УДК 004.02

РЕШЕНИЕ ЗАДАЧИ МАРШРУТИЗАЦИИ В СРЕДЕ VBA

А.В. Иваницкая
Е.Н. Едемская, ст. преподаватель
Донецкий национальный технический университет
Svetivadon@yandex.ru

Разработана математическая постановка задачи маршрутизации в виде задачи коммивояжера. Для решения задачи по алгоритму локальной оптимизации составлена программа в среде VBA.

The mathematical raising of task of routing as a task of traveling salesman is developed. For the decision of task on the algorithm of peep-hole optimization the VBA program is proposed.

Розроблена математична постановка задачі маршрутизації у вигляді задачі комівояжера. Для вирішення задачі по алгоритму локальної оптимізації складена програма в середовищі VBA.

ЗАДАЧА МАРШРУТИЗАЦИИ, ЗАДАЧА КОММИВОЯЖЕРА, АЛГОРИТМ ЛОКАЛЬНОЙ ОПТИМИЗАЦИИ, ПРОГРАММА В СРЕДЕ VBA

Маршрутизация - процесс определения в сети пути, по которому пакет может достигнуть адресата. Задача маршрутизации может быть сведена к задаче коммивояжера [1]. В ней необходимо обойти фиксированное число городов, начиная с города, в котором он находится, и закончить свой маршрут, вернувшись в исходный город, не побывав нигде дважды. Известна стоимость пути между любой парой городов. При этом не очевидно, что наиболее короткий маршрут будет обладать минимальной стоимостью.

Требуется определить маршрут или последовательность посещения городов, которые обладают минимальной суммарной стоимостью пути среди всех возможных маршрутов.

Оценочной функцией в данной задаче является суммарная длина полного пути, начинающегося и оканчивающегося в некотором городе, ограничениями служат наличие или отсутствие рейса между отдельными городами рассматриваемого списка, а также необходимость посещения всех этих городов.

Задача коммивояжера представляет собой пример классической задачи комбинаторной оптимизации, которая выглядит очень простой по своей постановке, но требует самых серьезных усилий для нахождения точного решения. В бизнес-приложениях встречается также другой вариант этой задачи, известный как задача о переналадке оборудования или задача составления оптимального плана.

Задача коммивояжера может служить тестовой задачей для проверки вычислительных алгоритмов комбинаторной оптимизации. В данной работе рассматривается формулировка задачи коммивояжера в виде задачи

оптимизации на графах, используемая для формальной записи условий и решение соответствующей задачи оптимизации в виде модели булева программирования.

Задача коммивояжера может быть сформулирована в форме задачи оптимизации на графах. Рассмотрим связный ориентированный граф: $G = (V, E, h)$, в котором $V = \{v_1, v_2, \ldots, v_n\}$ — конечное множество вершин, $E = \{e_1, e_2, \ldots, e_m\}$ - конечное множество дуг, $h: E \to Z_+$ — весовая функция дуг. Для математической постановки задачи удобно обозначить отдельные значения весовой функции дуг через $c_{ij} = h(e_k)$, где дуга $e_k \in E$ соответствует упорядоченной паре вершин (v_i, v_j). Значение $c_{ij} = h(v_i, v_j)$ интерпретируется как длина участка (i, j) исходного графа.

Длина любого подмножества дуг $E_k \subset E$ в графе G равна сумме весов дуг, входящих в это подмножество. Требуется определить такое подмножество дуг, которое образует в графе G замкнутый путь, проходит через каждую вершину ровно один раз и обладает минимальной длиной.

Для формальной записи условий задачи коммивояжера в виде модели булева программирования следует отметить следующие особенности искомого маршрута в графе:

1. Каждая из вершин исходного графа должна иметь в искомом маршруте ровно одну инцидентную ей дугу, которая является входящей для этой вершины, и ровно одну инцидентную ей дугу, которая является выходящей для этой вершины. В противном случае такие вершины окажутся изолированными или тупиковыми и, следовательно, путь не будет проходить через все вершины исходного графа.
2. Общее количество дуг в искомом пути должно быть в точности равно n, где n - общее количество вершин исходного графа. Действительно, если некоторый путь содержит меньше n дуг, то он не будет проходить через все вершины или являться циклическим. Если же искомый путь содержит больше n дуг, то он не будет удовлетворять условию прохождения каждой вершины ровно один раз.
3. Искомый путь должен представлять собой единственный цикл и не должен распадаться на отдельные циклы с количеством дуг меньше n, где n - общее количество вершин исходного графа.

Введем булевы переменные x_{ij}, которые интерпретируются следующим образом. Переменная $x_{ij} = 1$, если дуга (v_i, v_j) входит в искомый маршрут минимальной длины, т.е. коммивояжер непосредственно переезжает из i-го города в j-й город, и $x_{ij} = 0$, если дуга (v_i, v_j) не входит в оптимальный маршрут, т.е. коммивояжер не переезжает из i-го города в j-й город.

Математическая постановка задачи коммивояжера может быть сформулирована следующим образом:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min_{x\in\Delta_\beta}, \qquad (1)$$

Множество допустимых альтернатив Δ_β формируется следующей системой ограничений:

$$\begin{cases} \sum_{j=1}^{r} x_{ij} = 1, (\forall i \in \{1,2,...,n\}); & (2) \\ \sum_{i=1}^{n} x_{ij} = 1, (\forall j \in \{1,2,...,n\}); & (3) \\ u_i - u_j + n \cdot x_{ij} \le n-1, (\forall i, j \in \{2,3,...,n\}, i \ne j); & (4) \\ x_{ij} \in \{0,1\}, (\forall i, j \in \{1,2,...,n\}); & (5) \\ u_i \in R^1, (\forall i \in \{2,3,...,n\}). & (6) \end{cases}$$

В данной математической модели используются вспомогательные переменные u_i $(\forall i \in \{2, 3,..., n\})$, которые могут принимать любые действительные значения. При этом ограничения (2) и (3) обеспечивают выполнение первых двух указанных ранее условий - искомый путь должен проходить через каждую вершину графа ровно один раз. Ограничения (4) обеспечивают выполнение третьего из указанных ранее условий - искомый путь не должен распадаться на отдельные циклы. Ограничения (5) обеспечивают выполнение условия: переменные x_{ij} должны принимать булевы значения, а ограничения (6) обеспечивают выполнение условия: переменные u_i, должны принимать вещественные значения.

Следует заметить, что те коэффициенты целевой функции c_{ij}, для которых весовая функция ребер h исходного графа не определена или равна 0, в математической постановке рассматриваемой задачи (1)-(6) нужно положить равными достаточно большому положительному значению.

Решение данной задачи с помощью программы MS Excel очень трудоемко. Предварительно следует определить $n(n-1)$ основных переменных и $n-1$ вспомогательных переменных, а также сформировать и ввести в ячейки рабочего листа n^2-n+2 ограничений вида (2)-(4), где n - число вершин исходного графа. Данный процесс для значений $n > 10$ представляется рутинной процедурой. По этой причине возникает необходимость разработать специальную программу, которая на основе заданной матрицы весов дуг произвольного ориентированного графа находит полный замкнутый путь минимальной длины.

В работе рассматриваются алгоритм и программа приближенного решения задачи коммивояжера, которые могут быть использованы для решения задач с большим числом городов.

Для решения используется метод локальной оптимизации [2], ориентированный на решение задачи коммивояжера в комбинаторной постановке. Алгоритм имеет следующую последовательность действий:

1. *Предварительное задание начального решения* x_0. До начала выполнения основных итераций алгоритма задается некоторая допустимая перестановка $(1, i_2, i_3, ..., i_{n-1}, i_n, 1)$ из элементов числового множества $\{1, 2, ..., n\}$ которой соответствует допустимое решение $x_0 \in \Delta_\beta$ и значение целевой функции $f(x_0)$. После чего следует перейти к выполнению действий шага 2.

2. *Нахождение локально-оптимального решения.* Для решения x_0 рассматриваются все допустимые решения в локальной окрестности $\delta_2(x_0)$ и среди них выбирается решение с минимальным значением целевой функции: $x_j = \textbf{\textit{argmin}} \{f(x_i)\}$ для всех $x_i \in \delta_2(x_0)$. После чего следует перейти к выполнению действий шага 3.
3. *Нахождение нового начального решения.* Если выполняется условие $f(x_0) \leq f(x_j)$, то в локальной окрестности $\delta_2(x_0)$ отсутствуют допустимые решения со значением целевой функции меньшим, чем значение целевой функции начального решения. На этом выполнение алгоритма заканчивается, и в качестве приближенного решения задачи принимается решение x_0. В противном случае допустимое решение x_j следует принять за новое начальное решение x_0 и перейти к выполнению действий шага 3.

Для решения задачи коммивояжера целесообразно реализовать алгоритм локальной оптимизации в виде пользовательской функции, позволяющей находить минимальный полный замкнутый путь в ориентированном графе для произвольного количества вершин и дуг.

Алгоритм локальной оптимизации для поиска полного замкнутого пути:

```
'--- Подготовка исходных массивов
For I = 1 To N
  For J = 1 To N
    Dis(I, J) = C(I, J)
  Next J
Next I
'--- Задание начального решения - рекорда
For I = 1 To N - 1
  VRec(I) = I + 1
Next I
VRec(N) = 1
Rec = SumPath(VRec, Dis, N)
LocOptRec = Rec
L = 0
For I = 1 To N
  VLocOpt(I) = VRec(I)
Next I
While Rec <= LocOptRec And L = 0
  LocOptRec = Rec
  L = 1
'--- Генерация новой перестановки - решения
  For I = 1 To N - 2
    For J = I + 1 To N - 1
      For K = 1 To N
        VLocVar(K) = VRec(K)
      Next K
      Im = VLocVar(I)
```

```
        VLocVar(I) = VLocVar(J)
        VLocVar(J) = Im
        FunLocVar = SumPath(VLocVar, Dis, N)
        If FunLocVar < LocOptRec And FunLocVar <> 0 Then
'--- новое решение лучшее в локальной окрестности
          LocOptRec = FunLocVar
          For K = 1 To N
            VLocOpt(K) = VLocVar(K)
          Next K
        End If
      Next J
    Next I
    If LocOptRec < Rec And LocOptRec <> 0 Then
'--- лучшее в локальной окрестности решение становится новым рекордом
      Rec = LocOptRec
      L = 0
      For K = 1 To N
        VRec(K) = VLocOpt(K)
      Next K
    End If
Wend
'--- Подготовка строки с результатом решения задачи
S = " (" & CStr(1) & "," & CStr(VRec(1)) & ")"
For I = 1 To N - 1
  S = S + " (" & CStr(VRec(I)) & "," & CStr(VRec(I + 1)) & "),"
Next I
S1 = "Минимальный полный замкнутый путь   "
S = S1 + S + " Fmin= " + CStr(Rec)
LocOptKom = S
```

Функция расчета длины полного замкнутого пути в графе:

```
Function SumPath(V, D, N As Variant) As Integer
Dim I, L1, Fun As Integer
Fun = 0
L1 = 1
If D(1, V(1)) <> 0 Then
  Fun = D(1, V(1))
 Else
  L1 = 0
End If
For I = 1 To N - 1
  If D(V(I), V(I + 1)) <> 0 Then
'--- для пары вершин расстояние не равно 0
    Fun = Fun + D(V(I), V(I + 1))
   Else
```

```
'--- для пары вершин расстояние равно 0
      L1 = 0
    End If
    If L1 = 1 Then
'--- если для пути нет пары вершин с расстоянием 0
      SumPath = Fun
     Else
'--- если для пути есть пара вершин с расстоянием 0
      SumPath = 0
    End If
Next I
End Function
```

Результат решения тестовой задачи показан на рисунке 1.

A10 =LocOptKom(B3:G8)

	A	B	C	D	E	F	G	H
1	**Матрица весов дуг исходного графа**							
2		v1	v2	v3	v4	v5	v6	
3	v1	0	4	10	13	4	8	
4	v2	2	0	9	7	6	7	
5	v3	8	5	0	5	5	9	
6	v4	5	8	5	0	7	10	
7	v5	6	4	4	9	0	4	
8	v6	5	1	4	8	3	0	
9								
10	Минимальный полный замкнутый путь	(1,5) (5,3), (3,4), (4,6), (6,2), (2,1), Fmin= 26						
11								

Рис. 1.

В работе разработана математическая постановка задачи маршрутизации в виде задачи коммивояжера и для решения задачи по алгоритму локальной оптимизации составлена программа в среде VBA.

Литература

1. Олифер В.Г, Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Санкт-Петербург: Питер, 2006. – 958 с.
2. Леоненков А.В. Решение задач оптимизации в среде MS Excel. СПб: БХВ-Петербург, 2005. – 704 с.