

УДК 338

ВЫБОР ОПТИМАЛЬНОГО ПУТИ В ТРАНСПОРТНОЙ СЕТИ (ЗАДАЧА КОММИВОЯЖЁРА)

© Николай Викторович Катаргин

Финансовый университет при Правительстве Российской Федерации, Москва, Россия
nnnkkk@yandex.ru

Аннотация. Разработана методика выбора кратчайшего маршрута в транспортной сети с заездом во все пункты. Предусмотрены варианты движения из начального пункта в наиболее удалённый или с возвращением в исходный пункт. Расчёты проведены в среде Excel с использованием сервиса "Поиск решения" (Solver). Задача решается в интерактивном режиме "человек-компьютер".

Ключевые слова: транспортная сеть, кратчайший маршрут, задача коммивояжёра.

CHOICE OF THE OPTIMUM WAY THROUGH A TRANSPORT NETWORK (THE PROBLEM OF A BUSINESS VOYAGER)

N.V.Katargin

Finacial University under the Government of Russian Federation, Moscow, Russia

Abstract. The technique of a choice of the shortest route in a transport network with arrival in all points is developed. Variants of movement are provided: from a starting point in the most remote; with returning in starting point. Calculations are spent in Excel using the Service Solver. The problem dares in an interactive mode "person-computer".

Keywords: transport network, shortest route, problem of business voyager.

Задача коммивояжера была поставлена в 1934 году. Постановка задачи: выбрать оптимальный маршрут поездки из одного города в другой с заездом в указанные города. Задача коммивояжера является одной из знаменитых задач теории комбинаторики и пользуется популярностью благодаря тому, что к ней сводится большое количество практических задач [1, 2]. Для решения задачи коммивояжера используют различные группы простейших методов: полный и случайный перебор, жадный и деревянный алгоритмы, метод имитации отжига. Широкое применение получили различные модификации более эффективных методов, таких как метод ветвей и границ, генетических алгоритмов, а также алгоритм муравьиных колоний [3, 4].

Решим задачу, используя Сервис «Поиск решения» (Solver) Excel, основанный на итерационной процедуре последовательного приближения к экстремуму целевой функции при соблюдении ограничений (метод Ньютона, сопряжённых градиентов или другие). Этот метод применяли для решения задач коммивояжёра малой размерности, так как количество варьируемых переменных растёт пропорционально квадрату количества узлов сети (городов), и возможностей Excel не хватает для решения задачи. В данной работе удалось решить эту проблему путём создания так называемого "короткого" плана перевозок.

Транспортная сеть состоит из n узлов (будем называть их также пунктами или городами), некоторые из которых соединены магистралями. Стоимость проезда по каждой из таких магистралей известна и отмечена на схеме (рис. 1). Надо найти оптимальный маршрут проезда из 0-го пункта в n -ый с заездом в указанные пункты. В данном примере целевая функция – суммарная стоимость проезда, а область допустимых решений является дискретным множеством – набором единиц и нулей, означающих проезд из одного города в другой или отказ от проезда.

Пусть сеть состоит из 11 узлов, соединённых магистралями согласно схеме. Стоимость проезда (или расстояние) из пункта i в пункт k равна R_{ik} , и элементы этой матрицы приведены на схеме.

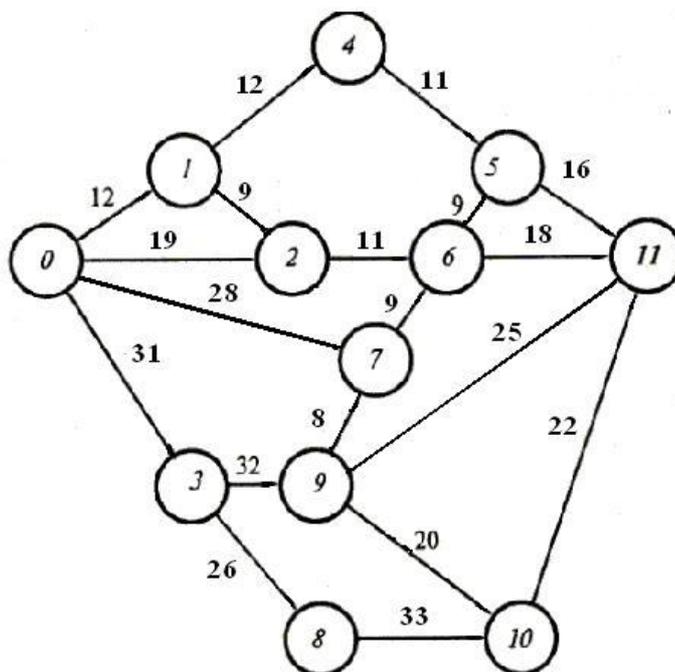


Рис.1. Транспортная сеть.

Найдём оптимальный маршрут из 0-го пункта в 11-ый. Для решения задачи надо построить таблицы: *Стоимость проезда (расстояния) R_{ik}* ; *План поездки X_{ik}* , который в данном случае представляет из себя матрицу из единиц и нулей, соответствующих перемещению или не перемещению из одного пункта в другой; таблицу произведений $R_{ik}X_{ik}$, то есть расходов на поездки (*Затраты*). Таблицы симметричны относительно диагонали, то есть возможны поездки в любом направлении; это даёт возможность произвольной нумерации пунктов, кроме первого и последнего. Решение задачи сводится к поиску комбинации X_{ik} , обеспечивающей проезд из пункта 0 в пункт 11 при минимальных затратах. Целевая функция в данной задаче – сумма по таблице *Затраты* $\sum X_{ik}R_{ik}$. Её надо минимизировать, изменяя ячейки таблицы *План поездки X_{ik}* , при ограничениях: все X_{ik} двоичные (0 или 1); если есть приезд в город, должен быть выезд; выезд из пункта 0 и прибытие в последний обязательны. Для программирования этих условий надо суммировать строки и столбцы таблицы *План поездки X_{ik}* ; ненулевое значение суммы по строке означает выезд из соответствующего пункта (из п.0 обязательно); ненулевое значение в сумме по столбцу означает приезд в соответствующий пункт (в п.11 обязательно). Приезд в какой-либо пункт, кроме п.11, требует обязательного выезда из него, т.е. соответствия сумм по столбцам суммам по строкам.

Количество ячеек в таблице X_{ik} достаточно велико, и компьютер может не справиться с решением задачи. Поэтому предлагается принципиально новый подход: создание дополнительной таблицы *План поездки в компактном виде*, из которой копируются значения в зависимые от нее ячейки таблицы *План поездки X_{ik}* . В ячейки таблицы *План поездки X_{ik}* вносятся формулы, связывающие ее с таблицей *План поездки в компактном виде*, в которую вносятся единицы, нули или опорный план на основе мнений экспертов.

Найдём оптимальный маршрут из п.0 в п.11 без дополнительных ограничений. В таблицах 2, 3, 4 представлены результаты работы «Поиска решения».

Таблица 1. Стоимость проезда (расстояния) R_{ik}
куда

откуда	0	1	2	3	4	5	6	7	8	9	10	11
0		12	19	31				28				
1	12		9		12							
2	19	9					11					
3	31								26	32		
4		12				11						
5					11		9					16
6			11			9		9				18
7	28						9			8		
8				26							33	
9				32				8			20	25
10									33	20		22
11						16	18			25	22	

Таблица 2. План поездки X_{ik} .

куда

		B	C	D	E	F	G	H	I	J	K	L	M	N
	откуда	0	1	2	3	4	5	6	7	8	9	10	11	выезд
29	0	0	0	1	0				0					1
30	1	0		0		0								0
31	2	0	0					1						1
32	3	0								0	0			0
33	4		0				0							0
34	5					0		0					0	0
35	6			0			0		0				1	1
36	7	0						0			0			0
37	8				0							0		0
38	9				0				0			0	0	0
39	10									0	0		0	0
40	11													
41	приезд		0	1	0	0	0	0	1	0	0	0	0	1

Таблица 3. План поездки в компактном виде.

	C	D	E	F	G	H	I	J	K	L	M	N
24	0	1	0	0	0	1	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	1	0
26	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0									

Таблица 4. Затраты $R_{ik} X_{ik}$. Нули и пустые строки удалены.

откуда	0	1	2	3	4	5	6	7	8	9	10	куда 11
0			19									
2							11					
6												18
												Целе- вая
											Σ	48

В окне «Поиска решения» следует задать: Целевую ячейку $\Sigma X_{ik}R_{ik}$, Изменяя ячейки: *План поездки в компактном виде*, Ограничения: *План поездки в компактном виде* двоичные; суммы по строкам таблицы *План поездки* X_{ik} (приезд), начиная со второй (с п.1) должны равняться суммам по столбцам (приезд), исключая последнее значение, п.11 (N31:N40=C42:L42). Суммы по первой строке (выезд из п.0, N30) и последнему столбцу (приезд в п.11, M42) не должны равняться 0. Запустите *Выполнить*.

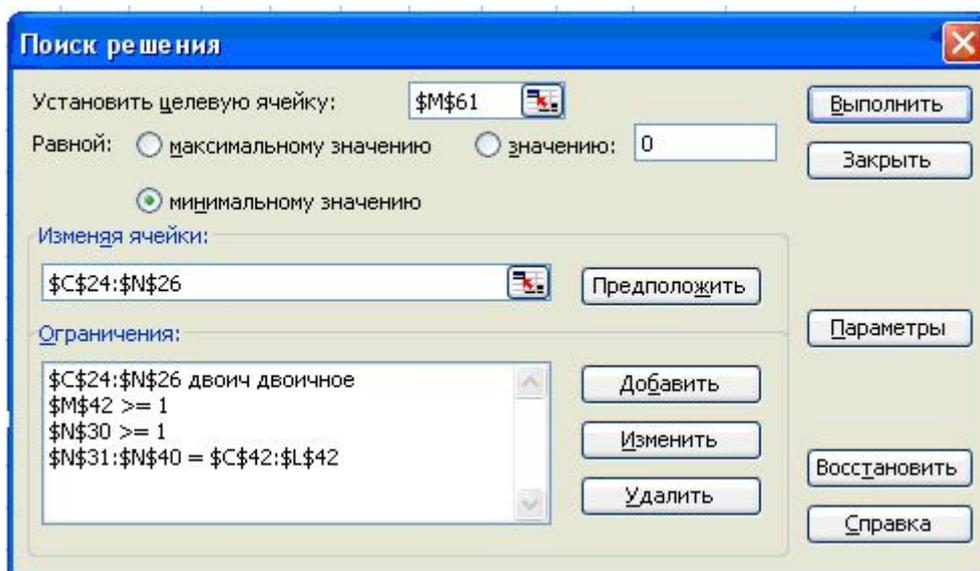


Рис.2. Окно «Поиска решения»

Оптимальный план поездки: пункты $0 \Rightarrow 2 \Rightarrow 6 \Rightarrow 11$, стоимость 48 у.е. (Эту технологию можно использовать для нахождения критического пути в сетевом графике комплекса работ, только целевую функцию надо максимизировать. Критический путь $0 \Rightarrow 3 \Rightarrow 8 \Rightarrow 10 \Rightarrow 11$, длина 112).

Приезд в город становится обязательным, если введено дополнительное ограничение: неравенство нулю суммы по соответствующему столбцу таблицы *План поездки*. Возможны несколько заездов, то есть сумма может быть больше 1. Если необходимо сделать обязательным приезд в один из группы городов — вычислите сумму по ячейкам приезда в эти города и введите ограничение: эта сумма больше или равна единице.

Наиболее интересная задача — спланировать маршрут, проходящий через все узлы сети. В общем виде эта «Задача коммивояжера» до сих пор не решена. Попробуем решить её с помощью компьютера. Для этого надо ввести дополнительное ограничение: все суммы по столбцам таблицы *План поездки* должны быть больше или равны 1. Полученный результат:

0=>1=>2=>6=>7=>9=>10=>11 и отдельно 3=>8=>3 и 4=>5=>4, стоимость 165 у.е. Компьютер выполнил все условия, но два «острова» 4-5 и 3-8 оказались не связанными с основным маршрутом. Решить задачу в общем виде не удаётся, но для практических целей эту технологию можно использовать. Для этого надо ввести дополнительные ограничения, вынуждающие компьютер войти на «острова» с основного маршрута.

Таблица 5. План поездки X_{ik} .

		B	C	D	E	F	G	H	I	J	K	L	M	N
	откуда	0	1	2	3	4	5	6	7	8	9	10	11	выезд
29	0	→	1	0	0				0					1
30	1		0	↘	1		0							1
31	2		0	-0				↗	1					1
32	3		0							↗	1	0		1
33	4			0				↘	1					1
34	5					↖	1		0				0	1
35	6				0			0		↘	1		0	1
36	7	0							0			↗	1	1
37	8				1								0	1
38	9				0				0				↗	1
39	10									0	0			↘
40	11													↗
41	приезд			1	1	1	1	1	1	1	1	1	1	

В данном примере на остров 3-8 можно попасть по маршрутам 0-3, 9-3 и 10-8, на остров 4-5 по маршрутам 1-4, 6-5 и 11-5. Исходя из конкретной ситуации, поставив ограничения: больше или равно единице в ячейках 9-3 ($E_{38} \geq 1$) и 1-4 ($F_{30} \geq 1$) получим маршрут 0=>1=>2=>1=>4=>5=>6=>7=> 9=>3=>8=>10=>11, стоимость 192 у.е. При этом пункт 1 посещается дважды. Этот маршрут не вполне оптимален. Если указать на маршрут 0=>2 (D29=1), то получится маршрут 0=>2=>1=>4=>5=>6=>7=>9=> 3=>8=>10=>11, стоимость которого 190 у.е.

Можно предоставить компьютеру большую свободу выбора маршрута на острова. Для этого надо суммировать содержимое ячеек таблицы План поездки X_{ik} , соответствующим входам на острова, и в «Поиске решения» потребовать для этих сумм ограничения ≥ 1 . Но, как показали эксперименты, при этом могут появиться новые острова.

При решении данной задачи проявилось свойство задач нелинейного программирования: компьютер может найти различные маршруты, соответствующие локальным минимумам целевой функции в зависимости от опорного плана.

Наиболее сложный вариант данной задачи — объехать все пункты сети с возвращением в исходный пункт. Маршрут приходится разбивать на две части: «вперёд» и обратно, иначе компьютер разбивает сеть на множество «островов». В данном случае, исходя из топологии сети, её можно разбить следующим образом: пункты 0, 1, 4, 5, 6, 7, 11, причём п.7 — конечный, так как он ближе к другому кластеру 3, 8, 9, 10, 0, и в нём п.7 является начальным. При дополнительных ограничениях на ячейки План поездки: запрет маршрута 0=>7 (иначе образуется остров) и Сумма ячеек 1-4 и 6-5 больше или равна 1, получается маршрут 0=>2=>1=>4=>5=>6=>11=>6=>7, стоимость 105 у.е. Обратный путь проще, компьютер решает задачу без дополнительных ограничений и выдаёт маршрут 7=>9=>10=>8=>3=>0, стоимость 118 у.е. Таким образом, общая стоимость поездки 223 у.е.

ВЫВОД: Итерационные градиентные методы, встроенные в сервис «Поиск решения» Excel, можно использовать для выбора кратчайшего маршрута через сеть, в том числе с заездом в

заданные пункты или во все. В общем виде задача с заездом во все пункты не решается, но решается путём взаимодействия человека и компьютера: человек задаёт разумный опорный план и ограничения, помогающие компьютеру решить задачу.

Список литературы

1. Экономико-математическое моделирование в управлении компанией: оптимизация транспортных потоков [Электронный ресурс] // URL: <http://www.cig-bc.ru/library/74190/78999/>.
2. http://ru.wikipedia.org/wiki/Задача_коммивояжера.
3. Задача – коммивояжер [Электронный ресурс] // URL: <http://www.ai08.org/index.php/term/..xhtml>.
4. Метод ветвей и границ [Электронный ресурс] // URL: <http://pco.iis.nsk.su/ICP/Practice/dd8-3/node9.html>.

Поступила: 04.03.13.