-матрицы; s – предварительная (нижняя) оценка длины маршрута, l – количество слоев (уровней) 3D-матрицы; cc() – двумерный массив, содержащий размерность по строкам и столбцам каждой матрицы, открытость/закрытость соответствующего ей узла дерева; smax – максимальная оценка нуля; imax – пункт отправления; jmax – пункт прибытия. Указанные выше, в качестве входных аргументов, переменные, сохраняют свои значения после работы соответствующей функции, что позволяет работать с ними в основной подпрограмме.

При разработке процедуры в VBA – Algorithm_Littla() – для хранения всех матриц расстояний каждого узла дерева используется трёхмерный массив. 3D-матрица (D от англ. dimension – «размерность») – трёхмерное множество элементов одного типа, имеющих общее имя. Если представить двухмерные матрицы в виде одной грани куба, то визуально можно показать, что эти 2D-матрицы накладываются одна на одну послойно, таким образом образуя из массив из двумерых массивов (рисунок 10).

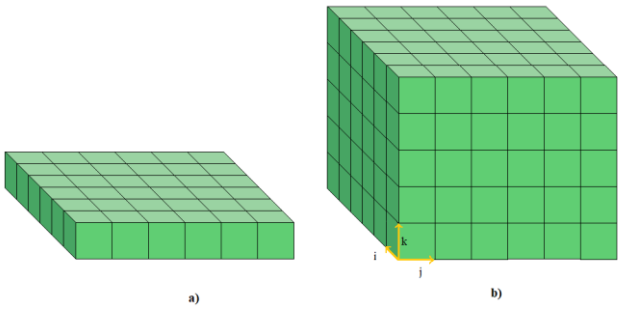

Рисунок 10 – Трёхмерная модель 3D-матрицы первого слоя (k = 0) (a) и 3D-матрица для 6-ти пунктов назначения (b)

Обращение к элементам 3D-матрицы происходит по тройному индексу: в качестве первого значения индекса (k) каждого элемента указывается номер слоя, второго значения индекса (i) – номер строки, третьего значения индекса (j) – номер столбца расположения элемента (рисунок 11).

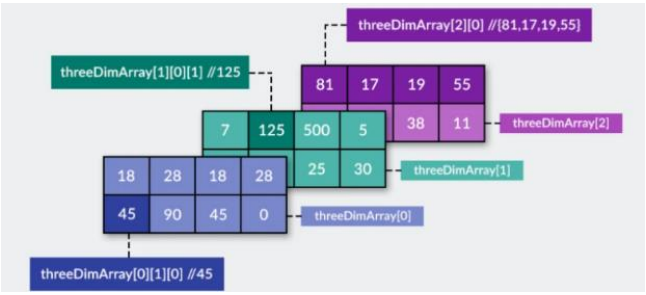

Рисунок 11 – Пример доступа к элементам трехмерного массива [6]

Блок-схема работы функции приращения текущей матрицы расстояний One() представлена на рисунке 12) и содержит следующие основные моменты:

- приращению подлежит матрица, соответствующая открытой в данный момент ветви дерева решений, с наименьшей оценкой;
- выбор слоя 3D-матрицы, содержащего матрицу с наименьшей оценкой, происходит в подпрограмме процедуре Algorithm_Littla() и передается в переменную t;
- значение оценки каждой 2D-матрицы расстояний хранится в соответствующих элементах с нулевыми индексами;
- поиск минимальных элементов производится по порядку расположения строк/столбцов, с соответствующим уменьшением каждой строки/столбца на выбранный минимальный элемент и увеличением оценки s на данный минимальный элемент.

Блок-схема работы функции для выбора нуля с максимальной оценкой в приведенной матрице расстояний Two() представлена на рисунке 13 и содержит следующие рабочие моменты:

- из рассмотрения и оценки исключается нулевой пункт;
- для подсчета промежуточной оценки нуля используется переменная sum;
- для запоминания номера строки и столбца матрицы, содержащих нуль, используются переменные i0 и j0 соответственно;
- при расположении нуля в первом столбце/первой строке – поиск минимального значения осуществляется со второго столбца/второй строки, в противном случае – с первого пункта;
- из поиска минимального значения в строке/столбце, при вычислении степени нулевых элементов, исключается оцениваемый нуль;
- для выбранного пункта отправления и выбранного пункта прибытия в приведенной матрице расстояний задается достаточно большое значение для запрета повторного отправления.

Блок-схема работы функции Three(), выполняющей удаление из матрицы расстояний строки и столбца с выбранными пунктами, представлена на рисунке 14, где следует отметить, что:

- из выбранного слоя 3D-матрицы с наименьшей оценкой значения передаются в текущую матрицу w();
- порядковые номера пунктов могут не совпадать с порядковыми номерами строк/столбцов матрицы, потому для индексирования строк, столбцов матрицы расстояний используются переменные *i*, *j*;
- для выбранных пунктов посещения транспортом в матрице расстояний устанавливается достаточно большое значение для устранения возможности повторного посещения;
- при уменьшении размерности текущей матрицы используется временный массив temp(), с количеством строк и столбцов меньшим на одну единицу, по сравнению с исходным, который после работы удаляется из памяти компьютера.

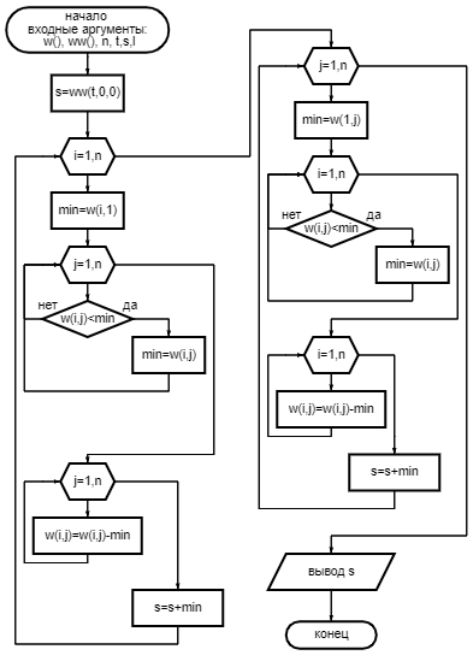

Рисунок 12 – Блок-схема работы функции приращения матрицы

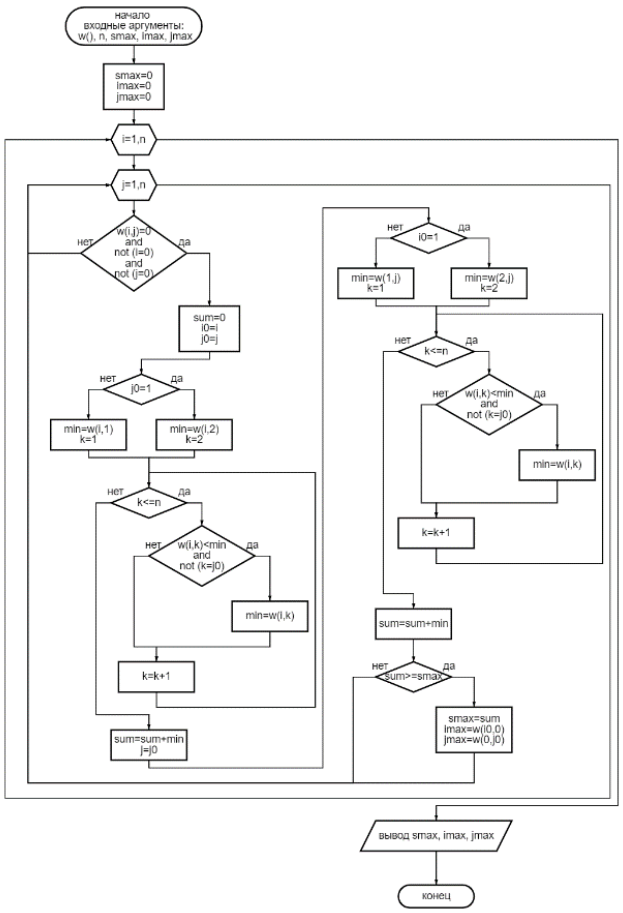

Рисунок 13 – Блок-схема работы функции оценки нулей

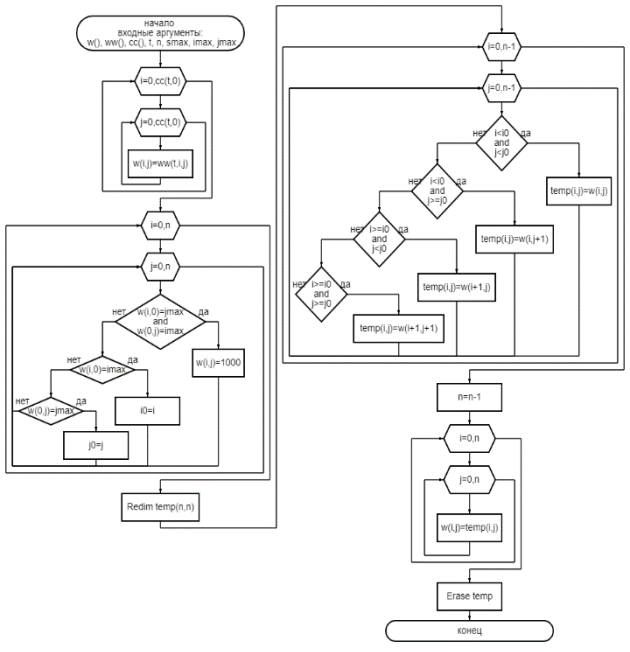

Рисунок 14 – Блок-схема функции для удаления строки и столбца, соответствующих пункту отправления и прибытия в матрице расстояний

Заключение

В наиболее популярных в Беларуси бесплатных картографических порталах, таких как Google Maps, Яндекс.Карты, построение кратчайшего маршрута возможно по двум пунктам и при добавлении дополнительных пунктов построение кратчайшего маршрута автоматически не осуществляется.

Разработанная в среде программирования VBA приложения ME подпрограмма Algorithm_Littla(), реализующая метод ветвей и границ в виде алгоритма Литтла, автоматически строит оптимальный маршрут для более 10-ти пунктов назначения. В настоящее время указанная подпрограмма проходит апробацию автоматизированного поиска кратчайшего маршрута спецавтотранспорта для вывоза твердых коммунальных отходов и вторичных материальных ресурсов на коммунальном предприятии г. Бреста в качестве инструмента повышения эффективности принятия решений организационного управления при решении задач рационального управления транспортными ресурсами.

Список цитированных источников

1. Land, A. H. An automatic method of solving discrete programming problems / A. H. Land, A. G. Doig // Econometrica. – No. 28. – 1960. – P. 497–520.
2. Эконометрика и экономико-математические методы и модели : учеб. пособие / Г. О. Читая [и др.] ; под ред. Г. О. Читая, С. Ф. Миксюк. – Минск : БГЭУ, 2018. – 511 с.
3. An algorithm for the traveling salesman problem / J. D. C. Little [et al.] // Operations Research. – No. 11 (6). – 1963. – P. 972–989.
4. Яндекс справка [Электронный ресурс] // Яндекс. – Режим доступа: https://yandex.ru/support/m-maps/index.html. – Дата доступа: 03.01.2022.
5. Марциновский, С. А. Реализация решения в вычислительных средах задачи составления расписания горячей обработки деталей / С. А. Марциновский, И. В. Тузик, Т. Г. Хомицкая // Современные проблемы математики и вычислительной техники : сборник материалов IX Республиканской научной конференции молодых ученых и студентов, Брест, 19–21 ноября 2015 года / Министерство образования Республики Беларусь, Брестский государственный технический университет ; редкол.: В. С. Рубанов [и др.]. – Брест : БрГТУ, 2015. – С. 50–52.
6. JavaRuch: Многомерные массивы [Электронный ресурс] // Java-университет. – Режим доступа: https://javarush.ru/groups/posts/-mnogomernye-massivy#Трёхмерные-массивы. – Дата доступа: 25.01.2022.

References

1. Land, A. H. An automatic method of solving discrete programming problems / A. H. Land, A. G. Doig // Econometrica. – No. 28. – 1960. – P. 497–520.
2. Ekonometrika i ekonomiko-matematicheskie metody i modeli : ucheb. posobie / G. O. Chitaya [i dr.] ; pod red. G. O. CHitaya, S. F. Miksyuk. – Minsk : BGEU, 2018. – 511 s.
3. An algorithm for the traveling salesman problem / J. D. C. Little [et al.] // Operations Research. – No. 11 (6). – 1963. – P. 972–989.
4. Yandeks spravka [Elektronnyj resurs] // Yandeks. – Rezhim dostupa: https://yandex.ru/support/m-maps/index.html. – Data dostupa: 03.01.2022.
5. Marcinovskij, S. A. Realizaciya resheniya v vychislitel'nyh sredah zadachi sostavleniya raspisaniya goryachej obrabotki detalej / S. A. Marcinovskij, I. V. Tuzik, T. G. Homickaya // Sovremennye problemy matematiki i vychislitel'noj tekhniki : sbornik materialov IX Respublikanskoj nauchnoj konferencii molodyh uchenyh i studentov, Brest, 19–21 noyabrya 2015 goda / Ministerstvo obrazovaniya Respubliki Belarus', Brestskij gosudarstvennyj tekhnicheskij universitet ; redkol.: V. S. Rubanov [i dr.]. – Brest : BrGTU, 2015. – S. 50–52.
6. JavaRuch: Mnogomernye massivy [Elektronnyj resurs] // Java-universitet. – Rezhim dostupa: https://javarush.ru/groups/posts/-mnogomernye-massivy#Tryohmernye-massivy. – Data dostupa: 25.01.2022.

Материал поступил в редакцию 04.05.2022

doi.org/10.36773/1818-1112-2022-128-2-168-172