

## РЕШЕНИЕ ЗАДАЧИ ПРОЕКТИРОВАНИЯ РАДИОЭЛЕКТРОННОЙ АППАРАТУРЫ В СРЕДЕ VBA

А.В. Рагулина

Е.Н. Едемская, ст. преподаватель

Донецкий национальный технический университет

[roxy\\_r@mail.ru](mailto:roxy_r@mail.ru)

*Разработана математическая постановка задачи проектирования радиоэлектронной аппаратуры в виде задачи о разбиении конечного множества объектов на фиксированное число классов разбиения так, чтобы обеспечивалось оптимальное значение критерия оптимизации. Для решения задачи по алгоритму локальной оптимизации составлена программа в среде VBA.*

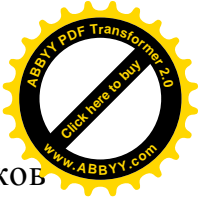
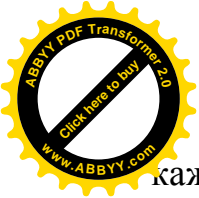
*The mathematical raising of task of planning a radio electronic apparatus as a task about breaking up a finite set of objects on the fixed number of classes of breaking up is developed so that an optimum value of optimization criterion was secured. For the decision of task on the algorithm of peep-hole optimization the VBA program is proposed.*

*Розроблена математична постановка задачі проектування радіоелектронної апаратури у вигляді задачі про розбиття кінцевої множини об'єктів на фіксоване число класів розбиття так, щоб забезпечувалося оптимальне значення критерію оптимізації. Для вирішення задачі по алгоритму локальної оптимізації складена програма в середовищі VBA.*

**ЗАДАЧА О РАЗБИЕНИИ, АЛГОРИТМ ЛОКАЛЬНОЙ ОПТИМИЗАЦИИ, ПРОГРАММА В СРЕДЕ VBA**

Сущность задачи проектирования радиоэлектронной аппаратуры заключается в следующем. Научно-производственное предприятие проектирует радиоэлектронное устройство, которое согласно проектной документации должно состоять из отдельных аппаратурных модулей. Для их изготовления используются радиоэлектронные компоненты. Известен общий состав этих компонентов, необходимых для изготовления одного устройства, и электрическая схема их соединения. При этом каждый компонент обладает некоторой мощностью потребления энергии, а отдельные компоненты должны быть соединены между собой электрическими шинами, количество проводников в которых определяется электрической схемой их соединения. Задана предельная величина мощности, которую при размещении компонентов в одном модуле по техническим условиям нельзя превышать.

Задача состоит в том, чтобы все имеющиеся компоненты разбить на отдельные модули, так чтобы общая мощность компонентов в каждом модуле не превышала заданной предельной величины, и при этом размещаемые в



каждом из модулей компоненты имели максимальное количество проводников электрических шин для связи между собой. Таким образом, задача проектирования радиоэлектронной аппаратуры сводится к задаче о разбиении конечного множества объектов на фиксированное число классов разбиения так, чтобы обеспечивалось оптимальное значение критерия оптимизации.

Исходная постановка задачи о разбиении может быть сформулирована в форме задачи оптимизации на графах. Рассмотрим связный неориентированный граф:  $G = (V, E, d, h)$ , в котором:  $V = \{v_1, v_2, \dots, v_n\}$  — конечное множество вершин,  $E = \{e_1, e_2, \dots, e_m\}$  — конечное множество ребер,  $d: V \rightarrow Z_+$  — весовая функция вершин,  $h: E \rightarrow Z_+$  — весовая функция ребер. Дополнительно задано некоторое положительное число  $w$ , интерпретируемое как максимально допустимый вес вершин в каждом из подмножеств разбиения, а также натуральное  $r$  — максимальное количество подмножеств разбиения.

Требуется определить такое разбиение вершин графа на подмножества, при котором обеспечивается максимум суммы весов ребер для всех вершин, входящих в каждое из подмножеств разбиения, и удовлетворяются следующие условия:

1. Каждая из вершин исходного графа может принадлежать только одному подмножеству разбиения. В противном случае решение не будет удовлетворять определению разбиения.
2. Общая сумма весов вершин в каждом из подмножеств искомого разбиения не должна превышать заданного максимально допустимого веса вершин подмножества  $w$ .

Для формулировки задачи о разбиении в виде модели булева программирования введем следующие переменные:  $x_{ik} = 1$ , если вершина  $v_i$  входит в  $k$ -е подмножество разбиения, и  $x_{ik} = 0$  в противном случае, т. е. если вершина  $v_i$  не входит в  $k$ -е подмножество разбиения.

Математическая постановка задачи о разбиении имеет вид:

$$\frac{1}{2} \sum_{k=1}^r \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ik} x_{jk} \rightarrow \max_{x \in \Delta_\beta}, \quad (1)$$

Множество допустимых альтернатив  $\Delta_\beta$  формируется следующей системой ограничений типа равенств и неравенств:

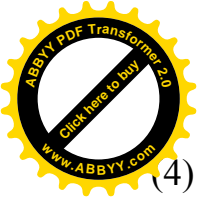
$$\left\{ \begin{array}{l} \sum_{k=1}^r x_{ik} = 1, (\forall i \in \{1, 2, \dots, n\}); \end{array} \right. \quad (2)$$

$$\left\{ \begin{array}{l} \sum_{i=1}^n x_{ik} \leq w, (\forall k \in \{1, 2, \dots, r\}); \end{array} \right. \quad (3)$$

$$\left\{ \begin{array}{l} x_{ik} \in \{0, 1\}, (\forall i \in \{1, 2, \dots, n\}, \forall k \in \{1, 2, \dots, r\}). \end{array} \right. \quad (4)$$

Здесь  $c_{ij} = h(v_i, v_j)$  — вес ребра  $\{v_i, v_j\}$ , которому соответствует  $e_i \in E$ ;  $d(v_i)$  — вес вершины  $v_i \in V$ ;  $r$  — общее количество подмножеств разбиения.

В данной математической модели ограничения (2) обеспечивают выполнение первого из двух указанных ранее условий. Ограничения (3) обеспечивают выполнение второго из указанных ранее условий. Ограничения



(4) обеспечивают выполнение общего условия: переменные  $x_{ij}$  должны принимать булевы значения. Общее количество ограничений (2) и (3) равно:  $n + r$ .

Задача о разбиении может быть также сформулирована как задача комбинаторной оптимизации. Каждому допустимому решению задачи о разбиении соответствует некоторое разбиение  $R = \{A_1, A_2, \dots, A_r\}$ , где каждый класс разбиения  $A_i$  ( $\forall i \in \{1, 2, \dots, r\}$ ) состоит из вершин исходного графа  $G = (V, E, d, h)$  или, что эквивалентно, из элементов числового множества  $A = \{1, 2, \dots, n\}$ .

Математическая постановка задачи о разбиении в форме задачи комбинаторной оптимизации может быть сформулирована в следующем виде:

$$f(R) \rightarrow \max_{R \in \Delta_B}, \quad (5)$$

Множество допустимых альтернатив  $\Delta_B$  содержит все возможные разбиения множества вершин исходного графа или числового множества  $A = \{1, 2, \dots, n\}$  на  $r$  классов, которые удовлетворяют следующим ограничениям:

$$d(v_{i1}) + d(v_{i2}) + \dots + d(v_{iq}) \leq w \quad (\forall A_i \in R) \quad (6)$$

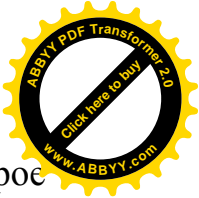
Каждый класс  $A_i$  ( $\forall i \in \{1, 2, \dots, r\}$ ), входящий в допустимое разбиение  $R$ , состоит из вершин  $A_i = \{v_{i1}, v_{i2}, \dots, v_{iq}\}$  или из элементов  $A_i = \{i_1, i_2, \dots, i_q\}$ . При этом для каждого разбиения предполагаются выполненными общие условия непересекаемости классов и принадлежности каждого элемента исходного множества одному из классов допустимого разбиения. Значения целевой функции  $f(R)$  определяются как общая сумма величин связности каждой пары элементов, попадающих в один класс допустимого разбиения, аналогично выражению (1).

Задача булева программирования может быть решена в среде MS Excel. Однако, предварительно следует определить  $n^2$  основных переменных, а также сформировать и ввести в ячейки рабочего листа  $2m$  ограничений вида (2) и (3) и  $m$  произведений переменных для вычисления целевой функции, где  $n$  - число вершин исходного графа,  $m$  - число ребер или дуг исходного графа. Данный процесс для больших значений является рутинной процедурой. Поэтому необходимо разработать специальную программу, которая на основе заданной матрицы весов дуг произвольного ориентированного графа находит разбиение вершин с максимальным значением целевой функции.

В работе рассматриваются алгоритм и программа приближенного решения задачи о разбиении, которые могут быть использованы при проектировании радиоэлектронной аппаратуры с большим количеством компонентов и модулей.

Для приближенного решения используется метод локальной оптимизации [1], ориентированный на решение задачи о разбиении в комбинаторной постановке. Он основывается на рекуррентном характере процедуры построения последовательности локальных окрестностей для допустимых разбиений. Алгоритм приближенного решения задачи имеет итеративный характер и заключается в выполнении следующих действий:

1. *Предварительное задание начального разбиения  $x_0$ .* До начала



выполнения основных итераций алгоритма задается некоторое допустимое разбиение, которому соответствует вектор  $(i_1, i_2, i_3, \dots, i_{n-1}, i_n)$  из элементов числового множества  $\{1, 2, \dots, n\}$ . Этому разбиению соответствует допустимое решение  $x_0 \in \Delta_B$  и значение целевой функции  $f(x_0)$ . После чего следует перейти к выполнению действий шага 2.

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

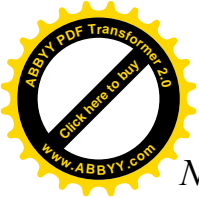
Для решения задачи алгоритм локальной оптимизации реализован в виде программы в среде VBA. Она позволяет находить допустимое разбиение вершин графа для произвольно заданного количества вершин и дуг.

Алгоритм приближенного решения задачи о разбиении:

```
For I = 1 To N '--- Задание начального разбиения - рекорда
  RazRec(I) = I
Next I
Rec = FuncRaz(RazRec, Sim, N)
M = 0
For I = 1 To N
  RazLocOpt(I) = RazRec(I)
Next I
RecLocOpt = Rec
While Rec >= RecLocOpt And M = 0
  RecLocOpt = Rec
  M = 1 '--- Генерация нового разбиения в локальной окрестности
  For K = 1 To N - 1
    For J = K + 1 To N
      For L = 1 To N
        RazLocVar(L) = RazLocOpt(L)
      Next L
      RazLocVar(J) = K
    For L = 1 To N
      If RazLocVar(L) <> L And _
        RazLocVar(RazLocVar(L)) <> RazLocVar(L) Then
        RazLocVar(L) = L
    Next L
  Next J
Next K
End While
```



```
End If
Next L
FuncLocVar = FuncRaz(RazLocVar, Sim, N)
If ProvRaz(RazLocVar, Wt, Wmax, N) = True And FuncLocVar > RecLocOpt Then
'--- Новое разбиение лучшее в локальной окрестности
  RecLocOpt = FuncLocVar
  For I = 1 To N
    RazLocOpt(I) = RazLocVar(I)
  Next I
End If
Next J
Next K
If RecLocOpt > Rec Then
'--- Лучшее в локальной окрестности разбиение становится новым рекордом
  M = 0 : Rec = RecLocOpt
  For I = 1 To N
    RazRec(I) = RazLocOpt(I)
  Next I
End If
Wend
'--- Подготовка строки с результатом решения задачи
SI = "Локально-оптимальное разбиение: " : S = ""
For I = 1 To N
  M = 0
  For J = 1 To N
    If RazRec(J) = I And M = 1 Then '---в классе разбиения уже есть элементы
      S = S + CStr(J) + ", "
    End If
    If RazRec(J) = I And M = 0 Then '---в классе разбиения еще нет элементов
      S = S + "{" + CStr(J) + ", "
      M = 1
    End If
    If J = N And M = 1 Then '---элементы в классе разбиения закончились
      S = Mid(S, 1, Len(S) - 2) + "},"
    End If
  Next J
Next I : S = SI + S + " Fmin= " + CStr(Rec)
  Функция расчета целевой функции (сумма весов дуг для вершин в одном
классе разбиения):
Function FuncRaz(Raz, Sim, N As Variant) As Integer
V = 0
For I = 1 To N - 1
  Im = Raz(I)
  For J = I + 1 To N
    If Raz(J) = Im Then V = V + Sim(I, J)
```



```

Next J
Next I
FuncRaz = V
End Function

```

Функция проверки допустимости разбиения:

```

Function ProvRaz(Raz, Wt, Wmax, N As Variant) As Boolean
Bol = True
For K = 1 To N
Wvar = 0
For I = 1 To N
If Raz(I) = K Then
Wvar = Wvar + Wt(I)
End If
Next I
If Wvar > Wmax Then
Bol = False
End If
Next K
ProvRaz = Bol
End Function

```

Результат решения тестовой задачи показан на рисунке 1.

A13      fx =LocOptRaz(B3:I10;J3:J10;J11)										
A	B	C	D	E	F	G	H	I	J	K
1	<b>Матрица весов дуг исходного гафа</b>									
2	v1	v2	v3	v4	v5	v6	v7	v8		Веса вершин
3	v1	0	9	0	7	0	1	0	0	4
4	v2	0	0	4	0	0	0	0	0	2
5	v3	0	0	0	5	0	0	0	1	3
6	v4	0	0	0	0	3	0	0	0	2
7	v5	0	0	0	0	0	2	0	9	3
8	v6	0	0	0	0	0	0	6	0	1
9	v7	0	0	0	0	0	0	0	2	3
10	v8	0	0	0	0	0	0	0	0	5
11	Максимальный вес вершин в одном классе разбиения									12
12										
13	Локально-оптимальное разбиение: {1, 2, 3, 4}, {5, 6, 7, 8}, Fmin= 44									
14										

Рис. 1.

В работе разработана математическая постановка задачи проектирования радиоэлектронной аппаратуры в виде задачи о разбиении конечного множества объектов на фиксированное число классов разбиения так, чтобы обеспечивалось оптимальное значение критерия оптимизации. Для решения задачи по алгоритму локальной оптимизации составлена программа в среде VBA.

### Литература

1. Леоненков А.В Решение задач оптимизации в среде MS Excel. СПб: БХВ-Петербург, 2005. – 704 с.