

УДК 519.17

МЕТОД ЛИТТЛА С ОПТИМАЛЬНЫМ ПРИВЕДЕНИЕМ МАТРИЦЫ

И.Ф. Борханов, В.Р. Фазылов

Аннотация

В работе рассматривается применение традиционного и оптимального приведений матрицы стоимостей в методе Литтла для несимметричной задачи коммивояжера. На ряде тестовых задач показана высокая эффективность оптимального приведения.

1. Традиционное и оптимальное приведения

Как известно, в методе Литтла, Мэрти, Суини, Кэрролл для задачи коммивояжера (см., например, [1, с. 369], [2, с. 78]) используется процедура приведения матрицы стоимостей (весов дуг) W порядка n , которая представляет собой последовательность операций приведения строк и столбцов. Операция приведения i -й строки заключается в вычитании из строки матрицы некоторой константы r_i – константы приведения i -й строки. Аналогично, операция приведения j -го столбца заключается в вычитании из столбца матрицы некоторой константы c_j – константы приведения j -го столбца.

Целью применения процедур приведения по строкам и столбцам является получение *приведенной матрицы* – неотрицательной матрицы стоимостей W' ($w'_{ij} = w_{ij} - r_i - c_j$, $i = 1, \dots, n$, $j = 1, \dots, n$), каждая строка и каждый столбец которой содержат хотя бы один нулевой элемент. При этом стоимость любого решения исходной задачи будет отличаться от стоимости этого же решения, вычисленного по приведенной матрице, ровно на сумму констант приведения. А так как очевидная нижняя оценка стоимости любого маршрута для приведенной матрицы равна нулю, то нижняя оценка стоимости любого маршрута для исходной матрицы будет равна сумме констант приведения, что и используется в методе Литтла.

В работе [3] отмечено, что простые способы приведения матрицы стоимостей обычно не дают максимума суммы констант приведения, и предложена процедура оптимального приведения. Опишем далее традиционное и оптимальное приведения матрицы стоимостей.

Традиционное приведение. Процедура заключается в следующем: сначала из каждой строки матрицы вычитается минимальный элемент строки, а потом из каждого столбца получившейся матрицы вычитается минимальный элемент столбца. При этом, очевидно, все константы приведения будут неотрицательными, и в каждой строке и в каждом столбце будут нулевые элементы.

Оптимальное приведение. Отыскание оптимальных констант приведения состоит в решении следующей задачи:

$$\max \left(\sum_{i=1}^n r_i + \sum_{j=1}^n c_j \right), \quad (1)$$

$$r_i + c_j \leq w_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, n. \quad (2)$$

Для решения задачи (1), (2) воспользуемся алгоритмом Манкреса для задачи о назначениях (см. [4, 5]):

1. В каждой строке матрицы найти минимальный элемент строки и вычесть его из каждого элемента этой строки. Найденные минимальные элементы строк будут начальными значениями констант приведения строк.

2. В каждом столбце полученной матрицы найти наименьший элемент столбца и вычесть его из каждого элемента этого столбца. Найденные минимальные элементы столбцов будут начальными значениями констант приведения столбцов.

3. В полученной матрице найти нулевой элемент (Z). Если нет помеченного звездочкой нуля в его строке и столбце, то помечаем Z звездочкой. Повторить это для всех нулей матрицы.

4. Закрыть каждый столбец, содержащий нуль со звездочкой. Если закроются все столбцы, то останов: нули со звездочками определяют оптимальное решение задачи о назначениях.

5. Если в матрице все нули закрыты, то перейти к п. 7, иначе найти незакрытый нуль и пометить его штрихом. Если в строке с этим нулем нет нуля со звездочкой, то перейти к п. 6, иначе закрыть эту строку и открыть столбец, содержащий нуль со звездочкой. Перейти к началу п. 5.

6. Построить ряд чередующихся помеченных штрихом и звездочкой нулей следующим образом.

6.1. Выбрать за Z незакрытый помеченный штрихом нуль, найденный в п. 5.

6.2. Поместить Z в конец ряда. Если в столбце с Z есть нуль со звездочкой, то выбрать его за Z , иначе перейти к п. 6.4.

6.3. Поместить Z в конец ряда. В строке с Z выбрать нуль со штрихом и выбрать его за Z . Перейти к п. 6.2.

6.4. Снять звездочку с нулей со звездочкой в ряду, пометить звездочкой нули со штрихом в ряду, удалить все оставшиеся штрихи и открыть все строки и столбцы матрицы. Перейти к п. 4.

7. Найти из незакрытых элементов матрицы минимальный, обозначить его через A . Добавить A к каждому элементу закрытой строки, и вычесть A из каждого открытого столбца, уменьшить на A значения констант приведения соответствующих строк и увеличить на A значения констант приведения соответствующих столбцов. Перейти к п. 5 (без изменений звездочек, штрихов и признаков закрытых строк и столбцов матрицы).

Очевидно, что найденные этим алгоритмом константы приведения строк и столбцов согласно теореме 3 [3] будут решением задачи (1), (2) (то есть оптимальными константами приведения), так как они удовлетворяют условию (8) из [3].

Заметим, что одновременно с константами приведения получается оптимально приведенная матрица и оптимальное решение задачи о назначениях, которое может оказаться решением задачи коммивояжера. Проверка этого имеет сложность $O(n)$ и не представляет большого труда, но в случае успеха сокращает трудоемкость получения решения задачи коммивояжера.

Применение алгоритма Манкреса для решения задачи (1), (2) вместо симплекс-метода обусловлено тем, что первый имеет сложность $O(n^3)$, тогда как сложность только одной итерации симплекс-метода для задачи (1), (2) равна $O(n^4)$.

Проиллюстрируем применение описанных выше процедур приведения на простом примере. Пусть задана матрица W (знак ∞ означает запрет элемента):

$$\begin{pmatrix} \infty & 7 & 1 \\ 7 & \infty & 1 \\ 5 & 5 & \infty \end{pmatrix}.$$

Традиционное приведение даст следующие константы приведения: $r_1 = 1$, $r_2 = 1$, $r_3 = 5$, $c_1 = 0$, $c_2 = 0$, $c_3 = 0$, их сумма равна 7, а приведенная матрица имеет вид:

$$\begin{pmatrix} \infty & 6 & 0 \\ 6 & \infty & 0 \\ 0 & 0 & \infty \end{pmatrix}.$$

Оптимальное приведение даст следующие константы приведения: $r_1 = 1$, $r_2 = 1$, $r_3 = -1$, $c_1 = 6$, $c_2 = 6$, $c_3 = 0$, их сумма равна 13, а приведенная матрица имеет вид:

$$\begin{pmatrix} \infty & 0 & 0 \\ 0 & \infty & 0 \\ 0 & 0 & \infty \end{pmatrix}.$$

В дальнейшем через W' будем обозначать оптимально приведенную матрицу. *Назначением* будем называть любой набор элементов матрицы, задающий допустимое решение задачи о назначениях для матрицы W' , *оптимальным назначением* будем называть назначение с нулевой суммой элементов. В матрице W' , как доказано в [3], существует хотя бы одно оптимальное назначение с нулевой суммой выбранных элементов.

Назовем *оценкой элемента матрицы* сумму констант приведения матрицы, полученной из исходной запретом этого элемента.

Теорема 1. *Элемент оптимально приведенной матрицы W' , имеющий положительную оценку, входит во все оптимальные назначения.*

Доказательство. Пусть оценка элемента w'_{ij} положительна и существует оптимальное назначение, не содержащее w'_{ij} . Тогда в матрице \widetilde{W} , полученной из W' запретом элемента w'_{ij} , это назначение сохранится. А так как все элементы этого назначения равны нулю, то матрица \widetilde{W} будет приведенной, и следовательно, оценка элемента w'_{ij} будет равна нулю. Полученное противоречие завершает доказательство. \square

Таким образом, согласно теореме 1 поиск элемента с максимальной оценкой в методе Литтла можно проводить только среди элементов оптимального назначения, найденного с помощью алгоритма Манкреса. Это упрощает поиск элемента с максимальной оценкой.

Заметим, что в матрице W' может и не быть элемента с положительной оценкой. Но и в этом случае мы рекомендуем выбирать искомый элемент из элементов оптимального назначения.

2. Тестовые расчеты

Расчеты проводились на компьютере со следующими характеристиками: процессор Pentium 4 3.15 GHz, объем оперативной памяти 512 Mb.

В численном эксперименте решение задачи коммивояжера проводилось следующими алгоритмами:

1 (DEF). Алгоритм Литтла с традиционной процедурой приведения.

2 (APS). Алгоритм Литтла с оптимальной процедурой приведения. Оптимальное приведение используется при приведении исходной матрицы и при каждой фиксации элемента (левые ветви дерева перебора решений). При запрете элемента (правые ветви дерева перебора решений) используется традиционное приведение строки и столбца, соответствующих запрещенному элементу. Поиск элемента с максимальной оценкой проводится среди всех нулевых элементов матрицы.

3 (APSA). Алгоритм APS с поиском элемента с максимальной оценкой (элемент для ветвления) только среди элементов оптимального назначения, найденного алгоритмом Манкреса при оптимальном приведении матрицы. При запрете элемента после традиционного приведения оптимальное назначение неизвестно, и поиск элемента с максимальной оценкой производится среди всех нулей матрицы.

4 (APSAC). Алгоритм APSA с проверкой, не является ли очередное оптимальное назначение допустимым решением задачи коммивояжера.

Для оценки результата решения тестовых задач фиксировались следующие показатели:

1. Время T_i , затраченное на решение тестовой задачи в миллисекундах (i – номер алгоритма);
2. Количество просмотренных узлов E_i в дереве перебора решений метода Литтла;
3. Трудоемкость по времени алгоритмов 2–4 в сравнении с алгоритмом 1 (T_1/T_i , $i = 2, 3, 4$);
4. Трудоемкость по количеству просмотренных узлов алгоритмов 2–4 в сравнении с алгоритмом 1 (E_1/E_i , $i = 2, 3, 4$).

Несимметричные задачи коммивояжера

Генерировались случайные несимметричные матрицы порядка n от 10 до 100 с шагом 1. Диапазон значений элементов – от 0 до $500000000/n$. Для каждого n решено более 200 задач.

В табл. 1 приведены средние значения показателей T_i и E_i для задач с одинаковым порядком матрицы, шаг по n в таблице равен 5. На рис. 1 приведены средние значения показателя T_i для задач с матрицами одного порядка, для построения графика использованы полные данные эксперимента, полученные с шагом по n , равным 1.

Табл. 1

n	DEF		APS		APSA		APSAC	
	T_1	E_1	T_2	E_2	T_3	E_3	T_4	E_4
10	0.1	18.6	0.1	11.8	0.1	11.8	0.1	6.8
15	0.1	60.2	0.1	25.3	0.1	25.3	0.1	16.5
20	0.2	147.2	0.1	37.7	0.1	37.7	0.1	26.7
25	0.8	432.2	0.4	60.7	0.6	60.7	0.4	46.5
30	4.5	995.6	0.9	75.6	1	75.6	1.2	59.5
35	18	2774.3	0.9	107.9	1.7	107.9	1.4	88.3
40	41.5	5330.4	2.3	136.3	2.5	136.3	2.2	112
45	165.7	18450.8	2.4	156	3.5	156	3.4	131.3
50	432	40359.2	4.6	180.4	5	180.4	4.1	151.1
55	869.1	69021.7	6.7	195.2	6.1	195.2	6.1	166.1
60	2466.4	173449.7	10.4	250.5	8.9	250.5	8.9	214.8
65	8298.8	496245.4	13.5	294.9	13.3	294.9	11.9	256
70	22227.1	1143516.5	17.7	319	16.6	319	15.3	279.3
75	83209.4	3865399.9	24.1	345.2	22.8	345.2	22	301.9
80	182205.7	7677165.3	28.7	366.4	26.9	366.4	27	322.2
85	323268.7	12393663.5	42.1	448.7	40.5	448.7	39.6	397.3
90	544506	18988318.7	51.1	507.9	49.8	507.9	48.9	457.7
95	970715.9	30820562.1	61.5	538.2	62	538.2	60.9	481.3
100	1117811.5	32222002	80.3	644.1	80.3	644.1	78.5	585.4

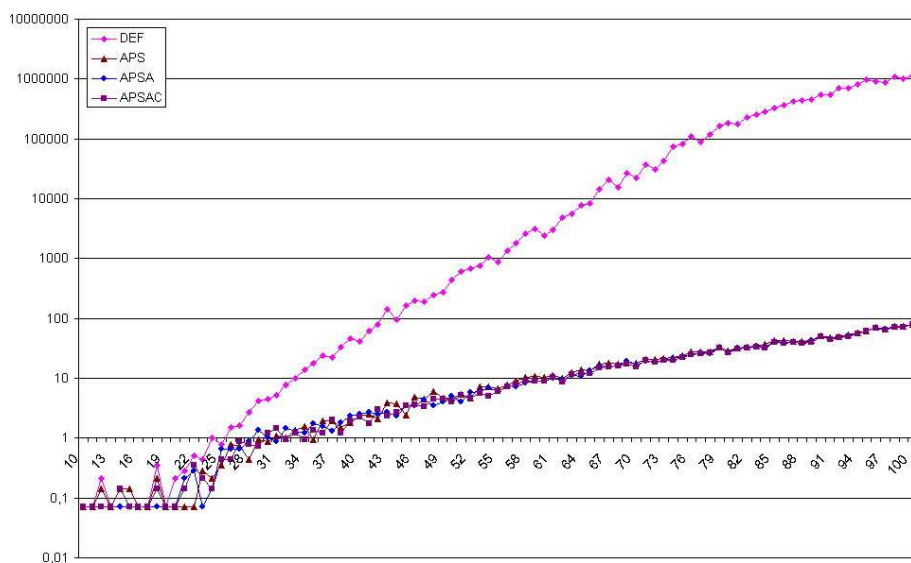


Рис. 1. Зависимость средней трудоемкости алгоритмов от порядка матрицы

В табл. 2 приведены средние значения показателей T_1/T_i и E_1/E_i для задач с одинаковым порядком матрицы, шаг по n в таблице равен 5. На рис. 2 приведены средние значения показателя T_1/T_i для задач с матрицами одного порядка, для построения графика использованы полные данные эксперимента, полученные с шагом по n , равным 1.

Табл. 2

n	APS		APSA		APSAC	
	T_1/T_2	E_1/E_2	T_1/T_3	E_1/E_3	T_1/T_4	E_1/E_4
10	0.1	1.6	0.1	1.6	0.1	5.6
15	0.1	2.7	0.1	2.7	0.1	9.6
20	0.2	4.2	0.2	4.2	0.2	15.9
25	0.8	9.6	0.8	9.6	0.8	33.6
30	4.4	17.8	4.3	17.8	4.2	83.3
35	16.9	33.6	16.7	33.6	16.2	75.4
40	37	54.6	32.1	54.6	34.4	225.4
45	133.6	210	123.3	210	132.4	3137.2
50	245.2	316.9	265.3	316.9	268.3	1845.2
55	513.5	495.7	525.1	495.7	447.8	1775
60	1152.4	1259.5	882	1259.5	1081.9	11749.3
65	1752	2884.8	2567.4	2884.8	2202.8	7313.8
70	4559.6	6387	4129.1	6387	4770.9	63128.6
75	7859.9	14478.5	6630.4	14478.5	5938.9	22204.7
80	12549.9	34592.4	12920.1	34592.4	14957.1	290688.8
85	10098.1	44845.4	10904.2	44845.4	15011.5	448510.1
90	17637.5	78716.6	18005.4	78716.6	19623.9	393283.8
95	24388.2	133898.8	26006.7	133898.8	26371.4	612443.6
100	22096.4	108588.7	21641.3	108588.7	26679.4	404530.4

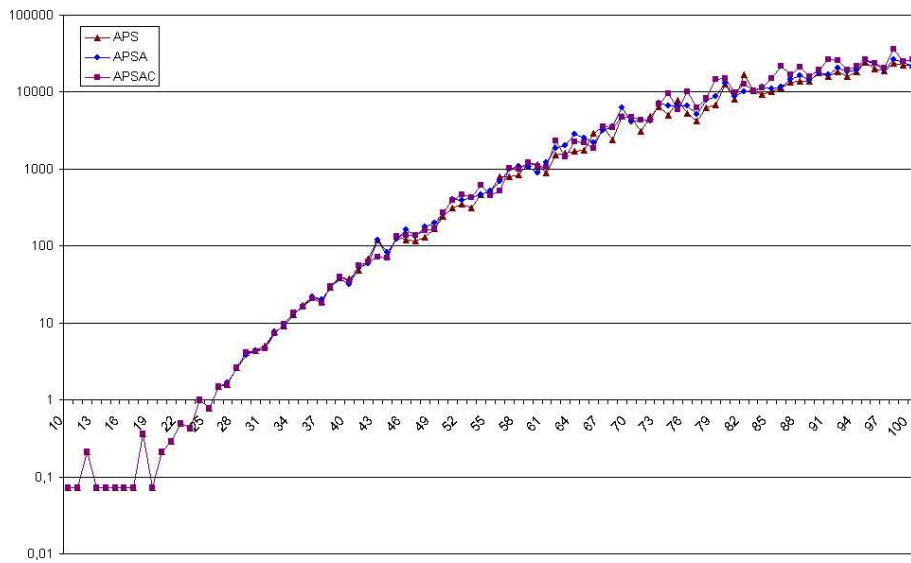


Рис. 2. Зависимость средних значений относительной трудоемкости алгоритма с традиционным приведением и алгоритмов с оптимальным приведением от порядка матрицы

Симметричные задачи коммивояжера

Генерировались случайные симметричные матрицы порядка n от 10 до 60 с шагом 1. Элемент матрицы w_{ij} вычислялся как евклидово расстояние между точками (x_i, y_i) , (x_j, y_j) , координаты которых выбирались случайно из диапазона от 0 до $250000000/n$. Для каждого n решено более 200 задач. Заметим, что для тестовых расчетов использовались алгоритмы 1–4 без каких-либо дополнений, учитывающих симметричность матрицы.

Табл. 3

n	DEF		APS		APSA		APSAC	
	T_1	E_1	T_2	E_2	T_3	E_3	T_4	E_4
10	0.2	47	0.5	28	0.1	28	0.1	24
15	1.2	212	0.6	95	1.2	95	0.9	85
20	5.4	985	3.5	252	3.4	252	3.2	235
25	28.1	3958	13.4	598	13.2	598	12.7	574
30	152.1	16401	39.6	1342	38.4	1342	37.6	1312
35	721	62339	119	2688.4	114	2688	114.2	2644
40	3430.1	244500	359.5	5969	344	5969	346.1	5916
45	15042.3	925826	1231.6	16826	1183.6	16826	1187.8	16757
50	74466.4	3893821	3280.6	31142	3132.7	31142	3164.2	31058
55	356217	16225433	7822.3	56549	7572.7	56549	7621	56454
60	914692	35522112	22827	132753	22588	132753	22429	132643

В табл. 3 приведены средние значения показателей T_i и E_i для задач с одинаковым порядком матрицы, шаг по n в таблице равен 5. На рис. 3 приведены средние значения показателя T_i для задач с матрицами одного порядка, для построения графика использованы полные данные эксперимента, полученные с шагом по n , равным 1.

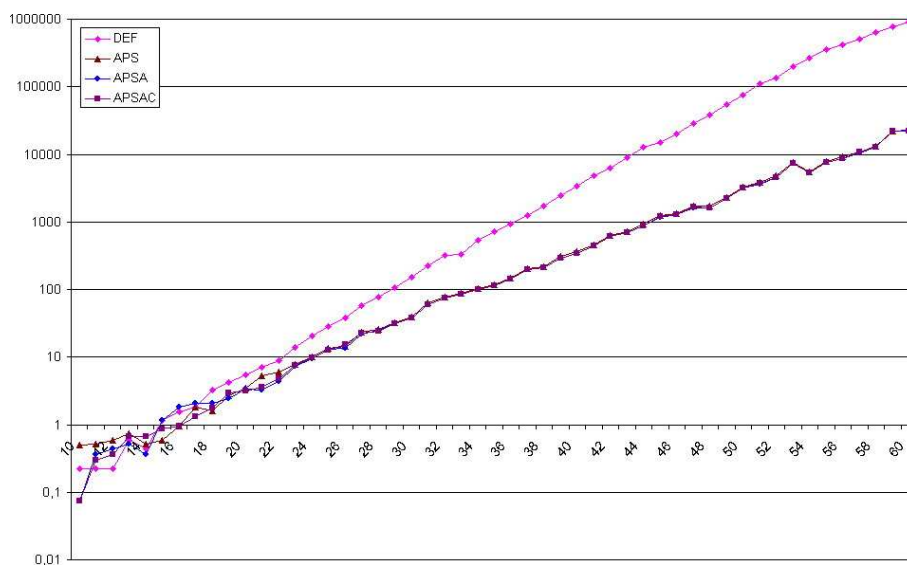


Рис. 3. Зависимость средней трудоемкости алгоритмов от порядка матрицы

В табл. 4 приведены средние значения показателей T_1/T_i и E_1/E_i для задач с одинаковым порядком матрицы, шаг по n в таблице равен 5. На рис. 4 приведены средние значения показателя T_1/T_i для задач с матрицами одного порядка, для построения графика использованы полные данные эксперимента, полученные с шагом по n , равным 1.

Табл. 4

n	APS		APSA		APSAC	
	T_1/T_2	E_1/E_2	T_1/T_3	E_1/E_3	T_1/T_4	E_1/E_4
10	0.2	1.9	0.2	1.9	0.2	2.7
15	1.2	2.9	1.2	2.9	1.1	3.4
20	4.9	4.9	4.3	4.9	4.3	5.5
25	9.9	9.2	7.4	9.2	8.5	10.9
30	9.4	18.8	9.9	18.8	9.5	19.8
35	8.6	34.9	9.1	34.9	9.8	36.3
40	16.1	72.7	16.7	72.7	16.7	74.2
45	25.6	137.1	26.6	137.1	26.7	139.4
50	41.4	238.4	43.3	238.4	43.1	241.2
55	115.5	735.5	119.6	735.5	119	740.3
60	99.5	709.3	101.3	709.3	102.3	711.6

Как для несимметричных, так и для симметричных случайных тестовых задач результаты численных экспериментов показывают, что применение оптимального приведения дает значительный выигрыш не только по количеству просмотренных узлов, но и по времени решения задачи, причем этот выигрыш возрастает с ростом размерности задачи (см. табл. 2, 4, и рис. 2, 4). Отсутствие заметного преимущества на малых размерностях (до 25 пунктов в несимметричных задачах и до 15 пунктов в симметричных задачах) не удалось выявить из-за трудности достаточно точной фиксации времени решения ввиду его малости.

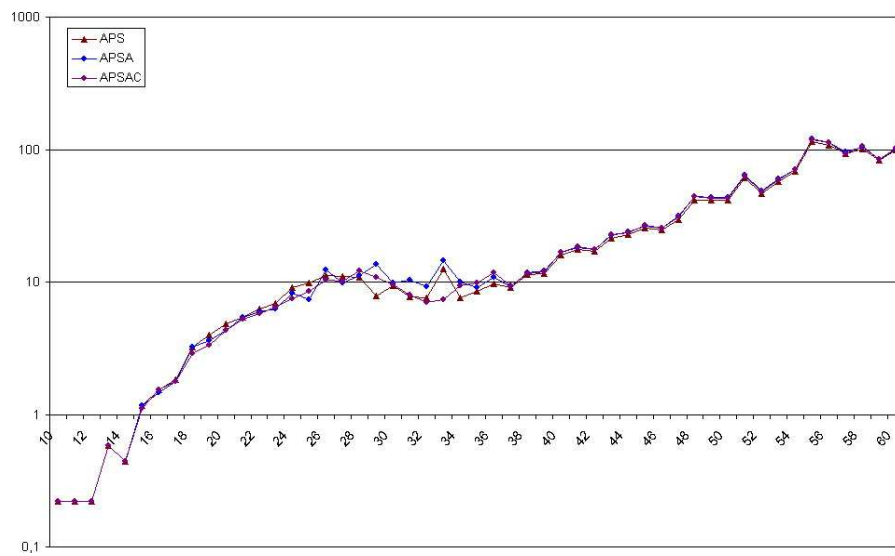


Рис. 4. Зависимость средних значений относительной трудоемкости алгоритма с традиционным приведением и алгоритмов с оптимальным приведением от порядка матрицы

Табл. 5

Имя задачи	n	DEF	APS	APSA	APSAC
		T_1	T_2	T_3	T_4
ft70	70		160328	147265	149187
ftv33	34	156	1046	968	1046
ftv35	36	78	46	31	46
ftv38	39	312	46	31	46
ftv44	45	1500	93	78	78
ftv47	48	20968	437	406	421
ftv55	56	615500	2640	2437	2437
ftv64	65	951750	2187	1984	2015
ftv70	71		9515	8203	8390
ftv90	91		1609	1484	1500
ftv130	131		130813	114265	116484
ftv140	141		122593	109578	108375
ftv150	151		152562	132078	134531
ftv160	161		4045046	3771093	3970437
rbg323	323		20968	3015	2968
rbg358	358		86515	7140	7328
rbg403	403		128062	5781	5484
rbg443	443		73906	4328	3578
td316	317			11890	11921
td1000	1001		1237515	1245875	1227671
br17	17	5078	6140	5437	5625
kro124p	100		3688984	3505406	3568203
ry48p	48	571140	57765	54406	54406
atex1	16	234	437	390	406
atex3	32	281	62	46	46
code198	198	234	171	78	62
code253	253		1818203	1255640	1261203

Табл. 6

Имя задачи	n	DEF	APS	APSA	APSAC
		E_1	E_2	E_3	E_4
ft70	70		3421546	3421546	3421486
ftv33	34	20737	217020	217096	217035
ftv35	36	8496	3562	3562	3544
ftv38	39	28328	2123	2123	2093
ftv44	45	116002	3013	3010	2990
ftv47	48	1814510	13205	13123	13117
ftv55	56	34685374	90719	88640	88611
ftv64	65	48391535	62403	59254	59236
ftv70	71		1422764	142949	142860
ftv90	91		8075	8075	8017
ftv130	131		3393901	3520841	3520771
ftv140	141		299846	298833	298760
ftv150	151		3744592	3431861	3431649
ftv160	161		20704188	20278643	20278579
rbg323	323		8459	322	242
rbg358	358		31802	357	322
rbg403	403		40598	402	248
rbg443	443		11924	442	81
td316	317			316	315
td1000	1001		3060	1000	843
br17	17	1909572	1301491	1256528	1256524
kro124p	100		11884544	11884544	11884029
ry48p	48	407404475	2296581	2296533	2296533
atex1	16	80013	98322	98373	98357
atex3	32	43633	2229	2253	2225
code198	198	421	197	197	196
code253	253		559582	559582	559299

Известные тестовые задачи

Эти задачи используются для тестирования алгоритмов решения задачи коммивояжера и свободно доступны в сети интернет. С описанием задач и их происхождением можно ознакомиться, например, в статье [8].

В табл. 5 и 6 приведены значения показателей T_i и E_i соответственно. Пустые клетки означают, что задача не была решена соответствующим алгоритмом за отведенных два часа на решение одной задачи.

Как видно из приведенных результатов, основной вывод, сделанный ранее для несимметричных и для симметричных случайных задач, — эффективность применения оптимального приведения — подтверждается и для решенных известных тестовых задач. В некоторых случаях, когда при традиционной реализации метода Литтла не удавалось найти решение за отведенных два часа, алгоритмы с оптимальным приведением находили решение.

Отметим особенно результаты решения задачи td316. Здесь интересно то, что применение только оптимального приведения не дало решение за отведенных два часа. Однако алгоритмы с оптимальным приведением и поиском элемента с максимальной оценкой только среди элементов оптимального назначения, найденного алгоритмом Манкреса при оптимальном приведении матрицы (алгоритмы APSA, APSAC) нашли решение менее чем за 12 секунд. На наш взгляд, это объясняется особой эффективностью выбора элемента для ветвления из текущего назначения.

В заключение отметим, что ряд тестовых задач – ft53 (53 – количество пунктов), ftv100 (101), ftv110 (111), ftv120 (121), ftv170 (171), td100 (101), p43 (43), atex4 (48), atex5 (72), atex8 (600), big702 (702), dc112 (112), dc126 (126), dc134 (134), dc176 (176), dc188 (188), dc563 (563), dc849 (849), dc895 (895), dc932 (932) – не были решены за отведенных два часа ни одним из методов, причем среди них есть задачи небольшой размерности (например, ft53, p43, atex4, atex5). По-видимому, для разработки хороших алгоритмов для их решения требуется изучение специфики этих задач.

Summary

I.F. Borhanov, V.R. Fazylov. Little's method with optimal reduction of a matrix.

In this paper are considered application traditional and optimum reduction of a matrix in Little's method for asymmetrical problem of the traveling salesman representative. On a number of test problems high efficiency of optimum reduction is shown.

Литература

1. Аюф Р., Сасиени М. Основы исследования операций. – М.: Мир, 1971. – 534 с.
2. Конвей Р.В., Максвелл В.Л., Миллер Л.В. Теория расписаний. – М.: Наука, 1975. – 359 с.
3. Борханов И.Ф., Фазылов В.Р. Об оптимальном приведении матрицы стоимостей // Уч. зап. Казан. ун-та. Сер. Физ.-матем. науки. – 2006. – Т. 148, кн. 2. – С. 18–22.
4. Munkres J. Algorithms for the assignment and transportation problems // J. SIAM. – 1957. – V. 5, No 1. – P. 32–38.
5. Jin Kue Wong. A new implementation of an algorithm for the optimal assignment problem: an improved version of Munkres' algorithm // BIT Numerical Mathematics. – V. 19, No 3. – P. 418–424.
6. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978. – 432 с.
7. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Теория и практика. – М.: Мир, 1970. – 476 с.
8. Cirasella J., Johnson D.S., McGeoch L.A., Zhang W. The asymmetric travelling salesman problem: algorithms, instance generators, and tests // ALENEX. – 2001. – V. 2153. – P. 32–59.

Поступила в редакцию
08.09.06

Борханов Ильнар Фирсович – аспирант кафедры экономической кибернетики Казанского государственного университета.

Фазылов Валерий Рауфович – доктор физико-математических наук, заведующий кафедрой экономической кибернетики Казанского государственного университета.

E-mail: *Valery.Fazylov@ksu.ru*