

酶动力学的图形理论*

——定态反应体系

周 国 城 江 寿 平

(中国科学院上海生物化学研究所)

刘 为 民

费 志 浩

(上海群益机械厂技校)

(上海计算技术研究所)

摘 要

在酶动力学中采用图形方法,能够直接把计算规则和表征反应体系的图形联系起来,这不仅使计算较为直观和方便,且有助于对反应体系的分析与讨论。

本文分析了目前国际上在酶动力学中所采用的图形计算方法,指出其缺陷所在。然后提出一套新的酶动力学的图形计算方法,包含三个法则。法则 1 有助于纠正现有方法在计算时常易出现的一些差错。法则 2 和法则 3 则分别为计算含酶物浓度的分子和分母提供了新的图形方法,可用来处理较复杂的酶反应体系。

在附录中,还给出了这三个法则的数学原理。

定态反应体系是酶学研究中的一个重要组成部分。虽然它比非定态体系处理起来要方便得多,但现代酶学中遇到的定态问题,往往涉及到一个酶分子与多个底物(包括变构抑制物、变构活化物等)分子间的相互协同作用。因此对于这样的酶反应体系,若用通常的数学方法来处理是非常麻烦的。为此,King 和 Altman^[1]提出了用图形方法来计算。随后,Volkenstein 和 Goldstein^[2]及 Fromm^[3]又分别从不同的角度对 King-Altman 的图形方法作了改进。

一般说来,在酶动力学中应用图形方法,能够把计算规则与图形直接联系起来,这就不便于计算,而且也使反应图象变得更为直观,从而有助于对反应体系的分析与讨论^[4]。然而,下面我们将指出,在现有的图形方法中,又各有其不足之处。特别是在处理较复杂的反应体系时,问题就显得更为严重。因此有必要加以改进。

首先从一般的问题着手。考虑某个有 n 种不同含酶物 $E_i (i = 1, 2, \dots, n)$ 的反应体系。设 $E_i \rightarrow E_j$ 的速度常数是 k_{ij} , $E_j \rightarrow E_i$ 的速度常数是 k_{ji} 。注意: 这里 k_{ij} 是一级速度常数或假一级速度常数。为了数学表达的简便起见,根据 k_{ij} 的定义,我们可以规定 $k_{ii} = 0$ 。于是,当反应体系处于定态时,就有

本文 1977 年 6 月 6 日收到。

* 参与这项工作的还有李家钟(上海交通大学)等同志。

$$e_i \sum_{j=1}^n k_{ij} - \sum_{j=1}^n k_{ji} e_j = 0, (i = 1, 2, \dots, n), \quad (1)$$

$$\sum_{i=1}^n e_i = e_0, \quad (2)$$

式中 e_i, e_j 分别代表 E_i, E_j 的浓度¹⁾, e_0 是体系中所有含酶物的总浓度. 方程组(1)是线性相关的, 因为这 n 个方程的左边之和恒等于 0. 为了求得定解, 可以去掉(1)式中的任一个方程, 例如第 m 个, 补上方程(2), 便可得到关于 n 个未知数的 n 个联立方程. 根据 Cramer 法则, 我们有

$$\frac{e_m}{e_0} = \frac{N_m}{D}, \quad (3)$$

式中

$$N_m = \begin{vmatrix} \text{(第 } m \text{ 列)} \\ \Sigma k_{1j} - k_{21} \cdots 0 \cdots -k_{n1} \\ -k_{12} \Sigma k_{2j} \cdots 0 \cdots -k_{n2} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ 1 \quad 1 \cdots 1 \cdots 1 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ -k_{1n} - k_{2n} \cdots 0 \cdots \Sigma k_{nj} \end{vmatrix} \quad \text{(第 } m \text{ 行)} = \begin{vmatrix} \text{(第 } m \text{ 列)} \\ \Sigma k_{1j} - k_{12} \cdots \cdots -k_{1n} \\ -k_{21} \Sigma k_{2j} \cdots \cdots -k_{2n} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ \cdots \cdots \cdots \cdots \cdots \cdots \\ -k_{n1} - k_{n2} \cdots \cdots \Sigma k_{nj} \end{vmatrix} \quad \text{(第 } m \text{ 行)}, \quad (4)$$

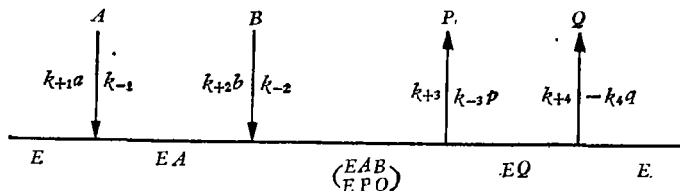
$$D = \begin{vmatrix} \Sigma k_{1j} - k_{21} \cdots -k_{m1} \cdots -k_{n1} \\ -k_{12} \Sigma k_{2j} \cdots -k_{m2} \cdots -k_{n2} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ 1 \quad 1 \cdots 1 \cdots 1 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ -k_{1n} - k_{2n} \cdots -k_{mn} \cdots \Sigma k_{nj} \end{vmatrix} = \begin{vmatrix} \Sigma k_{1j} - k_{12} \cdots 1 \cdots -k_{1n} \\ -k_{21} \Sigma k_{2j} \cdots 1 \cdots -k_{2n} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ -k_{m1} - k_{m2} \cdots 1 \cdots -k_{mn} \\ \vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ -k_{n1} - k_{n2} \cdots 1 \cdots \Sigma k_{nj} \end{vmatrix}. \quad (5)$$

根据方程组(1)的线性相关性以及行列式的性质, 不难看出 D 是同 m 的选取无关的. 故由(2)和(3)式可得

$$e_m = \frac{N_m}{\sum_{i=1}^n N_i} e_0. \quad (6)$$

由此可见, N_m 实际上是体系中第 m 个含酶物浓度的相对比数. 虽然原则上我们可以用行列式展开的方法来求得 N_m 和 D , 或者求得所有的 $N_i (i = 1, 2, \dots, n)$, 然后分别代到(3)或(6)式以求得 e_m . 然而, 实际上除了那些非常简单情况外, 用这样的方法来计算含酶物的浓度的极繁的, 而且很容易出差错.

为了克服这个困难, King 和 Altman^[1] 首先提出用图形方法来进行计算. 按照他们的方法, 对于如下的有序 $B_i B_i$ 反应²⁾



1) 本文规定: 体系中的反应物一律用大写字母表示, 而其浓度则用相应的小写字母来表示. 例如对于底物 $A, B, C, \dots$, 我们分别用 $a, b, c, \dots$ 来表示它们的浓度.

2) 这里我们用近年来国际上较流行的 Cleland^[5] 表示方式来描述酶反应体系.

若取 $E_1 = E$, $E_2 = EA$, $E_3 = EQ$, $E_4 = \begin{pmatrix} EAB \\ EPQ \end{pmatrix}$, 则可得到如图 1 所示的赋权有向图¹⁾ $\mathcal{D}$ (K)。以后我们简称之为有向图 $\mathcal{D}$ 。King 和 Altman 指出, 含酶物 E_1 的浓度的相对比数 N_1 可以表示成:

$$\begin{aligned}
 N_1 = & \begin{array}{c} E_1 \xleftarrow{k_{-1}} \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{-2}} \bullet \\ \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} E_1 \xleftarrow{k_{-1}} \bullet \xleftarrow{k_{+3}} \bullet \xleftarrow{k_{-2}} \bullet \\ \bullet \xleftarrow{k_{+3}} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} E_1 \xleftarrow{k_{-1}} \bullet \xleftarrow{k_{-2}} \bullet \xleftarrow{k_{+3}p} \bullet \\ \bullet \xleftarrow{k_{+3}p} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} E_1 \xleftarrow{k_{-1}} \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{+2}b} \bullet \\ \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{+2}b} \bullet \end{array} \quad (7) \\
 = & k_{-1} k_{-2} k_{+4} + k_{-1} k_{+3} k_{+4} + k_{-1} k_{-2} k_{+3}p + k_{+2} k_{+3} k_{+4} b.
 \end{aligned}$$

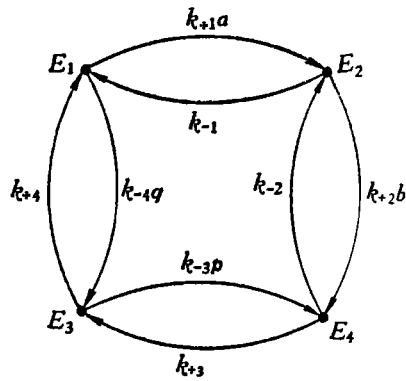


图 1 表示酶反应体系的赋权有向图 $\mathcal{D}(k)$

换言之, (7) 式右边中的每一项实际上相当于有向图 $\mathcal{D}$ 中以 E_1 为汇的一棵支撑入树, 而 N_1 实际上就等于所有这些支撑入树上各条弧的权之积的和。一般在计算 N_m 时, 则应考虑以 E_m 为汇的支撑入树。故对于 N_2, N_3, N_4 我们分别有

$$\begin{aligned}
 N_2 = & \begin{array}{c} \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{+1}a} E_2 \xleftarrow{k_{-2}} \bullet \\ \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} \bullet \xleftarrow{k_{+3}p} \bullet \xleftarrow{k_{+1}a} E_2 \xleftarrow{k_{-2}} \bullet \\ \bullet \xleftarrow{k_{+3}p} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} \bullet \xleftarrow{k_{-4}q} \bullet \xleftarrow{k_{+1}a} E_2 \xleftarrow{k_{-2}} \bullet \\ \bullet \xleftarrow{k_{-4}q} \bullet \xleftarrow{k_{-2}} \bullet \end{array} + \begin{array}{c} \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{+1}a} E_2 \xleftarrow{k_{+2}b} \bullet \\ \bullet \xleftarrow{k_{+4}} \bullet \xleftarrow{k_{+2}b} \bullet \end{array} \\
 = & k_{+1} k_{-2} k_{+4} a + k_{+1} k_{-2} k_{+3} ap + k_{-2} k_{+3} k_{-4} pq + k_{+1} k_{+3} k_{+4} a, \\
 N_3 = & k_{-1} k_{-2} k_{-4} q + k_{-1} k_{+3} k_{-4} q + k_{+2} k_{+3} k_{-4} bq + k_{+1} k_{+2} k_{+3} ab, \\
 N_4 = & k_{+1} k_{+2} k_{+4} ab + k_{-1} k_{-3} k_{-4} pq + k_{+2} k_{-3} k_{-4} bpq + k_{+1} k_{+2} k_{-3} abp.
 \end{aligned}$$

1) 为了便于论述, 同时也为了论述的严密起见, 有必要在文中引入一些图论术语。这些术语的含义, 可参阅附录 I

将上述结果代入(6)式,即可求得含酶物的浓度。

由此可见,King-Altman 方法能够把计算结果中的每一项与图形一一对应起来,这是它的一大优点。然而,对于较复杂的反应体系,支撑入树的数目一般比较多,计算时往往会遗漏掉一些树,从而导致错误的结果。

虽然 Volkenstein 和 Goldstein^[2] 应用图的拓扑理论对若干特殊的有向图提出了一些简化计算的步骤,但这并不能一般地把上述困难排除掉。例如,对于图 3 所示的有向图,Volkenstein 和 Goldstein 的简化步骤就难以生效。而这个有向图的支撑树的数目却是比较多的,故在计算时要判断是否发生树的遗漏仍然不是一件容易的事。

之后,Fromm^[3] 曾提出一种试图不通过考虑支撑入树而能算得结果的方法。这种方法实质上是利用类似于布尔代数中的“逻辑或”的运算来代替一般的加法。按照他的方法虽可立即写出形式解,但展开和消去的工作量却很大,且容易出差错,特别是在识别和消去那些含有回路的项时,很容易搞错,往往会把应该消去的项保留了下来,而把不该消去的项却丢掉了。

因此,考虑如下的问题就显得十分必要:能否找到一种简便的方法可事先算得所求结果中项的数目呢?或用图论的术语来讲,是否有一种方法能事先算得有向图中以任一点为汇的支撑入树的数目呢?

更值得我们考虑的是,对于较复杂的反应体系,即使知道了以某点为汇的支撑入树的数目,但由于树的数目较多,要把所有的支撑入树都画出来仍然是一件相当麻烦的事。对于这种较复杂的反应体系,是否能给出一种较为简便的图形计算方法呢?

此外,无论是按照 King-Altman 方法,还是按照 Volkenstein-Goldstein 方法,抑或是 Fromm 方法,都只有把所有的 $N_i (i = 1, 2, \dots, n)$ 统统算出来之后,才能真正算得某个含酶物的浓度。因此,按照他们的方法,计算一个含酶物浓度的工作量,就几乎与计算所有含酶物浓度的工作量一样了。而在实际问题中,通常感兴趣的是计算产物的生成率。这时,往往只需要求得反应体系中的一、二个含酶物的浓度就够了,完全不必把所有含酶物的浓度都算出来。在这种情况下,是否有一种较为简便的计算方法呢?

下面,我们将直接给出解决上述三个问题的具体方法,而它们的原理将放到附录来论述。

二

I. 法则 1 (计算 N_m 中项的数目 P_m)

(1) 按照表征反应体系的有向图 $\mathcal{D}(K)$ 作一个矩阵 $\mathbf{A}(\mathcal{D}) = [a_{ij}]$, 其中

$$a_{ij} = \begin{cases} 1, & \text{若 } k_{ij} \neq 0, \\ 0, & \text{若 } k_{ij} = 0. \end{cases} \quad (8)$$

(2) 再作矩阵 $\mathbf{B}(\mathbf{A}) = [b_{ij}]$, 其中

$$b_{ij} = \begin{cases} \sum_{i=1}^n a_{ii}, & \text{若 } i = j, \\ -a_{ij}, & \text{若 } i \neq j. \end{cases} \quad (9)$$

(3) 所求的 P_m 就等于矩阵 $\mathbf{B}(\mathbf{A})$ 的第 m 行中任意元素(例如 (m, l) 元素)的代数余子式,即

$$P_m = \text{cof} [\mathbf{B}(\mathbf{A})]_{m,l}, \quad (10)$$

式中符号 cof 表示对后面指定的矩阵元素取代数余子式。

(4) 如果 $\mathcal{D}$ 是一个对称有向图¹⁾, 那末 P_m 与 m 的选取无关. 在这种情况下, 不妨把它们的公共值记为 P .

若要知道法则 1 的数学原理, 请参阅附录 II.

例 1. 对图 1 所示的酶反应体系, 按法则 1 我们有

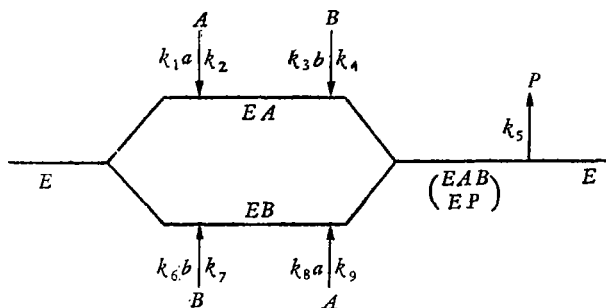
$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}(\mathbf{A}) = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}.$$

由于图 1 是对称有向图, 故有

$$P = \text{cof} [\mathbf{B}(\mathbf{A})]_{1,1} = \begin{vmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ -1 & -1 & 2 \end{vmatrix} = 4,$$

即 $N_m (m = 1, 2, 3, 4)$ 中项的数目均为 4. 例如, 对于 N_1 而言, 这 4 项就分别对应于 (7) 式中以 E_1 为汇的 4 棵支撑入树.

例 2. 考虑如下无序 *Bi Uni* 反应体系



若取 $E_1 = E$, $E_2 = EA$, $E_3 = EB$, $E_4 = \begin{pmatrix} EAB \\ EP \end{pmatrix}$, 便得到如图 2 所示的有向图 $\mathcal{D}$. 应用法则 1, 算得:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}(\mathbf{A}) = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix},$$

1) 所谓对称有向图就是指若存在弧 $\widehat{E_i E_j}$, 则必定也存在弧 $\widehat{E_j E_i}$ 的有向图, 对这种有向图, $\mathbf{A}$ 和 $\mathbf{B}(\mathbf{A})$ 都是对称矩阵, 故 $\mathbf{B}(\mathbf{A})$ 中的任意元素的代数余子式都相等, 它们的公共值就等于有向图中以任一点为汇的支撑入树的数目.

$$P_1 = \text{cof} [\mathbf{B}(\mathbf{A})]_{1,1} = \begin{vmatrix} 2 & 0 & -1 \\ 0 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix} = 8,$$

$$P_3 = P_2 = \text{cof} [\mathbf{B}(\mathbf{A})]_{2,1} = - \begin{vmatrix} -1 & -1 & 0 \\ 0 & 2 & -1 \\ -1 & -1 & 3 \end{vmatrix} = 6,$$

$$P_4 = \text{cof} [\mathbf{B}(\mathbf{A})]_{4,1} = - \begin{vmatrix} -1 & -1 & 0 \\ 2 & 0 & -1 \\ 0 & 2 & -1 \end{vmatrix} = 4.$$

故总共有 $\sum_{m=1}^4 P_m = 24$ 棵支撑入树. 本例中的 $\mathcal{D}$ 不是对称有向图, 故 P_m 的值不完全一样.

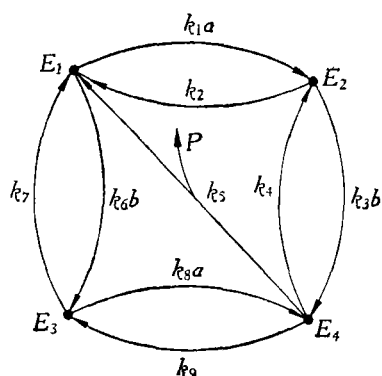


图 2 例 2 中酶反应体系的有向图 $\mathcal{D}$

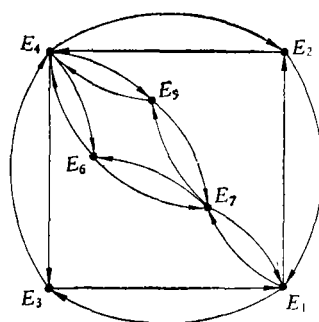


图 3 例 3 中酶反应体系的有向图 $\mathcal{D}$

例 3. 对于图 3 所示的酶反应体系 $(E_1 = E, E_2 = EA, E_3 = EB, E_4 = \begin{pmatrix} EAB \\ EPQ \end{pmatrix}, E_5 = EP, E_6 = EQ, E_7 = F)^{1)}$, 用法则 1 算得:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{B}(\mathbf{A}) = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & 0 & -1 \\ -1 & 2 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 2 & -1 \\ -1 & 0 & 0 & 0 & -1 & -1 & 3 \end{bmatrix}.$$

图 3 是对称有向图, 故以任一点为汇的支撑入树的数目都相等, 其公共值

$$P = \text{cof} [\mathbf{B}(\mathbf{A})]_{1,1} = 48.$$

这说明对于较复杂的反应体系, N_m 中项的数目是相当大的. 如果没有一种方法能事先算得

1) 本例虽取自文献[3], 但对其中诸 $E_i (i = 1, 2, \dots, 7)$ 分别所代表的具体含酶物作了改正. 因为定态测量一般不能探测到异构化作用, 除非这个含酶物的两个异构物都能和体系中的反应物结合^[4].

这个数,则在应用图形方法计算时,是难以判断自己所得的结果是否可靠的. 这个例子同时也说明,对于较复杂的反应体系,应用 King 等人的方法来计算就得先画出很多支撑入树,这本身也是一件相当麻烦的事. 为了克服这个困难,下面我们提出一种新的计算 N_m 的图形方法.

II. 法则 2 (一种新的计算 N_m 的方法)

(1) 先画出表征酶反应体系的赋权有向图 $\mathcal{D}$.

(2) 在 $\mathcal{D}$ 中的各点 $E_i (i = 1, 2, \dots, n)$ 上分别添加一个权为 $\sum_{i=1}^n k_{ii}$ 的圈.

(3) 然后将点 E_m 以及连接该点的弧统统抹去;而把其余各弧的权一一变号,即把 k_{ij} 改为 $-k_{ij} \begin{pmatrix} i, j \neq m \\ i \neq j \end{pmatrix}$.

(4) 经(1)–(3)而得到的赋权有向图记为 $\mathcal{D}_m^+$, 于是所求的 N_m 就可表示为:

$$N_m = \sum_i (-1)^{q_i^m} f(h_i^m). \quad (11)$$

式中 h_i^m 是 $\mathcal{D}_m^+$ 中的第 i 个 1-因子, 即由互不相交的回路所组成的 $\mathcal{D}_m^+$ 的第 i 个支撑子图; q_i^m 是 h_i^m 中包含偶数条弧的回路数目; $f(h_i^m)$ 代表 h_i^m 中所有弧的权之积.

(5) 为便于校核,也可按下式算得 $\mathcal{D}_m^+$ 中 1-因子的数目 n_m^+ :

$$n_m^+ = \text{per } \mathbf{A}(\mathcal{D}_m^+).$$

式中 $\mathbf{A}(\mathcal{D}_m^+)$ 是按(8)式对 $\mathcal{D}_m^+$ 写出的矩阵, 符号 per 表示对后面的矩阵取其承袭式值.

关于法则 2 的数学原理, 及矩阵承袭式的定义、性质与计算方法, 请分别参阅附录 III 和附录 IV.

例 4. 考虑图 4 所示的酶反应体系. 若用 King 等人的方法来计算 N_1 , 得

$$\begin{aligned} N_1 &= \begin{array}{c} E_1 \\ \nearrow k_{31} \\ \bullet \\ \searrow k_{23} \end{array} + \begin{array}{c} E_1 \\ \nearrow k_{31} \quad \searrow k_{21} \\ \bullet \end{array} + \begin{array}{c} E_1 \\ \bullet \\ \searrow k_{21} \\ \nearrow k_{32} \end{array} \\ &= k_{23} k_{31} + k_{21} k_{31} + k_{32} k_{21}. \end{aligned}$$

现用法则 2 来计算: 取 $m = 1$, 然后按该法则的变换方法将图 4 的 $\mathcal{D}$ 变为图 5 所示的 $\mathcal{D}_1^+$,

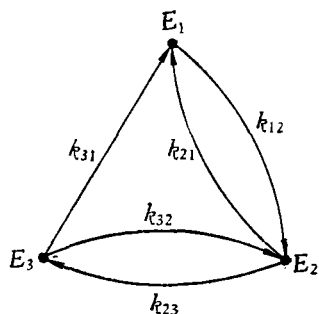


图 4 例 4 中酶反应体系的有向图 $\mathcal{D}$

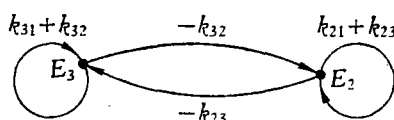


图 5 由图 4 按法则 2 而得到的变换图 $\mathcal{D}_1^+$

于是由(11)式算得

$$N_1 = (-1)^0 \left\{ \begin{array}{c} \text{Diagram 1} \\ k_{31} + k_{32} \quad k_{21} + k_{23} \end{array} \right\} + (-1)^1 \left\{ \begin{array}{c} \text{Diagram 2} \\ -k_{32} \quad -k_{23} \end{array} \right\}$$

$$= (k_{31} + k_{32})(k_{21} + k_{23}) - k_{32}k_{23} = k_{23}k_{31} + k_{21}k_{31} + k_{32}k_{21}.$$

由此可见, 两种计算方法所得的结果完全一致. 前者要考虑 3 棵支撑入树——这也可由法则 1 事先算得; 后者要考虑 2 个 1-因子——这也可由法则 2 的 (5) 事先算得. 事实上, 由图 5, 我们得

$$n_1^+ = \text{per } \mathbf{A}(\mathcal{D}_1^+) = \text{per} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = 2.$$

这说明在上述应用法则 2 的计算中, 我们没有遗漏掉 1-因子. 虽然对于这样简单的酶反应体系, 还不能看出法则 2 的优越性, 但下面我们将看到, 对于较复杂的反应体系, 用法则 2 来计算的优越性就十分明显了.

例 5. 计算图 6 所示反应体系中的 N_1 . 按照法则 2, 取 $m = 1$, 将图 6 的 $\mathcal{D}$ 变为图 7 所示的 $\mathcal{D}_1^+$. $\mathcal{D}_1^+$ 的 1-因子数目为

$$n_1^+ = \text{per } \mathbf{A}(\mathcal{D}_1^+) = \text{per} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = 6.$$

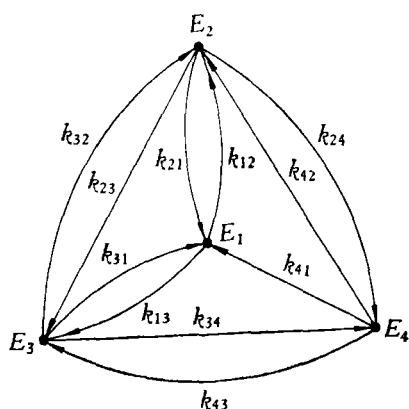


图 6 例 5 中酶反应体系的有向图 $\mathcal{D}$

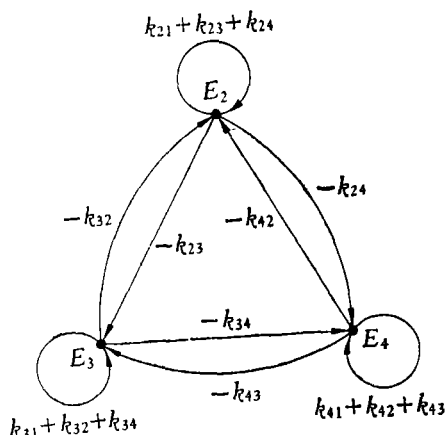


图 7 由图 6 按法则 2 而得到的变换图 $\mathcal{D}_1^+$

图 8 给出了这 6 个 1-因子以及与之相应的 $(-1)^{a_i} f(h_i^+)$ ($i = 1, 2, \dots, 6$) 的值, 把它们加起来即得到 N_1 . 但如果用 King 等人的方法来计算, 由法则 1 知, 就得考虑 16 棵支撑入树, 这显然工作量要大得多.

例 6. 考虑图 3 所示的酶反应体系. 若用 King 等人的方法来计算 N_4 , 得画出 48 棵支撑入树 (参阅例 3). 现让我们看看用法则 2 需考虑多少个 1-因子. 图 9 给出了按法则 2 (取 $m = 4$) 而得到的变换图 $\mathcal{D}_4^+$. 它的 1-因子的个数为:

$$n_4^+ = \text{per } \mathbf{A}(\mathcal{D}_4^+) = \text{per} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} = 10.$$

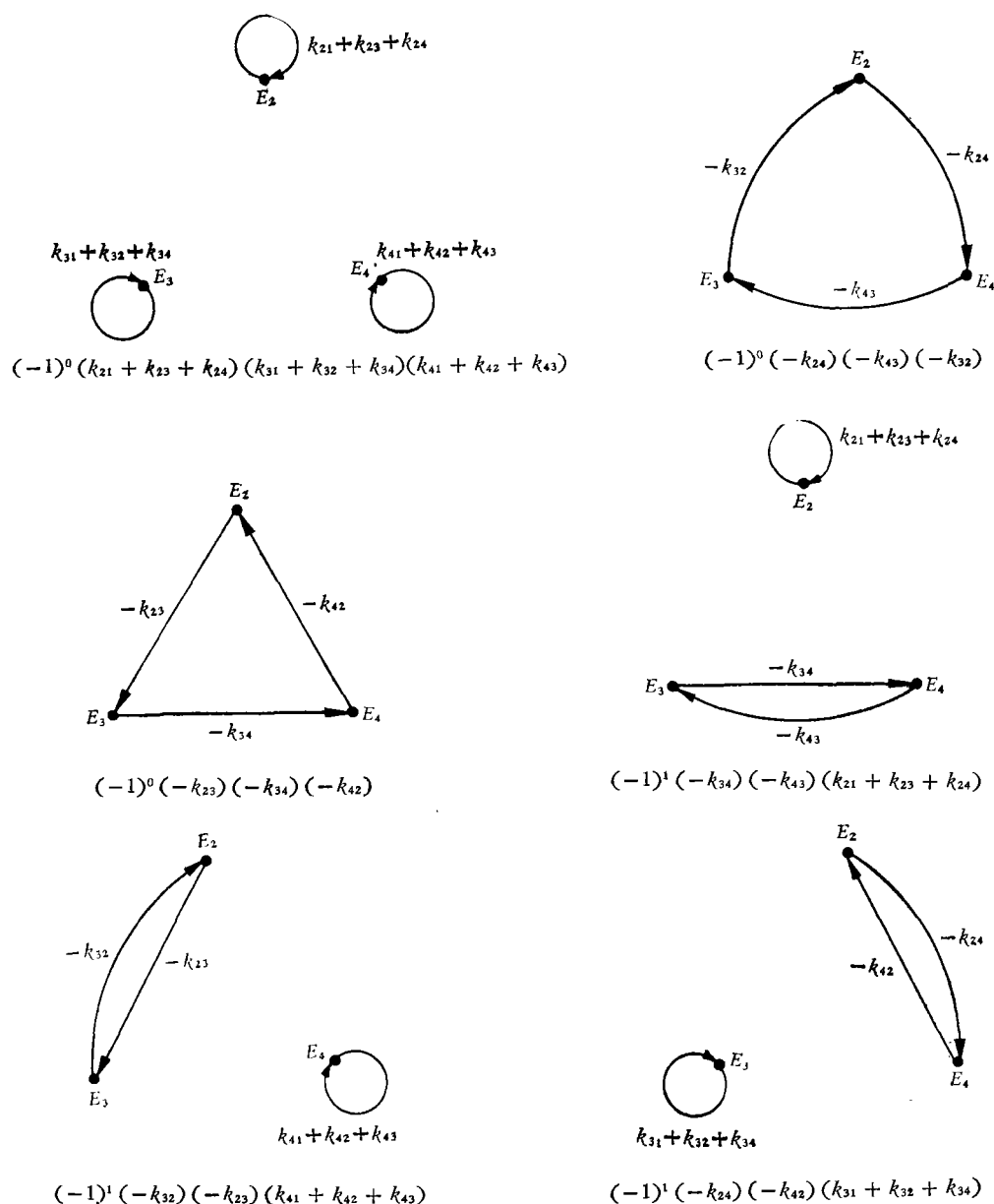


图 8 由图 7 中的 $\mathcal{D}^+$ 分解出来的 6 个 1-因子及其相应的 $(-1)^{q_i} f(h_i^1)$ ($i = 1, 2, \dots, 6$) 的值

即只需考虑 10 个 1-因子, 这显然要简单得多。

III. 法则 3 (一种直接计算 D 的方法)

(1) 画出表征反应体系的赋权有向图 $\mathcal{D}$ 。在 $\mathcal{D}$ 中任取一点 (例如 E_m) 为参考点。然后在点 E_m 上添加一个权为 1 的圈, 而在所有其它各点 E_i ($i \neq m$) 上分别添加一个权为 $\sum_{i=1}^n k_{ii}$ 的圈。

(2) 将所有以参考点 E_m 为终点的弧 $\widehat{E_i E_m}$ ($i \neq m$) 的权都改为 1; 若 $\mathcal{D}$ 中不存在弧

$\widehat{E_i E_m}$, 即 $\widehat{E_i E_m}$ 的权为 0, 也要添上 $\widehat{E_i E_m}$, 并赋之以权 1.

(3) 把其余弧权变号, 即把 k_{ij} 改为 $-k_{ij} \begin{pmatrix} i \neq j \\ j \neq m \end{pmatrix}$.

(4) 经 (1)–(3) 而得到的赋权有向图记为 $\mathcal{D}_m^*$. 于是所求的分母 D 就可表示为:

$$D = \sum_i (-1)^{Q_i^m} f(H_i^m), \quad (12)$$

式中 H_i^m 是 $\mathcal{D}_m^*$ 中的第 i 个 1-因子, Q_i^m 是 H_i^m 中包含偶数条弧的回路数目, $f(H_i^m)$ 代表 H_i^m 中所有弧的权之积.

(5) $\mathcal{D}_m^*$ 中 1-因子的数目 $n_m^* = \text{per } \mathbf{A}(\mathcal{D}_m^*)$.

关于法则 3 的数学原理请参阅附录 III.

例 7. 对于图 2 所示的反应体系, 产物的生成率

$$v = \frac{dp}{dt} = k_5 e_4 = k_5 \frac{N_4}{D} e_0.$$

我们用法则 3 来计算上式中的分母 D . 若取图 2 中的 E_1 为参考点, 就得到如图 10 所示的赋权有向图 $\mathcal{D}_1^*$. 它的 1-因子数目为:

$$n_1^* = \text{per } \mathbf{A}(\mathcal{D}_1^*) = \text{per} \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} = 11.$$

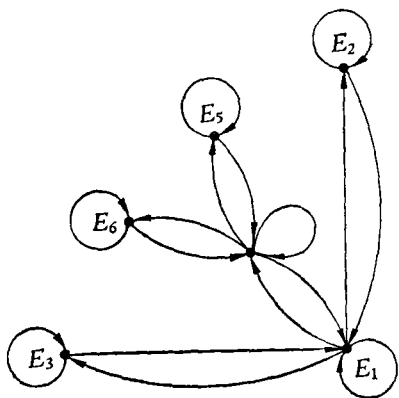


图 9 由图 3 按法则 2 得到的变换图 $\mathcal{D}_1^*$

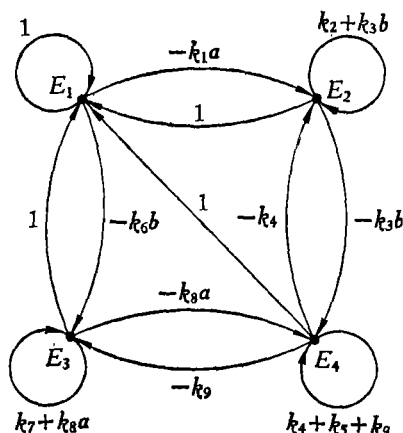


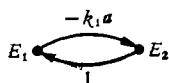
图 10 由图 2 按法则 3 得到的变换图 $\mathcal{D}_1^*$

图 11 给出了这 11 个 1-因子以及与之相应的 $(-1)^{Q_i^1} f(H_i^1)$ ($i = 1, 2, \dots, 11$) 的值, 把它们加起来, 便得到 D . 而若用 King 等人的方法来计算, 就得考虑 $\sum_{m=1}^4 p_m = 24$ 棵支撑入树 (见例 2).

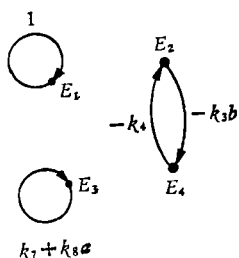
例 8. 对于图 3 所示的反应体系, 应用法则 3 并取 E_4 为参考点, 我们得



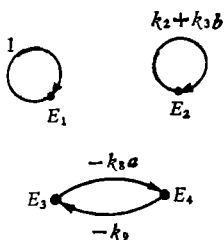
$$(-1)^0 \cdot 1 \cdot (k_2 + k_3b) \\ (k_7 + k_8a)(k_4 + k_5 + k_9)$$



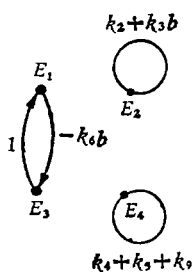
$$(-1)^1 \cdot 1 \cdot (-k_1a)(k_7 + k_8a) \\ (k_4 + k_5 + k_9)$$



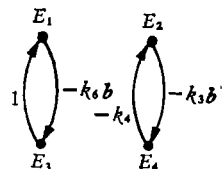
$$(-1)^1 \cdot 1 \cdot (-k_3b)(-k_4) \\ (k_7 + k_8a)$$



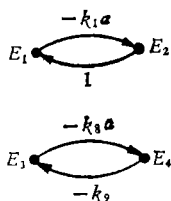
$$(-1)^1 \cdot 1 \cdot (-k_8a)(-k_9) \\ (k_2 + k_3b)$$



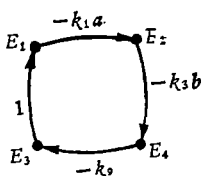
$$(-1)^1 \cdot 1 \cdot (-k_6b)(k_2 + k_3b) \\ (k_4 + k_5 + k_9)$$



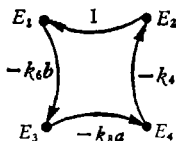
$$(-1)^2 \cdot 1 \cdot (-k_6b)(-k_4)(-k_3b)$$



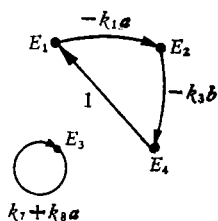
$$(-1)^2 \cdot 1 \cdot (k_1a)(-k_8a)(-k_9)$$



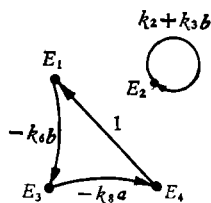
$$(-1)^1 \cdot 1 \cdot (-k_1a)(-k_3b)(-k_9)$$



$$(-1)^1 \cdot 1 \cdot (-k_4)(-k_8a) \\ (-k_6b)$$



$$(-1)^0 \cdot 1 \cdot (-k_1a)(-k_3b)(k_7 + k_8a)$$



$$(-1)^0 \cdot 1 \cdot (-k_6b)(-k_8a)(k_2 + k_3b)$$

图 11 由图 10 中的 $\mathcal{Q}_1^*$ 分解出来的 11 个 1-因子及其相应的 $(-1)^{Q_i} (H_i^!)$ ($i = 1, 2, \dots, 11$) 的值

$$n_1^* = \text{per} \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} = 74,$$

即要考虑74个1-因子。但若按照 King 等人的方法去算,就得考虑 $48 \times 7 = 336$ 棵支撑入树(参见例3)。

当然,我们也可用法则2把所有的 N_i 都算出,然后取其和而得到 D 。显然,对于较复杂的反应体系,这样来计算 D 也比 King 等人的方法方便得多,但是否比法则3方便呢? 答复是:当我们只计算一、二个含酶物浓度时,还是用法则3来计算较为方便。因为虽然在这两种情况下1-因子的数目可能相差不大,但用法则2每算一个 $N_i (i = 1, 2, \dots)$ 都得画一个变换图 $\mathcal{D}_i^*$, 而用法则3,在指定了一个参考点 E_m 后,只需画一个变换图 $\mathcal{D}_m^*$ 就可以了。

最后,应当指出:与酶动力学中现有的图形方法一样,本文的方法也仅适用于定态反应体系;在下一文^[7]中,我们将把图形方法推广到非定态反应体系。

附录 I. 本文所涉及到的图论术语

由于图论是从许多不同的学科抽象提高而形成的一个数学分支,它的术语至今尚未统一,故我们有必要介绍一下正文中所涉及到的图论术语的含义。

有向图、点、弧、圈 一个有向图是由有限个点和有限条弧组成的,其中每条弧按照一定的指向连结两个点。例如, $x = \overrightarrow{UV}$ 表示从点 U 到点 V 的弧: U 称为该弧的起点, V 为终点。起点和终点相同的弧称为圈。

赋权有向图 对有向图中的每条弧都赋予一个权的有向图称为赋权有向图。根据各种具体的研究对象,赋权有向图中的点和权可以有不同的含义。在本文中,所谓点是代表含酶物,权是代表反应速度常数。在赋权有向图中,权为0的弧不必画出。这样,就可用一个赋权有向图来表示某个酶反应体系(参见正文中的图1、图2、...等)。

途径、路、回路 对于有向图中点与弧的交替序列 $V_1, x_1, V_2, \dots, V_{m-1}, x_{m-1}, V_m$, 若其中每条弧 $x_i = \overrightarrow{V_i V_{i+1}} (i = 1, \dots, m-1)$ 时,则称之为从点 V_1 到点 V_m 的途径。若途径中各点互异,则称为路。若途径中除了 $V_1 = V_m$ 外,其余各点互异,则称为回路。

半途径、半路、半回路 在上述定义中,若把条件 $x_i = \overrightarrow{V_i V_{i+1}}$ 改为 $x_i = \overleftarrow{V_i V_{i+1}}$ 或 $x_i = \overleftarrow{V_{i+1} V_i}$, 就分别相应得到半途径、半路、半回路。显然,途径、路、回路分别是特殊的半途径、半路、半回路。

汇、入树 若有向图中所有各点都至少存在一条路可以到达某点 U , 则点 U 就称为这个有向图的汇。一个有汇而没有半回路的有向图称为入树。

子图、支撑子图 由有向图 $\mathcal{D}$ 中的一部分点和一部分弧所组成的有向图 $\mathcal{D}_1$ 称为 $\mathcal{D}$ 的子图。若 $\mathcal{D}_1$ 的子图 $\mathcal{D}_2$ 包含 $\mathcal{D}$ 中所有的点,则 $\mathcal{D}_1$ 就称为 $\mathcal{D}$ 的支撑子图。

支撑入树、1-因子 若 $\mathcal{D}$ 的支撑子图 $\mathcal{D}_1$ 也是入树,则称 $\mathcal{D}_1$ 为 $\mathcal{D}$ 的支撑入树。若 $\mathcal{D}$ 的支撑子图 $\mathcal{D}_1$ 是由互不相交的回路所组成的,则称 $\mathcal{D}_1$ 为 $\mathcal{D}$ 的1-因子。

附录 II. 法则 1 的数学原理

正文中已述及, N_m 中项的数目 P_m 实际上就等于相应向图 $\mathscr{D}$ 中以 E_m 为汇的支撑入树的数目. 而经法则 1 中的步骤(1)–(2)所得到的矩阵 B 又可以表示成:

$$B = A^* - A$$

式中 A^* 是一个对角矩阵, 对角元 a_{ii}^* 就等于矩阵 A 的第 i 行元素之和. Tutte 首先证明了, 在有向图 $\mathscr{D}$ 中以点 E_m 为汇的支撑入树的数目就等于矩阵 $(A^* - A)$ 的第 m 行中任一元素的代数余子式^[6]. 这就是法则 1 的数学原理.

附带指出, 法则 1 也可用来计算含多重弧的有向图 $\mathscr{D}$ 中的支撑入树的数目. 不过在这种情况下, 矩阵 A 的元素 a_{ij} 应等于有向图 $\mathscr{D}$ 中从点 E_i 到点 E_j 的弧的数目. 但在实际问题中, 我们总可将多重弧合并成一条弧来处理^[2], 故通常无需考虑多重弧的情况.

附录 III. 法则 2 和法则 3 的数学原理

首先指出, 某个 n 阶方阵 $X = [x_{ij}]$ 可以用一个含 n 个点 $(V_1, V_2, \dots, V_n)$ 的赋权有向图 $\mathscr{D}(X)$ 来表示. 表示的方法是: 若 $x_{ij} \neq 0$, 就画出从点 V_i 到点 V_j 的弧 $\widehat{V_i V_j}$ 并赋之以权 x_{ij} ; 若 $x_{ij} = 0$, 则不必画出从点 V_i 到点 V_j 的弧. 这样, 赋权有向图和方阵之间就建立了一一对应的关系.

现在可以看出, 经法则 2 和法则 3 的步骤(1)–(3)而得到的赋权有向图所对应的矩阵的行列式, 正好分别是(4)式的 N_m 和(5)式的 D .

可以证明: 对于任意方阵 $X = [x_{ij}]$, 我们有

$$\det X = \sum_{\tau} (-1)^{q_{\tau}} f(h_{\tau}),$$

式中 h_{τ} 是矩阵 X 所对应的赋权有向图 $\mathscr{D}(X)$ 中的第 τ 个 1-因子, q_{τ} 是该 1-因子中含偶数条弧的回路数目, $f(h_{\tau})$ 代表 h_{τ} 中所有弧的权之积.

证明方法如下. 如所周知,

$$\det X = \sum_{\sigma} \operatorname{sgn} \sigma x_{i_1 j_1} x_{i_2 j_2} \cdots x_{i_n j_n}.$$

如果 $x_{i_1 j_1} x_{i_2 j_2} \cdots x_{i_n j_n} \neq 0$, 则它显然就等于有向图 $\mathscr{D}(X)$ 中由弧 $\widehat{V_{i_1} V_{j_1}}, \widehat{V_{i_2} V_{j_2}}, \dots, \widehat{V_{i_n} V_{j_n}}$ 所组成的子图的权之积. 在这些弧中, 每个指定的下标正好都出现两次: 一次出现在弧的起点, 另一次出现在弧的终点. 故这个子图必定是 $\mathscr{D}(X)$ 中的一个 1-因子. 反之, 对于 $\mathscr{D}(X)$ 中的每个 1-因子, 在 $\det X$ 中也必定有一个非 0 项与之有这样的对应关系. 换言之, $\det X$ 中的非 0 项与 $\mathscr{D}(X)$ 中的 1-因子是一一对应的. 即我们有

$$\det X = \sum_{\tau} \operatorname{sgn} \sigma_{\tau} f(h_{\tau}),$$

式中 $\operatorname{sgn} \sigma_{\tau}$ 是 $\det X$ 中与第 τ 个 1-因子 h_{τ} 相对应的项的符号. 现在让我们来看看 $\operatorname{sgn} \sigma$ 与 1-因子的关系. 我们知道, 每个 1-因子都是由若干互不相交的回路所组成的. 设 $V_{i_1}, \widehat{V_{i_1} V_{j_1}}, V_{i_2}, \dots, \widehat{V_{i_l} V_{j_l}}, V_{i_l}$ 是第 τ 个 1-因子 h_{τ} 中的一个回路, 它含有 l 条弧. 按照置换 σ 中逆序个数与符号的关系, 显而易见, 这个回路所对应的项 $x_{i_1 j_1} x_{i_2 j_2} \cdots x_{i_l j_l}$ 对 $\operatorname{sgn} \sigma_{\tau}$ 将贡献一个 $(-1)^{l-1}$ 的符号因子. 这样, 若将 h_{τ} 中所有的回路都考虑进去, 使得

$$\operatorname{sgn} \sigma_{\tau} = \prod_{\left(\begin{array}{l} l = h_{\tau} \text{ 中每个回} \\ \text{路的弧的数目} \end{array} \right)} (-1)^{l-1}.$$

而

$$(-1)^{l-1} = \begin{cases} -1, & \text{当 } l \text{ 为偶数时,} \\ 1, & \text{当 } l \text{ 为奇数时,} \end{cases}$$

故有

$$\operatorname{sgn} \sigma_r = (-1)^{\{h_r \text{ 中含偶数条弧的回路数目的}\}} = (-1)^{q_r}.$$

以上就是法则2的(11)式和法则3的(12)式的数学原理。

附录 IV. 矩阵的承袭式及其基本性质

设 $X = [x_{ij}]$ 为 n 阶方阵, 它的承袭式定义为

$$\operatorname{per} X = \sum_{\sigma} x_{1j_1} x_{2j_2} \cdots x_{nj_n},$$

式中 $\sigma = \{j_1, j_2, \dots, j_n\}$, 它是 $\{1, 2, \dots, n\}$ 这 n 个数的置换, $\sum_{\sigma}$ 表示对所有可能的置换求和, 由此可知, 承袭式的项与行列式的项是一样的, 也是从方阵的每一行和每一列中各取一个、并只取一个元素组成的 n 个元素的乘积, 所不同的只是不必附加符号 $\operatorname{sgn} \sigma$.

现若 $\mathcal{O}(X)$ 是方阵 $X = [x_{ij}]$ 所对应的有向图, 则根据承袭式的定义和矩阵 $A(\mathcal{O})$ 的结构(参见正文的(8)式), 我们不难看出:

$$\operatorname{per} A(\mathcal{O}) = \det X \text{ 的展开式中非 } 0 \text{ 项的数目.}$$

另一方面, 由附录 III 知

$$\det X \text{ 的展开式中非 } 0 \text{ 项的数目} = \mathcal{O}(X) \text{ 中 } 1\text{-因子的数目, 故得}$$

$$\operatorname{per} A(\mathcal{O}) = \mathcal{O}(X) \text{ 中 } 1\text{-因子的数目.}$$

这就是法则2的(6)和法则3的(5)的数学原理。

下面, 我们给出承袭式的一些性质:

性质1 矩阵 X 经转置后, 其承袭式的值不变, 即

$$\operatorname{per} X^T = \operatorname{per} X.$$

性质2 交换矩阵中的任意两行(或两列), 承袭式的值不变。

性质3 承袭式可按它的任一行(或列)展开; 例如

$$\begin{aligned} \operatorname{per} \begin{bmatrix} x_{11} & x_{12} \cdots x_{1n} \\ x_{21} & x_{22} \cdots x_{2n} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \cdots x_{nn} \end{bmatrix} &= x_{11} \operatorname{per} \begin{bmatrix} x_{22} & \cdots & x_{2n} \\ \vdots & & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} + \\ &x_{12} \operatorname{per} \begin{bmatrix} x_{21} & x_{23} \cdots x_{2n} \\ \vdots & \vdots \\ x_{n1} & x_{n3} \cdots x_{nn} \end{bmatrix} + \cdots + x_{1n} \operatorname{per} \begin{bmatrix} x_{21} \cdots x_{2(n-1)} \\ \vdots \\ x_{n1} \cdots x_{n(n-1)} \end{bmatrix}. \end{aligned}$$

推论 某一行(或列)全为0的矩阵的承袭式值为0。

性质4 设 J 是所有元素都为1的 n 阶方阵, 则

$$\operatorname{per} J = n!.$$

推论1 设 $J_n^{(1)}$ 是只有一个元素为0、其余元素都为1的 n 阶方阵, 则

$$\operatorname{per} J_n^{(1)} = n! - (n-1)! = (n-1) [(n-1)!].$$

推论2 设 $J_n^{(k)}$ 是有 k 个独立的0(即任两个0都不在同一行或同一列中)、其余元素都为1的 n 阶方阵 ($k \leq n$), 则

$$\operatorname{per} J_n^{(k)} = n! - \binom{k}{1} (n-1)! + \binom{k}{2} (n-2)! - \cdots + (-1)^k \binom{k}{k} (n-k)!.$$

注意 性质 2 是与行列式的性质不同的。所以计算承袭式的值时, 一般不能像计算行列式那样把某一行(或列)的倍数加到另一行(或列)上去。

参 考 文 献

- [1] King, E. L. & Altman, C., *J. Phys. Chem.*, **60** (1956), 1375—1378.
- [2] Volkenstein, M. V. & Goldstein, B. N., *Biochem. Biophys. Acta.*, **115** (1966), 471—477.
- [3] Fromm, H. J., *Biochem. Biophys. Res. Commun.*, **40** (1970), 692—697.
- [4] Cornish-Bowden, A., *Principles of Enzyme Kinetics* 1976, 34—51.
- [5] Cleland, W. W., *Biochem. Biophys. Acta.*, **67** (1963), 104—137.
- [6] Harary, F., *Graph Theory*, Addison-Wesley, 1969, 203—204.
- [7] 周国城、刘为民, 酶动力学的图形理论——瞬变态反应体系(待发表)