

## АЛГОРИТМИЗАЦИЯ ВЫБОРА КОНСТРУКТИВНО-СИЛОВЫХ СХЕМ ТУРБОМАШИН

### ALGORITHMIZATION OF SELECTION CONSTRUCTIONAL AND POWER SCHEES OF TURBOMACHINE

**БЕЛОУСОВ Анатолий Иванович**, д.т.н., профессор,  
**МЕДНИКОВ Максим Викторович**, аспирант,  
Самарский государственный аэрокосмический университет  
им. С.П. Королева, г. Самара, Россия.

*Abstract. The basis of automatic selection constructional and power schemes of tubomachines is the algorithm. Some approaches of algorithmization task of selection schemes are rating in this work.*

Требования к показателям технического совершенства и надежности ДЛА постоянно растут. Это приводит к повышению трудоемкости процесса конструирования, увеличению сроков и стоимости разработки. Поэтому требуется автоматизация проектно-конструкторских работ.

Задача выбора конструктивно-силовой схемы представляет собой задачу синтеза: выбора таких типов и количества структурных элементов, а также построения связей между ними, которые в целом отвечали бы тактико-техническим требованиям.

Для алгоритмизации этой задачи используются формализованные методы решения. Но, к сожалению, не все проектные процедуры могут быть формализованы (особенно процедуры принятия решения). Поэтому целесообразно применять диалоговый режим работы конструктора в САПР. В таком режиме конструктор имеет возможность вводить и изменять исходные данные, оценивать получаемые варианты решений и выбирать наиболее предпочтительный, с его точки зрения, вариант.

Подходы к алгоритмизации выбора конструктивно-силовой схемы турбомашин могут быть различными. Формально задачу синтеза конструктивно-силовой схемы можно рассматривать как процесс получения некоторого частного решения по значениям параметров (факторов), определяющих выбор этого решения. Решениями могут быть программы вычерчивания выбранных конструктивно-силовых схем на принтере или ото-

бражения их на дисплее ЭВМ и т.п. Параметрами являются типовые структурные элементы и связи между ними. Основой автоматизированного решения такой задачи является алгоритм, относящийся к классу алгоритмов выбора решений. Алгоритм выбора решений предусматривает целенаправленную проверку выполнимости всех или некоторых условий из заданного множества их и выбор соответствующего решения, обладающего нужными свойствами, из получаемого множества решений.

Любой алгоритм, в том числе алгоритм проектирования (и конструирования), в силу свойств дискретности и результативности может рассматриваться как система.

Действительно, элементами алгоритма проектирования как системы является множество преобразований (элементарных работ) и множество связей между ними.

Задача описания структуры алгоритма заключается в описании отношений между элементами множества преобразований и множества связей.

Удобным средством описания структуры конструкции и алгоритма проектирования конструкции являются *графы* [1]. Граф представляет собой геометрическое изображение, в котором двум элементам некоторого множества  $U$  ставится в соответствие один элемент некоторого множества  $V$ . Такое геометрическое изображение образует геометрическую конфигурацию, состоящую из нескольких точек и некоторого количества соединяющих эти точки отрезков. При этом точки называются вершинами графа, а отрезки - ребрами графа.

Различают ориентированные и неориентированные графы. Если каждое ребро графа ориентировано, т.е. имеет начало и конец (является вектором, а не отрезком), то граф называется ориентированным. Если начало каждого ребра графа может быть с равным основанием помещено в любую из вершин, являющихся концами ребер, то граф называется неориентированным. Характерной особенностью ориентированных графов являются так называемые бинарные отношения, т.е. отношения связи между парами элементов множества  $U$ , в которых определено, какой из элементов является предшествующим (первым), а какой последующим (вторым).

Использование графов для описания структуры алгоритмов проектирования и объектов проектирования повышает наглядность описания. Вместе с тем, являясь средством формализованного описания, графы позволяют разрабатывать алгоритмы, наиболее рациональные с точки зрения объема вычислений.

То, что графы являются средством формализованного описания, вытекает из возможности их представления в виде таблиц или матриц. Любой вершинный граф может быть представлен в виде таблицы или матрицы, называемой матрицей смежности вершин. Матрица смежности вершин представляет собой квадратную матрицу, строки и столбцы которой являются номерами элементов множества  $U$ , а клетки - элементами множества

отношений связи  $V$ .

Для указания наличия связи между двумя элементами в клетке матрицы с соответствующими номерами ставится 1; если элементы независимы, то в клетке ставится 0 или она остается пустой. Для указаний бинарных отношений принято предшествующим считать элемент с номером строки.

При другом возможном подходе к алгоритмизации выбора конструктивно-силовой схемы турбомашин при автоматизированном проектировании, развитым В.И. Баулиным [1] и в [2], предлагается использование *таблиц применяемости*. Эти таблицы оказались эффективным средством описания принятия решений не только при выборе конструктивно-силовых схем турбомашин, но и в задачах медицинской и технической диагностики.

Рассмотрим кратко структуру таблицы применяемости. Пусть  $X_1, X_2, \dots, X_n$  - некоторые параметры. Множество дискретных значений, принимаемых  $i$ -м параметром

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{ii}\}.$$

Параметром может быть, например, характеристика компонентов топлива - высококипящие или криогенные компоненты.

Обозначим множество решений

$$Y = \{Y_1, Y_2, \dots, Y_m\},$$

которые также могут принимать дискретные значения

$$Y_i = \{y_{i1}, y_{i2}, \dots, y_{ik}\}.$$

Таблица применяемости строится следующим образом (табл. 1). Столбцам присваиваются значения параметров, сгруппированных по их наименованиям. В каждой строке справа записывается некоторое решение из множества решений  $Y$ , соответствующее множеству значений параметров в строке. Клетки таблицы заполняются единицами и нулями. Наличие в клетке единицы означает, что параметр принимает соответствующее значение. Нули, как правило, не пишутся в таблице, и соответствующие клетки остаются пустыми. Таким образом, в каждой строке таблицы описано однозначное соответствие между множеством значений параметров и решением.

Особенностью табличного описания алгоритма является то, что оно позволяет легко составить соответствующую машинную программу. Действительно, каждая строка табл. 1 может быть записана следующим образом:

$$Y_1 = x_{11} \cdot x_{21} \cdot x_{31} \cdot x_{41} \cdot x_{52} \cdot x_{61},$$

$$Y_2 = x_{11} \cdot x_{22} \cdot x_{31} \cdot x_{42} \cdot x_{52} \cdot x_{62},$$

.....

$$Y_m = x_{12} \cdot x_{22} \cdot x_{32} \cdot x_{41} \cdot x_{52} \cdot x_{63}.$$

Таблица 1 - Таблица применяемости

$X_1$		$X_2$		$X_3$					$X_4$		$X_5$		$X_6$			$Y$
$x_{11}$	$x_{12}$	$x_{21}$	$x_{22}$	$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$	$x_{35}$	$x_{41}$	$x_{42}$	$x_{51}$	$x_{52}$	$x_{61}$	$x_{62}$	$x_{63}$	
1		1		1					1			1	1			$Y_1$
1			1	1						1		1		1		$Y_2$
1			1			1				1	1				1	$Y_3$
1		1			1					1	1				1	$Y_4$
.....																
	1	1					1			1	1				1	$Y_m$

Если произведение значений параметров  $i$ -й строки равно 1, то принимается соответствующее решение  $Y_i$ ; в противном случае (если произведение равно 0) данное решение не принимается.

Конструктивные схемы ТНА характеризуются большим числом параметров, их значений и типов схем, отличающихся по какому-либо классификационному признаку [2]. Поэтому может оказаться целесообразным последовательный метод решения. Он предусматривает построение алгоритма решения последовательным выбором частичных решений, которые становятся одним из условий выбора последующего (или последующих) частичного решения. Например, при синтезе конструктивной схемы ТНА такими частичными решениями могут быть:

- тип турбины (осевая или центробежная, окислительная или восстановительная, автономная или предкамерная и т.д.);
- взаимное расположение насосов и турбин(ы) (турбина между насосами, насос окислителя рядом с турбиной, насос горючего рядом с турбиной и т.п.);
- количество ступеней насоса (одно-, двух-, трёхступенчатый).

Перечень таких частичных решений можно было бы продолжить.

В тех случаях, когда решение представляет собой подмножество частичных решений, удобно использовать так называемый последовательный метод выбора решений. При последовательном методе решения алгоритм выбора решений строится таким образом, что частичные решения выбираются последовательно, одно за другим, причем ранее определенное частичное решение (или решения) может быть включено в множество условий выбора последующего (или последующих) частичного решения. Например, частичное решение "взаимное расположение насосов и турбин(ы)" может быть получено только после того, как выбраны час-

тичные решения, такие как “тип турбины”, “тип насоса”, “количество ступеней насоса”.

В этом случае алгоритм выбора решений  $Y$  описывается не одной, а несколькими таблицами применимости, которые называются частичными таблицами. Каждая из частичных таблиц по своей структуре отличается от описанной выше лишь тем, что среди параметров  $X_1, X_2, \dots, X_n$  могут находиться частичные решения  $Y_i^{(j)}$ , определенные ранее. Разумеется, что разработка частичных таблиц требует не только анализа зависимости между элементами множества параметров и множества частичных решений, но и анализа иерархической структуры множества частичных решений.

Соответствие между значениями параметров (факторов) и частичными решениями описывается частичными таблицами. При последовательном методе выбора конструктивная схема формально описывается произведением частичных решений:

$$Y_j = y_{1n} \cdot y_{2k} \cdot y_{3s} \dots y_{lt},$$

где  $y_{1n}, \dots, y_{lt}$  – выбранные частичные решения. Первый индекс (1, 2, ...) означает номер классификационного признака, а второй индекс – номер типа схемы, отличающейся по рассматриваемому признаку.

Для определенных исходных данных решение получается в виде множества вариантов схем. Необходимо уже на этапе синтеза схемы оценить каждый из вариантов с точки зрения качества конструкции ТНА.

Использование таблиц применимости:

- позволяет упорядочить процесс разработки алгоритмов синтеза схем;
- устанавливает единообразный язык, достаточно удобный для человека и одновременно близкий к языку ЭВМ;
- облегчает составление соответствующих машинных программ.

Ещё одним подходом к алгоритмизации выбора конструктивно-силовой схемы турбомашин при автоматизированном проектировании может быть *генетический алгоритм* [3].

Генетический алгоритм разработан Джоном Голландом (John Holland) в 1975 году в Мичиганском университете. Генетический алгоритм был получен в процессе обобщения и имитации в искусственных системах таких свойств живой природы, как естественный отбор, приспособляемость к изменяющимся условиям среды, наследование потомками жизненно важных свойств от родителей и т.д.

Так как алгоритм в процессе поиска использует некоторую кодировку множества параметров вместо самих параметров, то он может эффективно применяться для решения задач дискретной оптимизации, определенных как на числовых множествах, так и на конечных множествах произвольной природы. Поскольку для работы алгоритма в качестве информации об оптимизируемой функции используются лишь её значения в рассматриваемых точках пространства поиска и не требуется вычислений ни

производных, ни каких-либо иных характеристик, то данный алгоритм применим к широкому классу функций, в частности, не имеющих аналитического описания. Использование набора начальных точек позволяет применять для их формирования различные способы, зависящие от специфики решаемой задачи, в том числе возможно задание такого набора непосредственно человеком.

Генетический алгоритм работает с представленными в конечном алфавите строками  $S$  конечной длины  $L$ , которые используются для кодировки исходного множества альтернатив  $W$ . Строки представляют собой упорядоченные наборы из  $L$  элементов:  $S = (s_1, s_2, \dots, s_L)$ , каждый из которых может быть задан в своём собственном алфавите  $V_i, i = \overline{1, L}$ , т.е.  $s_i \in V_i$ , где алфавит  $V_i$  является множеством из  $r_i$  символов:  $V_i = \{v_{ij}, j = \overline{1, r_i}\}$ . Для решения конкретной задачи требуется однозначно отобразить конечное множество альтернатив  $W$  на множество строк подходящей длины (очевидно, что длина строк зависит от алфавитов, используемых для их задания).

Для работы алгоритма необходимо на множестве строк  $U^m (V_1^H, V_2^H, \dots, V_m^H)$  задать неотрицательную функцию  $F(S)$ , определяющую показатель качества, «ценность» строки  $S \in U$ . Алгоритм производит поиск строки, для которой  $F(S)$  имеет максимум.

Генетический алгоритм за один шаг производит обработку некоторой популяции строк. Популяция  $G(t)$  на шаге  $t$  представляет собой конечный набор строк:

$$G(t) = (S'_1, S'_2, \dots, S'_N), S'_k \in U^m_{(V_1, V_2, \dots, V_m)}, k = \overline{1, N},$$

где  $N$  — размер популяции, причём строки в популяции могут повторяться.

Анализ работы алгоритма удобно производить, используя аппарат схем. Схемой в генетическом алгоритме называют описание некоторого подмножества строк. Схема  $H = (h_1, h_2, \dots, h_m)$  может рассматриваться как строка, алфавиты для элементов которой дополнены специальным символом «#»:

$$H \in U^m_{(V_1^H, V_2^H, \dots, V_m^H)}, V_i^H = V_i \cup \{ \# \}.$$

Если в некоторой позиции  $r$  схемы  $H$  присутствует символ «#», то такая позиция называется свободной, а сам символ «#» интерпретируется как произвольный символ из алфавита  $V_r$ . Позиция  $q$  схемы  $H$  называется фиксированной, если в этой позиции присутствует один из символов алфавита  $V_q$ . Схема  $H$ , в которой определены фиксированные и свободные позиции, описывает подмножество  $U_H \subseteq U^m_{(V_1, V_2, \dots, V_m)}$ , содержащее такие строки,

у которых элементы, соответствующие фиксированным позициям схемы, совпадают с символами схемы, а элементы, соответствующие свободным позициям схемы, являются произвольно заданными в соответствующих алфавитах:

$$U_H = \left\{ S \mid S \in U^m_{(V_1, V_2, \dots, V_m)} \wedge (\forall i (i \in I_{[1, m]} \wedge h_i \neq \# \rightarrow (s_i = h_i))) \right\},$$

где  $I_{[1, m]}$  — множество целых чисел отрезка  $[1, m]$ .

Например, для множества строк  $U^5_{(V_1, V_2, V_3, V_4, V_5)}$ , где  $V_i = \{0, 1\}$ ,  $k = \overline{1, 5}$ , схема  $H_1 = "1\#\#\#0"$  задаёт такое множество строк, у которых первым элементом является символ "1", пятым — "0", а остальные — либо "0", либо "1". Строки "10010", "11110" являются примерами строк, принадлежащих множеству  $U_{H_1}$ .

Часть популяции  $G(t) = (S_1^t, S_2^t, \dots, S_N^t)$ , строки которой удовлетворяют схеме  $H$ , обозначают  $G_H(t) = (S_1^{Ht}, S_2^{Ht}, \dots, S_{n(H,t)}^{Ht})$ , где  $n(H, t)$  — число строк схемы  $H$  в популяции  $G(t)$  и называют подпопуляцией, соответствующей схеме  $H$ .

Кодировка строк популяции (конструктивно-силовых схем ТНА) производится следующим образом. Каждая строка (конструктивно-силовая схема ТНА) состоит из  $L$  элементов:  $S = (s_1, s_2, \dots, s_L)$ , где элемент  $s_i$  соответствует классификационному признаку (тип турбины, количество ступеней насоса и т.д.). Для каждого классификационного признака используется свой алфавит  $V_i$ . Например,  $s_2$  — количество ступеней насоса, алфавит  $V_2 = \{1, 2, 3\}$  ( $s_2=1$ , если насос одноступенчатый;  $s_2=2$ , если насос двухступенчатый;  $s_2=3$ , если насос трёхступенчатый)

Суть генетического алгоритма заключается в следующем.

Пусть на шаге  $t$  имеется популяция  $G(t)$ , состоящая из  $N$  строк. Для популяции вводится понятие средней ценности популяции  $F_{cp}(G(t))$ :

$$F_{cp}(G(t)) = \frac{1}{N} \sum_{k=1}^N F(S_k^t).$$

Аналогично для подпопуляции  $G_H(t)$ , удовлетворяющей схеме  $H$ , вводится понятие средней ценности подпопуляции  $F_{cp}(G_H(t))$ :

$$F_{cp}(G_H(t)) = \frac{1}{n(H,t)} \sum_{k=1}^{n(H,t)} F(S_k^{Ht}).$$

Генетический алгоритм осуществляет переход от популяции  $G(t)$  к популяции  $G(t+1)$  таким образом, чтобы средняя ценность составляющих её строк увеличивалась, причём количество новых строк в популяции равно  $K \cdot N$ , где  $K$  — коэффициент новизны. Если  $K < 1$ , то популяция будет перекрывающейся, т.е. в новой популяции сохраняются некоторые строки из старой, а если  $K = 1$ , то она будет неперекрывающейся, т.е. подвергнется полному обновлению.

Генетический алгоритм включает три операции: воспроизводство, скрещивание, мутация.

*Воспроизводство* представляет собой процесс выбора  $K \cdot N$  строк популяции  $G(t)$  для дальнейших генетических операций. Выбор производится

случайным образом, причём вероятность выбора строки  $S_i^t$  пропорциональна её ценности:

$$p_{\text{выб}}(S_i^t) = \frac{F(S_i^t)}{\sum_{k=1}^N F(S_k^t)}$$

Процесс выбора повторяется  $K \cdot N$  раз. Предполагаемое количество экземпляров строки  $S_i^t$  в популяции  $G(t+1)$  равно

$$n_{\text{выб}}(S_i^t) = p_{\text{выб}}(S_i^t) \cdot K \cdot N.$$

Операция воспроизводства увеличивает общую ценность последующей популяции путём увеличения числа наиболее ценных строк.

Воспроизводство оперирует со строками, уже присутствующими в рассматриваемой популяции, и само по себе не способно открывать новые области поиска. Для этой цели используется операция скрещивания.

*Скрещивание* представляет собой процесс случайного обмена значениями соответствующих элементов для произвольно сформированных пар строк. Для этого выбранные на этапе воспроизводства строки случайным образом группируются в пары. Далее каждая пара с заданной вероятностью  $p_{\text{скр}}$  подвергается скрещиванию. При скрещивании происходит случайный выбор позиции разделителя  $d$  ( $d=1, 2, \dots, L-1$ , где  $L$  — длина строки). Затем значения первых  $d$  элементов первой строки записываются в соответствующие элементы второй, а значения первых  $d$  элементов второй строки — в соответствующие элементы первой. В результате получаем две новых строки, каждая из которых является комбинацией частей двух родительских строк.

Операция скрещивания создаёт новые строки путём некоторой комбинации значений элементов наиболее ценных в популяции  $G(t)$  строк. Получившиеся в результате строки могут превосходить по ценности родительские строки.

Так как открытие новых областей поиска в операции скрещивания происходит лишь путём перегруппирования имеющихся в популяции комбинаций символов, то при использовании только этой операции некоторые потенциально оптимальные области могут оставаться не рассмотренными. Для предотвращения подобных ситуаций применяется операция мутации.

*Мутация* представляет собой процесс случайного изменения значений элементов строки. Для этого строки, получившиеся на этапе скрещивания, просматриваются поэлементно, и каждый элемент с заданной вероятностью мутации  $p_{\text{мут}}$  может мутировать, т.е. изменять значение на любой случайно выбранный символ, допустимый для данной позиции. Операция мутации позволяет находить новые комбинации признаков, увеличивающих ценность строк популяции.



Итак, в результате описанных выше операций получаем  $K \cdot N$  новых строк, которые либо полностью формируют новую популяцию  $G(t + 1)$  (при  $K=1$ ), заменяя при этом все строки популяции  $G(t)$ , либо составляют часть популяции  $G(t + 1)$ , заменяя собой  $K \cdot N$  наименее ценных строк предыдущей популяции. Затем выбираем наиболее ценную строку популяции, используя критерии оптимизации (по массе, надёжности, габаритам и т.д.).

Приведённые выше возможные подходы к алгоритмизации выбора конструктивно-силовых схем турбомашин имеют свои достоинства и недостатки. Графы и таблицы применимости наглядны и просты, но в тоже время не позволяют открывать новые области поиска оптимального решения, в отличие от генетического алгоритма. При автоматизации выбора конструктивно-силовых схем возможен подход, при котором на каждом этапе выбора используется один из вышеизложенных методов. Например, при создании исходной популяции в генетическом алгоритме можно использовать таблицы применимости. Т.е. при выборе типа структурных элементов турбонасосного агрегата (типа рабочего колеса насоса, типа турбины, количества ступеней насоса и т.д.) пользоваться таблицами применимости, а генетический алгоритм использовать при компоновке, взаимном расположении узлов, подшипников и т.п.

### ПЕРЕЧЕНЬ ССЫЛОК

1. Баулин В.И. Формализация конструирования двигателей летательных аппаратов в САПР: Учебное пособие.- М.: МАИ, 1983.- 49 с.
2. Белоусов А.И. Конструктивные и силовые схемы турбомашин двигателей летательных аппаратов: Учебное пособие.- Куйбышев: КуАИ, 1988.- 92 с.
3. Тихонов Г.Ю. Автоматизация проектирования и оптимизация конструкций полочных бандажей рабочих колёс авиационных газовых турбин: Автореферат диссертации на соискание уч. степ. канд. технич. наук.- Самара: СГАУ, 2001.- 16 с.