ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УДК 519.1

В. Т. Тозик

АЛГЕБРАИЧЕСКИЙ МЕТОД ОПРЕДЕЛЕНИЯ ПОЛНОГО МНОЖЕСТВА ПРОСТЫХ РАЗРЕЗОВ В ДВУХПОЛЮСНЫХ СЕТЯХ

Рассматривается задача поиска простых разрезов в двухполюсных структурно-сложных сетях. В основу предлагаемого метода положена алгебраическая модель сети, базирующаяся на алгебре кубических комплексов. Это позволяет предложить эффективную с точки зрения трудоемкости процедуру определения полного множества простых разрезов.

Ключевые слова: *двухполюсная сеть, простой разрез, структурная функция, алгебра кубических комплексов.*

Задача определения полного множества простых разрезов в двухполюсных сетях нетривиальна. Конструктивный метод ее решения предложен только для плоских графов путем построения полного множества циклов в соответствующих двойственных графах. Однако для достаточно больших графов с нетривиальной структурой (не сводимой к плоской) задача становится непреодолимо сложной ввиду комбинаторных трудностей полного перебора.

В основу предлагаемого в настоящей работе метода положена алгебраическая модель графа, использующая введенную алгебру простых цепей [1] и алгебру кубических комплексов [2], что позволяет предложить достаточно эффективную с точки зрения трудоемкости процедуру определения простых разрезов.

Как было показано в работе [1], структура двухполюсной сети (α, β) может быть представлена булевой структурной функцией (СФ) в дизъюнктивной нормальной форме (ДНФ) с помощью простых цепей (элементарных конъюнкций K_j ранга r):

$$f_{\alpha\beta} = \overset{m}{\underset{j=1}{V}} K_j = \overset{m}{\underset{j=1}{V}} \left[\overset{r}{\underset{i=1}{\Lambda}} x_i \right], \tag{1}$$

где m — число простых цепей сети.

Ту же функцию (1) можно представить в конъюнктивной нормальной форме (КНФ) с помощью простых разрезов (элементарных дизъюнкций D_j ранга s)

$$f_{\alpha\beta} = \overset{t}{\underset{j=1}{\Lambda}} D_i = \overset{t}{\underset{j=1}{\Lambda}} \left[\overset{s}{\underset{i=1}{V}} x_i \right], \tag{2}$$

где t — число простых разрезов двухполюсной сети.

В работе [1] представлен алгоритм, позволяющий определить кубическое покрытие $\prod(L_1)$, соответствующее ДНФ булевой СФ, записанной в виде простых цепей (1). Теперь рассмотрим выражение (2). С помощью правила де Моргана [2] можно перейти к дизъюнктивной форме отрицания булевой СФ:

$$\overline{f}_{\alpha\beta} = \bigvee_{j=1}^{t} \overline{D}_j , \tag{3}$$

$$\overline{D}_j = \overline{x_1 \vee x_2 \vee x_2 \vee \ldots \vee x_s} = \overline{x}_1 \overline{x}_2 \ldots \overline{x}_s . \tag{4}$$

Отрицанию булевой СФ (3) соответствует покрытие $\prod(L_0)$ множества вершин n-мерного куба L_0, на которых данная функция принимает нулевые значения, причем каждой конъюнкции $\overline{D}_j$ ранга s (4) соответствует некоторый куб $C_j \in \prod(L_0)$. Целью настоящей работы является создание метода определения полного множества таких кубов $\bigcup_{j=1}^{t} C_j$, каждый куб C_j которого соответствует конъюнкции $\overline{D}_j$ (4), поставленной в соответствие простому разрезу. Приведем теоретико-множественное представление этой двойственной задачи:

L_1 — подмножество состояний связности сети ($f_{\alpha\beta} = 1$):

$$L_1 \subseteq \Pi(L_1) = \bigcup_{j=1}^{m} C_j \Rightarrow f_{\alpha\beta} = \bigvee_{j=1}^{m} K_j ,$$

K_j — конъюнкция, поставленная в соответствие простой цепи;

L_0 — подмножество состояний несвязности сети ($f_{\alpha\beta} = 0$):

$$L_0 \subseteq \Pi(L_0) = \bigcup_{j=1}^{t} C_j \Rightarrow \overline{f}_{\alpha\beta} = \bigvee_{j=1}^{t} \overline{D}_j ,$$

$\overline{D}_j$ — конъюнкция, поставленная в соответствие простому разрезу.

Введем некоторые определения, основанные на положениях работы [2].

Определение 1. Куб $C = (a_1, a_2, \ldots, a_n)$ размерности n есть n-мерный вектор, каждая координата которого a_i принимает значения из множества $\{0, 1, \mathrm{X}\}$. Координаты $a_i \in \{0,1\}$ называются связанными, $a_i = \mathrm{X}$ — свободными.

Определение 2. Кубы $C_j = (a_1, a_2, \ldots, a_n)$ и $C_s = (b_1, b_2, \ldots, b_n)$ равны между собой $C_j = C_s$, если $\forall_i (a_i = b_i)$.

Определение 3. Кубы $C_j = (a_1, a_2, \ldots, a_n)$ и $C_s = (b_1, b_2, \ldots, b_n)$ находятся в отношении строгого включения $C_j \subset C_s$, если $\exists_i (a_i \in \{0,1\} \,\&\, b_i = \mathrm{X})$ и не $\exists_i (a_i = \mathrm{X} \,\&\, b_i \in \{0,1\} \vee a_i = \overline{b}_i)$.

Определение 4. Кубы C_j и C_s находятся в отношении нестрогого включения $C_j \subseteq C_s$, если $C_j = C_s$ или $C_j \subset C_s$.

Определение 5. Множество кубов $\prod$ и множество вершин L_i находятся в отношении нестрогого включения $L_i \subseteq \prod$, если любая вершина $l \in L_i$ включена в некоторый куб $C \in \prod$, т.е. $l \subseteq C$. В дальнейшем будем говорить, что множество $\prod$ покрывает множество L_i, если $L_i \subseteq \prod$.

Определение 6. Кубы C_j и C_s несравнимы $(C_j \sim C_s)$, если $C_j \not\subseteq C_s$ и $C_s \not\subseteq C_j$.

Ниже дается определение операции объединения, которое отличается от приведенного в работе [2]. Чтобы различать эти две операции, будем обозначать определенную ниже операцию символом “$\overset{+}{\bigcup}$”.

Определение *7.* Результат операции объединения двух кубов C_j и C_s определяется как

$$C_j \overset{+}{\bigcup} C_s = \begin{cases} C_j, & \text{если} \quad C_s \subseteq C_j; \\ C_s, & \text{если} \quad C_j \subseteq C_s; \\ \left\{C_s, C_j\right\}, & \text{если} \quad C_j \sim C_s. \end{cases}$$

Операция объединения $\overset{+}{\bigcup}$ обладает свойством коммутативности и ассоциативности. Результат операции объединения двух множеств кубов $\prod_d$ и $\prod_e$ определяется как множество $\prod = \prod_d \overset{+}{\bigcup} \prod_e$, полученное объединением $\prod_d$ и $\prod_e$ в обычном теоретико-множественном смысле с последующим удалением кубов C_j таких, что $C_j \subseteq C_s$, или C_s таких, что $C_s \subseteq C_j$; $C_j, C_s \in \prod$.

Объединение множества $\prod$ с самим собой приводит к множеству, в котором каждая пара кубов несравнима. В дальнейшем, чтобы отобразить это свойство, будем писать: $\prod = \overset{+}{\bigcup} \prod$.

Определение 8. Результат #-операции двух кубов $C_j = \left(a_1, a_2, \ldots, a_n\right)$ и $C_s = \left(b_1, b_2, \ldots, b_n\right)$ определяется как

$$C_j \# C_s = \begin{cases} C_j, & \text{если } \exists_i \left(a_i = \overline{b_i}\right); \\ \varnothing, & \text{если } C_j \subseteq C_s; \\ \left\{C_1, C_2, \ldots, C_t, \ldots, C_k\right\}, & \text{причем каждой паре координат } \overset{n}{\underset{i=1}{\forall}} \left(a_i, b_i\right) \text{ таких, что} \\ a_i = \mathrm{X}, b_i \neq \mathrm{X}, & \text{соответствует куб } C_t = \left(a_1, a_2, \ldots, a_{i-1}, \overline{b_i}, a_{i+1}, \ldots, a_{n-1}, a_n\right). \end{cases}$$

#-операция некоммутативна и неассоциативна. Для множества кубов и одного куба, а также для двух множеств кубов $\prod_d = \left\{C_1^d, C_2^d, \ldots, C_m^d\right\}$ и $\prod_e = \left\{C_1^e, C_2^e, \cdots, C_k^e\right\}$ #-операция определяется следующим образом:

$$\prod_d \# C_j = \left\{C_1^d \# C_j, C_2^d \# C_j, C_j, \ldots, C_m^d \# C_j\right\};$$

$$\prod_d \# \prod_e = \left(\left(\ldots\left(\left(\prod_d \# C_1^e\right) \# C_2^e\right) \# \ldots\right) \# C_k^e\right).$$

Определение 9. Максимальным для заданного множества кубов $\prod_d$ называется такой куб C_j, что $C_j \# \prod_d = \varnothing$, и при замене в C_j хотя бы одной связанной координаты на свободную $C_j \# \prod_d \neq \varnothing$.

Множество, содержащее все максимальные для $\prod_d$ кубы, будем обозначать max(Π_d) Легко показать, что для $\prod_d$ множество $\max(\prod_d)$ единственное.

Определение 10. Простым кубом C_j для булевой СФ называется такой, что $C_j \in \max\left(\prod\left(L_1\right)\right)$ либо $C_j \in \max\left(\prod\left(L_0\right)\right)$. Иными словами, простым кубом называется такой куб C_j, для которого в $\prod\left(L_1\right)$ или $\prod\left(L_0\right)$ не существует C_s такой, что $C_j \subseteq C_s$.

В дальнейшем множества простых кубов будем обозначать Z.

Можно показать, что

$$\overset{+}{\bigcup}\left(I \# \prod\right) = \max\left(I \# \prod\right), \tag{5}$$

где I — куб размерности n, в котором все компоненты свободные.

Поиск простых разрезов может быть осуществлен в два этапа. Вначале определяется полное множество простых цепей, а затем с помощью введенных выше операций и отношений алгебры кубов определяется полное множество простых разрезов.

В работе [1] было предложено осуществлять решение первой задачи на *n*-мерном кубическом комплексе S^n, это позволило найти покрытие булевой СФ в виде множества кубов $\prod(L_1) = \{C_1, C_2, \ldots, C_m\}$, соответствующих простым цепям.

Отрицанию булевой СФ (3) соответствует покрытие $\prod(L_0)$ множества вершин *n*-мерного куба L_0, на которых данная функция принимает нулевые значения, причем каждой конъюнкции $\bar{D}_j$ (4) соответствует $C_j \in \prod(L_0)$.

Утверждение 1. Если куб $C_j \in \prod(L_0)$ соответствует конъюнкции $\bar{D}_j$ (4), поставленной в соответствие *j*-му простому разрезу, то $C_j \in Z(L_0)$, т.е. C_j является простым кубом для L_0.

Д о к а з а т е л ь с т в о. Каждый простой разрез отличается от другого по крайней мере одним элементом, поэтому любые две конъюнкции $\bar{D}_j$ и $\bar{D}_s$, поставленные в соответствие *j*-му и *s*-му простым разрезам, отличаются друг от друга по крайней мере одной буквой. В соответствии с этим кубы C_j и C_s несравнимы $\left(C_j \sim C_s\right)$, т.е. $C_j \not\subseteq C_s$ и $C_s \not\subseteq C_j$ для $C_j, C_s \in \prod(L_0)$. Поскольку в каждую конъюнкцию $\bar{D}_j$ буквы входят только в инверсном виде и в силу того, что простой разрез является минимальным по включению множеством элементов, для любого куба $C_j \in \prod(L_0)$ с ценой R_j не существует куба $C_t \in \prod(L_0)$ с ценой $R_t \leq R_t - 1$ такого, что $C_j \subset C_t$. Таким образом, все кубы $C_j \in \prod(L_0)$ являются простыми.

Утверждение 2. Множество простых кубов $Z(L_0)$ является покрытием $\prod(L_0)$ множества вершин *n*-мерного куба L_0, каждый куб $C_j \in \prod(L_0)$ которого соответствует конъюнкции $\bar{D}_j$, записанной с помощью простых разрезов.

Д о к а з а т е л ь с т в о. Предположим обратное. Без потери общности можно допустить, что существует простой куб $C_j \in Z(L_0)$ такой, что $C_j \notin \prod(L_0)$. В соответствии с утверждением 1 все кубы $C_j \in \prod(L_0)$ являются простыми. Отсюда $\prod(L_0) \subseteq Z(L_0)$ и можно записать

$$Z(L_0) = \left\{\prod(L_0), C_j\right\}. \tag{6}$$

Так как оба множества простых кубов $Z(L_0)$ и $\prod(L_0)$ покрывают одно и то же множество вершин L_0, и только L_0, то

$$Z(L_0) \equiv \prod(L_0). \tag{7}$$

Поскольку множество простых кубов для L_0 единственно, то из (6) и (7) следует, что либо $C_j \notin Z(L_0)$, либо $C_j \in \prod(L_0)$. Полученное противоречие доказывает данное утверждение.

На основании вышеизложенного можно предложить алгоритм определения множества кубов $\prod(L_0)$, соответствующих множеству простых разрезов.

Алгоритм

1) определить множество кубов $\prod(L_1)$, соответствующих множеству простых цепей, с помощью алгоритма, предложенного в [1];

2) тогда $\prod(L_0) = \overset{+}{\bigcup}\left[I \# \prod(L_1)\right]$;

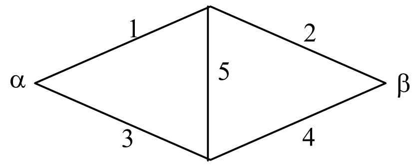
3) конец.

Утверждение 3. В результате алгоритма получается множество кубов $\prod(L_0)$, соответствующее полному множеству простых разрезов.

Доказательство. Для полностью определенной булевой СФ множество простых кубов $Z(L_0)$ совпадает с множеством максимальных кубов для L_0. Поэтому из (5) следует $Z\left(L_0\right)=\overset{+}{\bigcup}\left[I\,\#\prod\left(L_1\right)\right]$, а из утверждения 2 — $\prod(L_0)=Z\left(L_0\right)$. Утверждение доказано.

Замечание. Выполнение $\overset{+}{\bigcup}$-операции можно осуществлять как после завершения $\left[I\,\#\prod\left(L_1\right)\right]$, так и после каждого #-вычитания куба $C_j\in \varPi(L_1)$, проводимого на i-м шаге алгоритма. В приведенном ниже примере используется вторая модификация выполнения $\overset{+}{\bigcup}$-операции.

Пример. Рассмотрим предложенный метод на примере сети, представленной на рисунке. Применение алгоритма определения полного множества простых цепей к данной сети разобрано в [1]. Поэтому рассмотрим только работу алгоритма определения полного множества простых разрезов, считая заданным исходное множество простых цепей и соответствующих им кубов $\prod(L_1)$. Для упрощения примера предположим существование только реберных разрезов



$$\prod(L_1)=\Pi_1=\left\{\begin{matrix}\mathrm{X} & \mathrm{X} & 1 & 1 & \mathrm{X}\\ 1 & 1 & \mathrm{X} & \mathrm{X} & \mathrm{X}\\ 1 & \mathrm{X} & \mathrm{X} & 1 & 1\\ \mathrm{X} & 1 & 1 & \mathrm{X} & 1\end{matrix}\right\}\begin{matrix}C_1^1\\ C_2^1\\ C_3^1\\ C_4^1\end{matrix},$$

$$I=\left(\mathrm{X}\quad \mathrm{X}\quad \mathrm{X}\quad \mathrm{X}\quad \mathrm{X}\right).$$

В соответствии со свойствами #-операции

$$\Pi_2=I\,\#C_1^1=\left\{\begin{matrix}\mathrm{X} & \mathrm{X} & 0 & \mathrm{X} & \mathrm{X}\\ \mathrm{X} & \mathrm{X} & \mathrm{X} & 0 & \mathrm{X}\end{matrix}\right\}\begin{matrix}C_1^2\\ C_2^2\end{matrix},$$

$$\Pi_3=\Pi_2\,\#C_2^1=\left\{\begin{matrix}0 & \mathrm{X} & 0 & \mathrm{X} & \mathrm{X}\\ \mathrm{X} & 0 & 0 & \mathrm{X} & \mathrm{X}\\ 0 & \mathrm{X} & \mathrm{X} & 0 & \mathrm{X}\\ \mathrm{X} & 0 & \mathrm{X} & 0 & \mathrm{X}\end{matrix}\right\}\begin{matrix}\left.\begin{matrix}C_1^3\\ C_2^3\end{matrix}\right\}C_1^2\,\#C_2^1\\ \left.\begin{matrix}C_3^3\\ C_4^3\end{matrix}\right\}C_2^2\,\#C_2^1\end{matrix},$$

$$\Pi_4=\Pi_3\,\#C_3^1=\left\{\begin{matrix}0 & \mathrm{X} & 0 & \mathrm{X} & \mathrm{X}\\ 0 & 0 & 0 & \mathrm{X} & \mathrm{X}\\ \mathrm{X} & 0 & 0 & 0 & \mathrm{X}\\ \mathrm{X} & 0 & 0 & \mathrm{X} & 0\\ 0 & \mathrm{X} & \mathrm{X} & 0 & \mathrm{X}\\ \mathrm{X} & 0 & \mathrm{X} & 0 & \mathrm{X}\end{matrix}\right\}\begin{matrix}C_1^4=C_1^3\,\#C_3^1=C_1^3\\ \left.\begin{matrix}C_2^4\\ C_3^4\\ C_4^4\end{matrix}\right\}\quad C_2^3\,\#C_3^1,\\ C_5^4=C_3^3\\ C_6^4=C_4^3\end{matrix}$$

$$\Pi_5 = \overset{+}{\bigcup}\Pi_4 = \begin{Bmatrix} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & X & X & 0 & X \\ X & 0 & X & 0 & X \end{Bmatrix} \begin{matrix} C_1^5 = C_1^4 \supseteq C_2^4 \\ C_2^5 = C_4^4 \\ C_3^5 = C_5^4 \\ C_4^5 = C_6^4 \supseteq C_3^4 \end{matrix},$$

$$\Pi_6 = \Pi_5 \# C_4^1 = \begin{Bmatrix} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & 0 & X & 0 & X \\ 0 & X & 0 & 0 & X \\ 0 & X & X & 0 & 0 \\ X & 0 & X & 0 & X \end{Bmatrix} \begin{matrix} C_1^6 = C_1^5 \\ C_2^6 = C_2^5 \\ \left.\begin{matrix} C_3^6 \\ C_4^6 \\ C_5^6 \end{matrix}\right\} C_3^5 \# C_4^1 \\ C_6^6 = C_4^5 \end{matrix},$$

$$\Pi(L_0) = \Pi_7 = \overset{+}{\bigcup}\Pi_6 = \begin{Bmatrix} 0 & X & 0 & X & X \\ X & 0 & 0 & X & 0 \\ 0 & X & X & 0 & 0 \\ X & 0 & X & 0 & X \end{Bmatrix} \begin{matrix} C_1^7 = C_1^6 \supseteq C_4^6 \\ C_2^7 = C_2^6 \\ C_3^7 = C_5^6 \\ C_4^7 = C_6^6 \supseteq C_3^6 \end{matrix}.$$

Покрытие Π_7 представляет собой множество кубов $\Pi(L_0)$, соответствующее полному множеству простых разрезов: $\{(1,3),(2,3,5),(1,4,5),(2,4)\}$.

Оценка трудоемкости метода. Как правило, наиболее эффективными являются алгоритмы с трудоемкостью, степенной относительно размерности задачи. Понятие „степенного“ алгоритма близко к принятому в зарубежной литературе определению „хорошего“ алгоритма. Степенная оценка наглядно поясняется с позиций программирования. Линейную оценку имеют алгоритмы, просматривающие информацию единственный раз. Квадратичная оценка связана с „циклом в цикле“. Дальнейший рост порядка оценки соответствует наличию в алгоритме более длинных цепочек вложенных друг в друга циклов.

Другой класс составляют переборные алгоритмы, связанные с просмотром возможных ситуаций — „кандидатов в ответ“ задачи. Обычно число ситуаций экспоненциально возрастает относительно размерности задачи. Тем самым экспоненциальная (а тем более факториальная) трудоемкость переборного алгоритма растет быстрее, чем любая степенная функция.

В работе [1] доказана теоретическая эффективность алгоритма определения простых цепей. Оценка трудоемкости для него, измеряемая числом Δ-операций, ниже квадратичной относительно размерности задачи, представляемой числом простых цепей сети.

Трудоемкость алгоритма определения простых разрезов оценим по числу #-операций и операций сравнения кубов при выполнении $\overset{+}{\bigcup}$-операции. Число простых цепей будем по-прежнему обозначать буквой *m*, а число простых разрезов буквой *t*. Размерность задачи определяется числом простых разрезов в сети.

Число #-операций может быть оценено как произведение числа кубов *m* в исходном покрытии $\Pi(L_1)$ на среднее число кубов в покрытии Π_i при выполнении $\left(\Pi_i \# C_j^1\right)$. Среднее число кубов в Π_i может быть принято равным *t*, поскольку после выполнения

$\Pi_{i+1} = \overset{+}{\bigcup}\left(\Pi_i \# C_j^1\right)$ число кубов в Π_{i+1} приблизительно равно *t*. Таким образом, число #-операций при выполнении алгоритма приблизительно равно mt.

Число попарных операций сравнения при выполнении $\overset{+}{\bigcup}\Pi_i$ равно $g(g-1)/2$, где g — число кубов в Π_i. Приняв поправочный коэффициент, учитывающий превышение числа кубов в $\left(\Pi_i \# C_j^1\right)$ над $\overset{+}{\bigcup}\left(\Pi_i \# C_j^1\right)$, равным в среднем *k*, можно записать, что число сравнений при выполнении $\overset{+}{\bigcup}$-операции равно $kt(kt-1)/2 \approx (kt)^2/2$. Таким образом, общая трудоемкость алгоритма может быть оценена как $T \cong mt + (kt)^2/2$, т.е. трудоемкость алгоритма пропорциональна квадрату размерности задачи. Таким образом, можно сделать вывод об эффективности предложенного метода определения полного множества простых разрезов в двухполюсных сетях. Представление структуры сети в виде кубов кубического комплекса позволяет возможность предложить черезвычайно эффективные методы поиска простых цепей и разрезов, поскольку множества кубов хорошо представляются в памяти компьютера массивами двоичных кодов, а операции над ними — соответствующими программными средствами. Предложенные методы алгоритмически просты и не накладывают никаких ограничений на структуру исследуемых сетей.

СПИСОК ЛИТЕРАТУРЫ

1. *Тозик В. Т.* Математический аппарат для анализа структурных свойств сетей // Изв. вузов. Приборостроение. 2010. Т. 53, № 12. С. 22—30.

2. *Миллер Р.* Теория переключательных схем. Т.1. М.: Наука, 1970. 416 с.

Сведения об авторе

Вячеслав Трофимович Тозик — канд. техн. наук, доцент; Санкт-Петербургский государственный университет информационных технологий, механики и оптики, кафедра инженерной и компьютерной графики; заведующий кафедрой;
E-mail: tozik@mail.ifmo.ru

Рекомендована кафедрой
инженерной и компьютерной графики

Поступила в редакцию
11. 02.10 г.