

ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ ДЛЯ ЗАДАЧИ КОММИВОЯЖЕРА

Описывается методика применения динамического программирования для решения задачи коммивояжера. Излагается алгоритм, а также приводятся теоретические и экспериментальные оценки его временной сложности и требований к памяти ЭВМ.

1. Постановка задачи

Метод динамического программирования как способ глобальной оптимизации аддитивных и сводящихся к ним функционалов, автором которого является американский математик Р. Беллман, для решения задачи коммивояжера практически одновременно был предложен Р. Беллманом и М. Хелдом и Р. Карпом [1]. В дальнейшем мы следуем изложению Р. Беллмана.

Как известно, идея динамического программирования, опирающаяся на принцип оптимальности, состоит в том, что процесс оптимизации функционала рассматривается как последовательный n -шаговый процесс принятия решений, на каждом шаге которого принимается оптимальное решение. Для этого в рассмотрение вводится понятие состояний системы (задачи), относительно каждого из которых вычисляется оптимальное значение функционала, часто называемое значением функции Беллмана.

Смысл задачи коммивояжера состоит в том, что коммивояжер, выезжая из города n , в определенной последовательности посещает остальные $1, 2, \dots, n-1$ городов и возвращается в пункт выезда n . При этом требуется определить такой маршрут, который дает наименьшую стоимость объезда городов при условии, что коммивояжер не заезжает ни в один из них более одного раза, что в терминологии графов означает найти гамильтонов цикл наименьшей длины.

Применительно к этой задаче метод динамического программирования интерпретируется как n -шаговый процесс принятия решений, на каждом шаге которого коммивояжер должен определить оптимальный маршрут объезда оставшихся городов в зависимости от того, в каком из них он находится.

Предположим, коммивояжеру по пути наименьшей стоимости удалось достичь города i и до возвращения в город n осталось посетить k городов: j_1, j_2, \dots, j_k . Путь наименьшей стоимости из города i в город n через оставшиеся города j_1, j_2, \dots, j_k обозначим $V(n; j_1, j_2, \dots, j_k)$. Иными словами, так определим функцию Беллмана, значение которой устанавливается городом i и оптимальным порядком объезда оставшихся городов j_1, j_2, \dots, j_k . Следовательно, если будет найдено значение $V(n; j_1, j_2, \dots, j_{n-1})$, т. е. наименьшая стоимость объезда всех городов, задача коммивояжера будет решена. При этом очевидно, что состояниями в n -шаговом процессе принятия решений будут города $1, 2, \dots, n-1$. Например, в задаче с 5-ю городами коммивояжер из города 5 может выехать в любой из оставшихся $1, 2, \dots, 4$ городов, для которых и должны быть вычислены функции Беллмана.

Цель исследования – выяснение возможности использования на современных компьютерах метода динамического программирования для решения задачи коммивояжера.

Задача для достижения цели – представление функции Беллмана для решения указанной задачи.

2. Алгоритм решения

Действия алгоритма продемонстрируем на примере задачи с 5-ю городами и матрицей

$$\text{стоимости переездов из города в город: } C = \begin{vmatrix} \infty & 25 & 40 & 31 & 27 \\ 5 & \infty & 17 & 30 & 25 \\ 19 & 15 & \infty & 6 & 1 \\ 9 & 50 & 24 & \infty & 6 \\ 22 & 8 & 7 & 10 & \infty \end{vmatrix} [2].$$

Согласно концепции Р. Беллмана минимальная стоимость объезда городов окажется найденной, если на последнем 5-м шаге будет вычислено значение $V(5; j_1, j_2, j_3, j_4)$, т. е. определено направление наилучшего выезда из города 5. Так как из этого города коммивояжер может поехать в города 1, 2, 3, 4, а затем из каждого из них по оптимальному маршруту вернуться в город выезда, для 4-го шага принятия решений, рассматривая процедуру в обратном порядке (от 5-го к 1-му шагу), необходимо определить функции Беллмана для городов (состояний) 1, 2, 3, 4.

В связи с тем, что из города 1 коммивояжер в оптимальной последовательности должен проехать города j_2, j_3, j_4 , из города 2 города j_1, j_3, j_4 , из города 3 города j_1, j_2, j_4 и из города 4 города j_1, j_2, j_3 , функции Беллмана согласно $V(i; j_1, j_2, \dots, j_k)$ для этих состояний будут иметь вид $V(1; j_2, j_3, j_4)$, $V(2; j_1, j_3, j_4)$, $V(3; j_1, j_2, j_4)$, $V(4; j_1, j_2, j_3)$. На этом основании, учитывая, что $V(5; j_1, j_2, j_3, j_4)$ – минимальная стоимость объезда, для вычисления функции Беллмана на 5-м шаге процесса принятия решений получаем выражение

$$V(5; j_1, j_2, j_3, j_4) = \min \{ [c_{51} + V(1; j_2, j_3, j_4)], [c_{52} + V(2; j_1, j_3, j_4)], [c_{53} + V(3; j_1, j_2, j_4)], [c_{54} + V(4; j_1, j_2, j_3)] \}, \quad (1)$$

в котором $c_{51}, c_{52}, c_{53}, c_{54}$ – стоимость переездов из города 5 в города 1, 2, 3, 4, взятые из матрицы C .

В свою очередь значения функций Беллмана $V(1; j_2, j_3, j_4)$, $V(2; j_1, j_3, j_4)$, $V(3; j_1, j_2, j_4)$, $V(4; j_1, j_2, j_3)$, которые должны вычисляться на 4-м шаге принятия решений, с учетом того, что коммивояжер, выезжая из городов 1, 2, 3, 4, по оптимальному пути должен проехать три остальные города, определяются по выражениям:

$$\begin{aligned} V(1; j_2, j_3, j_4) &= \min \{ [c_{12} + V(2; j_3, j_4)], [c_{13} + V(3; j_2, j_4)], [c_{14} + V(4; j_2, j_3)] \}, \\ V(2; j_1, j_3, j_4) &= \min \{ [c_{21} + V(1; j_3, j_4)], [c_{23} + V(3; j_1, j_4)], [c_{24} + V(4; j_1, j_3)] \}, \\ V(3; j_1, j_2, j_4) &= \min \{ [c_{31} + V(1; j_2, j_4)], [c_{32} + V(2; j_1, j_4)], [c_{34} + V(4; j_1, j_2)] \}, \\ V(4; j_1, j_2, j_3) &= \min \{ [c_{41} + V(1; j_2, j_3)], [c_{42} + V(2; j_1, j_3)], [c_{43} + V(3; j_1, j_2)] \}. \end{aligned} \quad (2)$$

Очевидно, что вычислить эти выражения можно в том случае, если на 3-м шаге принятия решений найти значения функций Беллмана $V(1; j_2, j_3)$, $V(1; j_2, j_4)$, $V(1; j_3, j_4)$; $V(2; j_1, j_3)$, $V(2; j_1, j_4)$, $V(2; j_3, j_4)$; $V(3; j_1, j_2)$, $V(3; j_1, j_4)$, $V(3; j_2, j_4)$; $V(4; j_1, j_2)$, $V(4; j_1, j_3)$, $V(4; j_2, j_3)$. Так как приведенные записи этих функций означают, что, например, для города 1 они должны вычисляться на основании вариантов посещения коммивояжером городов в последовательности 2, 3 или 3, 2, городов 2, 4 или 4, 2 и, наконец, городов 3, 4 или 4, 3, выбор предпочтительного маршрута для каждого варианта определяется соответствующим выражением:

$$\begin{aligned} V(1; j_2, j_3) &= \min \{ [c_{12} + V(2; j_3)], [c_{13} + V(3; j_2)] \}, \\ V(1; j_2, j_4) &= \min \{ [c_{12} + V(2; j_4)], [c_{14} + V(4; j_2)] \}, \\ V(1; j_3, j_4) &= \min \{ [c_{13} + V(3; j_4)], [c_{14} + V(4; j_3)] \}. \end{aligned} \quad (3)$$

Рассуждая аналогично, получаем формулы (4), по которым вычисляются значения функций Беллмана для состояний 2, 3, 4:

$$\begin{aligned}
V(2; j_1, j_3) &= \min\{[c_{21} + B(1; j_3)], [c_{23} + B(3; j_1)]\}, \\
V(2; j_1, j_4) &= \min\{[c_{21} + B(1; j_4)], [c_{24} + B(4; j_1)]\}, \\
V(2; j_3, j_4) &= \min\{[c_{23} + B(3; j_4)], [c_{24} + B(4; j_3)]\}, \\
V(3; j_1, j_2) &= \min\{[c_{31} + B(1; j_2)], [c_{32} + B(2; j_1)]\}, \\
V(3; j_1, j_4) &= \min\{[c_{31} + B(1; j_4)], [c_{34} + B(4; j_1)]\}, \\
V(3; j_2, j_4) &= \min\{[c_{32} + B(2; j_4)], [c_{34} + B(4; j_2)]\}, \\
V(4; j_1, j_2) &= \min\{[c_{41} + B(1; j_2)], [c_{42} + B(2; j_1)]\}, \\
V(4; j_1, j_3) &= \min\{[c_{41} + B(1; j_3)], [c_{43} + B(3; j_1)]\}, \\
V(4; j_2, j_3) &= \min\{[c_{42} + B(2; j_3)], [c_{43} + B(3; j_2)]\}.
\end{aligned} \tag{4}$$

Из приведенных выражений следует, что их значения могут быть найдены, если на 2-м шаге принятия решений будут вычислены значения функций $V(1; j_2)$, $V(1; j_3)$, $V(1; j_4)$; $V(2; j_1)$, $V(2; j_3)$, $V(2; j_4)$; $V(3; j_1)$, $V(3; j_2)$, $V(3; j_4)$; $V(4; j_1)$, $V(4; j_2)$, $V(4; j_3)$. В свою очередь эти функции могут быть найдены, если учесть, что, например, $V(1; j_2)$, $V(1; j_3)$, $V(1; j_4)$ определяют оптимальные значения переезда коммивояжера из города 1 в города 2, 3, 4 и дальше из них в город 5, так как рассматривается 2-й шаг обратного процесса принятия решений.

Поэтому

$$\begin{aligned}
V(1; j_2) &= c_{12} + B(2; j_5), & V(1; j_3) &= c_{13} + B(3; j_5), & V(1; j_4) &= c_{14} + B(4; j_5); \\
V(2; j_1) &= c_{21} + B(1; j_5), & V(2; j_3) &= c_{23} + B(3; j_5), & V(2; j_4) &= c_{24} + B(4; j_5); \\
V(3; j_1) &= c_{31} + B(1; j_5), & V(3; j_2) &= c_{32} + B(2; j_5), & V(3; j_4) &= c_{34} + B(4; j_5); \\
V(4; j_1) &= c_{41} + B(1; j_5), & V(4; j_2) &= c_{42} + B(2; j_5), & V(4; j_3) &= c_{43} + B(3; j_5).
\end{aligned} \tag{5}$$

В связи с тем, что функции $V(1; j_5)$, $V(2; j_5)$, $V(3; j_5)$, $V(4; j_5)$, фигурирующие в этих выражениях, представляют собой стоимость переездов c_{15} , c_{25} , c_{35} , c_{45} , содержащихся в матрице C , процесс попятного принятия решений может быть начат с 1-го шага, сводящегося к перенесению указанных стоимостей в специальную таблицу, строки которой интерпретируются как номера шагов, а колонки определяют значения функций Беллмана.

На основании данных первой строки и выражений (5) получаем значения функций Беллмана для второй строки, по которым, используя выражения (3), (4), вычисляем значения функций третьей строки и т. д. Процесс вычислений и заполнения строк таблицы заканчиваем получением значения функции Беллмана для 5-го шага. Принятые обозначения и найденные значения приведены в табл. 1.

Из нее следует, что наименьшая стоимость объезда 5-и городов равна 62, что совпадает с результатом, полученным в [2] методом ветвей и границ, а также методом прямого перебора.

Для определения маршрута, обеспечивающего найденную стоимость, поступаем следующим образом. В выражении $V(5; j_1, j_2, j_3, j_4)$ найдем элемент, минимизирующий это выражение. Таким является элемент $c_{54} + V(4; j_1, j_2, j_3) = 10 + 52 = 62$. Это означает, что из города 5 коммивояжер выезжает в город 4 (показывают индексы компоненты c_{54}). Для того, чтобы определить, в какой город он направляется из города 4, необходимо в выражении для $V(4; j_1, j_2, j_3)$ найти элемент, доставляющий минимум этому выражению. Так как это элемент $c_{41} + V(1; j_2, j_3) = 9 + 43 = 52$, то из города 4 коммивояжер направляется в город 1. Минимум выражению $V(1; j_2, j_3)$ доставляет элемент $c_{12} + V(2; j_3) = 25 + 18 = 43$, из чего следует, что из города 1 коммивояжер выезжает в город 2. В свою очередь минимум $V(2; j_3) = c_{23} + V(3; j_5) = 17 + 1 = 18$. Следовательно, из города 2 коммивояжер

направляется в город 3, а из него возвращается в город 5. В результате получаем оптимальный маршрут объезда городов 5, 4, 1, 2, 3, 5, легко представляемый последовательностью дуг (5, 4), (4, 1), (1, 2), (2, 3), (3, 5) для удобной интерпретации на матрице стоимости переездов C . Дерево поиска решений и оптимальный маршрут на нем, выделенный жирными линиями со стрелками, показаны на рисунке.

Подводя итог изложенному выше, отметим, что алгоритм решения задачи коммивояжера методом динамического программирования, как и в большинстве случаев применения этого метода, состоит из двух этапов: вычисления оптимального значения функционала и на его основе определения последовательности оптимальных решений.

3. Оценки временной сложности алгоритма и требований к памяти ЭВМ

Теоретическую временную сложность алгоритма, т. е. зависимость времени решения задачи от ее размера n , определим традиционным путем: подсчитаем количество основных вычислительных операций H_n , которые должен выполнить алгоритм при решении задачи. Согласно совокупности его действий такими операциями являются операции сложения и сравнения. Подсчет суммарного количества этих операций выполним для каждого шага алгоритма отдельно.

На 2-м его шаге для получения значений функций Беллмана необходимо выполнить лишь операции сложения. Поэтому, поскольку число состояний задачи равно $n - 1$, всего согласно выражениям (5) для 2-го шага требуется выполнить $(n - 1) \cdot (n - 2)$ сложения.

На 3-м шаге алгоритма для вычисления значений функций Беллмана по каждому из $n - 1$ состояний согласно выражениям (3), (4) требуется вычислить последовательность минимальных значений для пар городов. При этом для каждой пары необходимо выполнить два сложения и одно сравнение. Общее же число операций, которое требуется выполнить для каждого состояния, определяется количеством пар городов, которое, как очевидно, зависит от размера задачи. Для определения этого числа, как функции n , обратимся к примерам.

В задаче из пяти городов 1, 2, 3, 4, 5, предполагая, что коммивояжер находится в городе 1, он может посетить города (2, 3) или (3, 2), города (2, 4) или (4, 2), города (3, 4) или (4, 3). Значения функций Беллмана для этого случая вычисляются по выражениям (3). Таким образом, размерность $n = 5$ определяет три пары (2, 3), (2, 4), (3, 4), т. е. общее количество пар $S = 1 + 2 = 3$.

Рассмотрим теперь задачу с шестью городами 1, 2, 3, 4, 5, 6. В этом случае пары городов, которые может посетить коммивояжер, находясь в городе 1, такие: (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5). В результате $S = 1 + 2 + 3 = 6$. По аналогии для задачи с семью городами получим такую последовательность пар: (2, 3), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 5), (4, 6), (5, 6), т. е. $S = 1 + 2 + 3 + 4 = 10$.

Как известно, общая формула для вычисления суммы ряда $1 + 2 + \dots + n$ такая: $S = \frac{n \cdot (n + 1)}{2}$. Учитывая, что в задаче n интерпретируется как число состояний, равное

$n - 1$, и отсчет пар начинается со второго элемента последовательностей городов, для числа пар по общей формуле получаем $S = \frac{[(n - 1) - 2] \cdot [(n - 1) - 1]}{2} = \frac{(n - 3) \cdot (n - 2)}{2}$. Учиты-

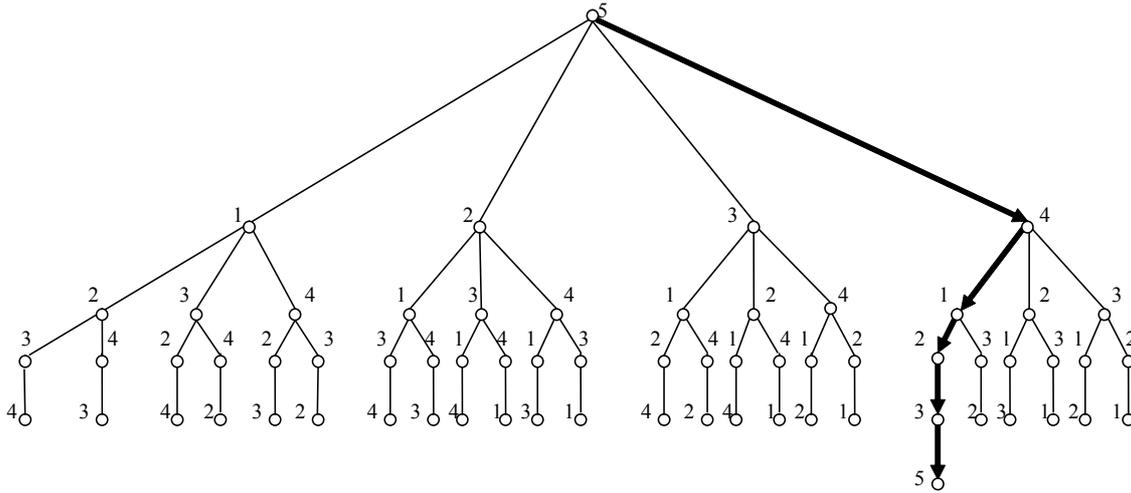
вая, что каждая пара требует выполнения трех основных операций и количество городов, для которых определяется S , равно $n - 1$, для числа основных операций третьего шага

получаем $H_3 = 3 \cdot S \cdot (n - 1) = \frac{3 \cdot (n - 3) \cdot (n - 2) \cdot (n - 1)}{2}$.

Как следует из примера, на четвертом шаге алгоритма по выражениям (2) требуется найти минимальные значения для последовательности троек городов, которые может посетить коммивояжер, находясь в каком-либо другом городе. Число таких троек, как и число пар, зависит от размера задачи.

Таблица 1. Обозначения и значения функции Беллмана

Обозначения и значения функции Беллмана													
$B(1;j_5)$	$B(2;j_5)$	$B(3;j_5)$	$B(4;j_5)$										
		1	6										
$B(1;j_2)$	$B(1;j_3)$	$B(1;j_4)$	$B(2;j_1)$	$B(2;j_3)$	$B(2;j_4)$	$B(3;j_1)$	$B(3;j_2)$	$B(3;j_4)$	$B(4;j_1)$	$B(4;j_2)$	$B(4;j_3)$		
				37	32	18	36	46	40	12	36	75	25
$B(1;j_2, j_3)$	$B(1;j_2, j_4)$	$B(1;j_3, j_4)$	$B(2;j_1, j_3)$	$B(2;j_1, j_4)$	$B(2;j_3, j_4)$	$B(3;j_1, j_2)$	$B(3;j_1, j_4)$	$B(3;j_2, j_4)$	$B(4;j_1, j_2)$	$B(4;j_1, j_3)$	$B(4;j_2, j_3)$		
				52	46	42	29	47	42	51	59	50	64
$B(1;j_2, j_3, j_4)$	$B(2;j_1, j_3, j_4)$			$B(3;j_1, j_2, j_4)$			$B(4;j_1, j_2, j_3)$						
				57			52						
$B(5;j_1, j_2, j_3, j_4)$	62												



Легко убедиться в том, что число пар городов определяется числом сочетаний C_{n-2}^2 . Тогда по аналогии число их троек равно C_{n-2}^3 , число четверок C_{n-2}^4 и т. д. Учитывая, что на последнем шаге алгоритма ищется минимальное значение стоимости маршрута для последовательности из $n-1$ городов, а на предпоследнем шаге – для последовательности из $n-2$ городов, получаем, что для этого шага число сочетаний равно $C_{n-2}^{n-2} = 1$. С другой стороны, так как на 3-м шаге алгоритма при минимизации необходимо выполнить два сложения и одно сравнение, на четвертом – три сложения и два сравнения, получаем, что для C_{n-2}^m потребуется выполнить $2 \cdot m - 1$ основных операций. На этом основании для C_{n-2}^{n-2} число таких операций равно $2 \cdot (n-2) - 1 = 2 \cdot n - 5$. Принимая во внимание, что на последнем шаге алгоритма необходимо найти минимум из $n-1$ элементов, в каждом из которых имеется операция сложения, получаем, что для этого шага необходимо выполнить $(n-2) + (n-1) = 2 \cdot n - 3$ основных операции.

Таким образом, на основании приведенных выкладок получаем, что общее число основных операций, которые должен выполнить алгоритм, решая задачу для n городов, может быть найдено по выражению:

$$\begin{aligned}
 H_n = & (n-1) \cdot (n-2) + (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + \\
 & + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 2 \cdot n - 5] + 2 \cdot n - 3 = (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + \\
 & + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 3 \cdot n - 7] + 2 \cdot n - 3,
 \end{aligned} \quad (6)$$

в котором $m = 2, 3, \dots, n-3$. К этому числу основных операций необходимо прибавить то их количество, которое потребуется выполнить при определении оптимального маршрута объезда городов на основе найденного значения функционала. Оно равно сумме, получаемой сложением числа операций каждого шага, заканчивая 2-м шагом и исключая n -й шаг, так как на этом шаге индекс элемента, доставляющий минимальное значение функционалу, легко фиксируется. Поэтому выражение для указанного числа операций имеет вид $[(n-2) + (n-3)] + [(n-3) + (n-4)] + \dots + 3 + 1 = (2 \cdot n - 5) + (2 \cdot n - 7) + \dots + 4$.

Таким образом, при двухэтапном процессе реализации алгоритма потребуется выполнить $H_n = (n-1) \cdot [(2 \cdot m - 1) \cdot C_{n-2}^2 + (2 \cdot m - 1) \cdot C_{n-2}^3 + \dots + (2 \cdot m - 1) \cdot C_{n-2}^{n-3} + 3 \cdot n - 7] + 2 \cdot n - 3 + (2 \cdot n - 5) + (2 \cdot n - 7) + \dots + 4$ основных операций.

Полученное выражение позволяет экспериментальным путем определить вычислительную сложность алгоритма. С этой целью для ряда размеров задачи $n = 5, 10, 15, 20, 25$ были вычислены значения H_n , представленные в табл. 2.

Таблица 2. Число основных вычислительных операций

n = 5	n = 10	n = 15	n = 20	n = 25
107	16190	1376803	106717205	4429186681

Данные табл. 2 приближены эмпирической формулой $H_n(n) = 4,0282e^{0,3912n}$, которая показывает, что при увеличении размера задачи число основных вычислительных операций, которые выполняет алгоритм, растет экспоненциально. При этом показатель экспоненты равен $0,3912 \cdot n$, а постоянный множитель – $4,0282$. Поэтому согласно этой формуле, например, при $n = 60$ число операций окажется примерно таким $H_n(60) = 4,0282 \cdot e^{0,3912 \cdot 60} = 4,0282 \cdot e^{23,472} = 62931937743$, с которым компьютер с тактовой частотой центрального процессора 4 ГГц справится не более, чем за 20 с.

Память ЭВМ при увеличении размера задачи растет как количество столбцов табл. 1, умноженное на количество ее строк $n - 1$, предназначенных для хранения значений функций Беллмана при условии, что для этих целей в простейшем случае используется матрица. В свою очередь количество столбцов определяется числом состояний $n - 1$, умноженным на

число пар операций $S = \frac{(n-2) \cdot (n-3)}{2}$, которые необходимо выполнить на 3-м шаге алгоритма. Таким образом, требуемая память определяется формулой

$P(n) = (n-1) \cdot (n-1) \cdot S = \frac{(n-1)^2 \cdot (n-2) \cdot (n-3)}{2}$, из которой следует, что при увеличении размера задачи n она растет как полином четвертой степени n .

Для $n = 60$ значение этого полинома равно 5754093. Умножив его на 4, получим число байт памяти ЭВМ, равное 23016372, т. е. около 22 МБ, которые необходимы для хранения значений функции Беллмана при решении задачи указанного размера. Очевидно, что для компьютеров, располагающих оперативной памятью, например, 4 ГБ, указанное число достаточно просто помещается в этой памяти.

Выводы

Таким образом, на основании анализа и расчетов получены новые результаты, показывающие объективную возможность решения задачи коммивояжера с числом городов 60 – 65 методом динамического программирования на современных персональных компьютерах за практически приемлемое время (не более 30 с).

Научная новизна работы – сконструирован алгоритм, получены оценки его временной сложности, которые показывают возможность его применения на современных компьютерах. *Практическая значимость* – алгоритм легко программируем, после чего он может быть применен на практике.

Список литературы: 1. Беллман Р. Применение динамического программирования к задаче о коммивояжере // Кибернетический сборник. М.: Мир, 1964. Т. 9. С. 219 – 228. 2. Гудман С., Хиднеттеми С. Введение в разработку и анализ алгоритмов. М.: Мир, 1981. 368 с.

Поступила в редколлегию 04.03.2014

Канцедал Сергей Андреевич, д-р техн. наук, профессор кафедры прикладной математики Западодонбасского института Межрегиональной академии управления персоналом. Научные интересы: математическое моделирование, теория расписаний и ее применение. Адрес: Украина, 51400, Днепропетровская обл., Павлоград, ул. Днепровская, 400/1.

Костикова Марина Владимировна, канд. техн. наук, доцент кафедры информационных технологий и мехатроники Харьковского национального автомобильно-дорожного университета. Научные интересы: математическое моделирование, теория расписаний и ее применение. Адрес: Украина, 61002, Харьков, ул. Петровского, 25, тел. 707-37-74.