

文章编号: 0258-2724(2004)05-0565-04

求简单有向图所有基本回路的强核图论算法

赵禹骅¹, 李可柏², 任伟民¹

(1. 同济大学经济与管理学院, 上海 200092; 2. 南昌大学管理科学与工程系, 江西 南昌 330047)

摘要: 求系统动力学模型的所有反馈环等价于求对应的简单有向图的所有基本回路, 其核心问题是算法的时间复杂度. 针对这一问题, 提出强核的概念, 基于强核概念设计了求简单有向图所有基本回路的算法, 给出相应算例, 并分析了算法复杂性. 在时间复杂度上, 本算法优于基于核概念的有向图的行列式算法.

关键词: 系统动力学; 图论; 反馈环; 基本回路

中图分类号: N941.3; O157.6 **文献标识码:** A

Strong Kernel Graphic Algorithm for Searching All Essential Circuits of Simple Directed Graph

ZHAO Yu-hua¹, LI Ke-bai², REN Wei-min¹

(1. School of Economics and Management, Tongji University, Shanghai 200092, China; 2. Dept. of Management Science and Eng., Nanchang University, Nanchang 330047, China)

Abstract: The problem of searching all feedback loops of a system dynamics model is equal to calculating all essential circuits of a corresponding simple directed graph. The key of the problem is time complexity. A new concept named strong kernel was defined, and the algorithm based on strong kernel for searching all essential circuits of a simple directed graph was proposed. An illustrative example was presented and the complexity of the algorithm was analyzed. In terms of time complexity, the proposed algorithm is superior to that of the determinant algorithm based on the concept of kernel of directed graphs.

Key words: system dynamics; graph theory; feedback loop; essential circuit

系统动力学(system dynamics, SD)是系统科学的一个重要分支, 是一门主要通过建立流位、流率系统来研究复杂反馈系统的科学^[1, 2]. 在SD的研究中, 反馈环分析处于基础性地位, 找到系统的全部反馈环, 才能更好地进行系统的主导反馈环分析和功能-结构分析^[3, 4]. 根据SD中反馈环的定义, 找到系统全部反馈环的问题等价于求模型对应的简单有向图 G 的所有基本回路, 这是图论中最困难的问题之一. 求有向图的Hamilton回路问题是求SD模型所有反馈环问题的子问题, 本身已是NP完全问题^[5], 至今未找到多项式算法, 且其多项式算法的存在性尚不能确定^[6]. 如用穷举法, 计算 N 个顶点的问题, 其时间复杂度为 $O(N!)$, 这就是所谓阶乘的“组合爆炸”^[7, 8]. 求解SD模型所有反馈环问题更加困难, 其时间复杂度为 $O\left[\sum_{k=2}^N (C_k^k - 1)\right]$. 因此, 如何尽可能地减少算法的时间复杂度, 成为解决这个问题的关键. 文献[9, 10]运用图论中核的概念, 将图约简为只有 n ($n \leq N$)个顶点的核, 并设计了有向图的行列式算法, 使算法的时间复杂度降至 $O(N!)$. 这种几何代数化的思想常用于求解图论中的问题^[11, 12], 有利于算法的实现. 但对原图的约简仍是降低算法时间复杂度的关键. 对原图的约简越充分, 其后的算法时间复杂度就越小. 约简图形

收稿日期: 2003-02-17

作者简介: 赵禹骅(1965—), 男, 经济师, 博士研究生.

的一种常用方法是图形分解^[13], 其数学原理在于, 当 $n = n_1 + n_2$ 且 $n_1 \geq 2, n_2 \geq 2$ 时, 有 $(n_1!) + (n_2!) < n!$ 和 $(n_1!) \times (n_2!) < n!$.

笔者运用图论知识, 提出强核的概念, 以强核为工具, 运用分裂原理分解并约简 SD 模型对应的有向图, 并设计基于强核概念的求解 SD 模型所有反馈环问题的算法, 力图进一步减小算法的时间复杂度.

1 基本概念及性质

设图 $G = (V, E)$ 是一简单有向图.

定义 1^[7] 图 G 是强连通的, 指的是对于它的任意 2 个顶点 i 和 j , 存在一条从 i 到 j 的有向道路和一条从 j 到 i 的有向道路.

定义 2^[7] 图 G 的极大强连通子图叫做强连通块.

性质 1^[7] 图 G 的强连通块用一个点表示, 得点的集合 V . 若 i 所代表的强连通块的点到 j 所代表的强连通块的点, 有边属于 E , 则构造有向边 $(i, j) \in E$, 得图 $G = (V, E)$. 称 G 为 G 的构成图, 则 G 是无回路的有向图.

由性质 1, 易知以下性质:

性质 2 图 G 的基本回路只能存在于各个强连通块内部, 而不可能存在于 2 个或 2 个以上的强连通块之间.

定义 3 强核. 在图 G 的任一顶点数 $N_i \geq 3$ 的强连通块 G_i 中, 对所有入度或出度为 1 的顶点进行递归收缩, 过程如下:

(1) 初始化. 置字符数组 $A = \emptyset, B = \emptyset$; 数组 $C_{ij} = 0, D_{ij} = 0$. A 用于记录可收缩弧的字符串; B 记录收缩的基本回路的字符串; C_{ij} 标记弧 (i, j) 是否已经加入 A 中, 未加入时, $C_{ij} = 0$, 加入后, $C_{ij} = 1$; D_i 标记弧 (i, j) 是否为收缩弧, 不是收缩弧置 $D_i = 0$, 是收缩弧记 $D(i, j) = 1; i, j = 1, \dots, N_i$.

(2) 如 G_i 为空, 转(8); 如 G_i 所有顶点的入度 d^- 和出度 d^+ 均大于 1, 转(7), 否则任取 G_i 中一个入度或出度为 1 的点 j , 设 i 和 k 为 j 的所有邻接顶点, 存在弧 (i, j) 和 (j, k) .

(3) 检查所有 (i, j) 和 (j, k) . 当道路 $(i, j, k) \notin A$ 时, 如 $D_{ij} = 1$ 或 $D_{jk} = 1$, 则 A 含 (i, \dots, j) 或 (j, \dots, k) , 将 (i, j, k) 按 (i, \dots, j) 和 (j, \dots, k) 的组合逐一展开后加入 A ; 否则, 将 (i, j, k) 直接加入 A .

(4) 如 $(i, k) \in G_i$ 且 $C_{ik} = 0$, 将 (i, k) 直接加入 A , 令 $C_{ik} = 1$.

(5) 如 $i = k$, 则 (i, j, k) 为 G_i 的一条收缩的基本回路, 将 (i, j, k) 加入 B .

(6) 令 $D_{ik} = 1$, 删除道路 (i, j, k) . 如 j 为孤立点, 删除 j . 如 $i \neq k$ 且 $(i, k) \notin G_i$, 在 G_i 中添加弧 (i, k) . 如 $i = k$, 且为孤立点, 删除 i . 转(2).

(7) 用穷举法求出 G_i 的所有收缩的基本回路, 加入 B .

(8) 将 B 中所有收缩的回路中 $D_{ij} = 1$ 的弧 (i, j) 按 A 中的相应组合逐一展开, 得 G_i 的所有基本回路. 结束.

记剩下的图为 G'_i , 称 G'_i 为 G_i 的强核.

由 G'_i 的生成过程, 可得以下性质:

性质 3 G'_i 或为空, 或其所有顶点的入度和出度均 ≥ 2 , 且顶点数 ≥ 3 .

性质 4 数组 B 按数组 A 展开后得到的基本回路与 G_i 的所有基本回路一一对应.

2 算法的过程描述

设 SD 模型对应顶点数为 N 的简单有向图 G .

(1) 求 G 的所有强连通块 $G_i, i = 1, \dots, m, m \leq N$. 具体方法见文献[7].

(2) 删去全部顶点数为 1 的 G_i ; 对每一顶点数为 2 的 G_i , 直接求其基本回路.

(3) 对每一顶点数 ≥ 3 的 G_i , 按定义 3 的过程求其所有基本回路. 当 G'_i 非空时, 设其顶点数为 $n(i)$, 用穷举法求收缩的基本回路. 算法如下:

① 令 $j = 2$.

- ② 按组合生成算法^[9], 从 G'_i 中任选 j 个顶点, 按序号由小到大排列, 检查这 j 个顶点是否存在按此序号排列的收缩的基本回路. 如存在, 将其加入 B .
- ③ 固定这 j 个顶点中序号最小的点, 按排列生成算法^[6], 对其余 $j-1$ 个顶点做排列生成运算, 对每一种排列, 检查是否存在按此排列数排列的收缩的基本回路. 如存在, 将其加入 B .
- ④ 令 $j=j+1$, 如 $j \leq n(i)$, 转②.
- (4) 以上求出的所有基本回路即为 G 的所有基本回路, 即为原 SD 模型的所有反馈环. 结束.

3 算 例

已知 SD 模型的结构流图对应有向图 G , 如图 1 所示, 求该模型的所有反馈环.

(1) 求 G 的所有强连通块 $G_i(i=1, \cdots, 6)$, 如图 2 所示.

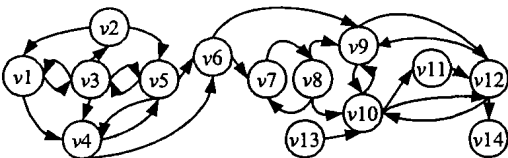


图 1 有向图 G
Fig. 1 Directed graph G

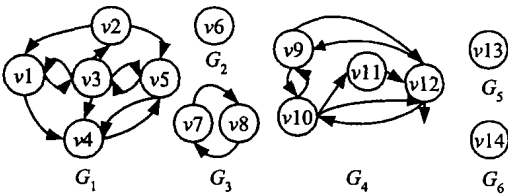


图 2 有向图 G 的所有强连通块 $G_i(i=1, \cdots, 6)$
Fig. 2 All strong connected components $G_i(i=1, \cdots, 6)$ on directed graph G

(2) G_2, G_5 和 G_6 的顶点数均为 1, 不含基本回路. G_3 的顶点数为 2, 其基本回路为(7, 8, 7).

(3) 求 G_1 和 G_4 的基本回路.

① 求 G_1 的强核 G'_1 及基本回路, 见图 3.

$d^+(2)=1$, 约简得图 G_1^1 , 记下可收缩弧序列(3, 1), (3, 2, 1), (3, 5), (3, 2, 5).

$d^-(4)=1$, 约简得图 G_1^2 , 记下可收缩弧序列(1, 4, 5), (3, 4, 5); 记下收缩的基本回路(5, 4, 5).

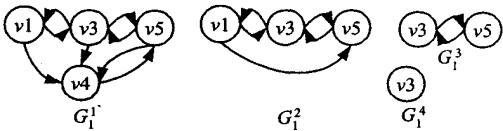


图 3 G_1 的约简步骤

Fig. 3 The reducing procedure on G_1

$d^-(1)=1$, 约简得图 G_1^3 , 展开弧(3, 1, 5), 并记下可收缩弧序列(3, 1, 4, 5), (3, 2, 1, 4, 5); 记下收缩的基本回路(3, 1, 3).

$d^+(5)=1$, 约简得图 G_1^4 , 记下收缩的基本回路(3, 5, 3), 3 为孤立点, 删除, 得强核 G'_1 为空. 约简结束, 有如下结果:

可收缩弧序列为(3, 1), (3, 2, 1), (3, 5), (3, 2, 5), (1, 4, 5), (3, 4, 5), (3, 1, 4, 5), (3, 2, 1, 4, 5); 收缩的基本回路为(5, 4, 5), (3, 1, 3), (3, 5, 3).

从可收缩弧序列展开收缩的基本回路, 得 G_1 的所有基本回路: (5, 4, 5), (3, 1, 3), (3, 2, 1, 3), (3, 5, 3), (3, 2, 5, 3), (3, 4, 5, 3), (3, 1, 4, 5, 3), (3, 2, 1, 4, 5, 3).

② 求 G_4 的强核 G'_4 及基本回路.

$d^+(11)=1$, 约简得图 4, 记下可收缩弧序列(10, 12), (10, 11, 12). 不可再约简, 得强核 G'_4 . 用穷举法求得 G'_4 的所有收缩的基本回路: (9, 10, 9), (9, 12, 9), (10, 12, 10), (9, 10, 12, 9), (9, 12, 10, 9).

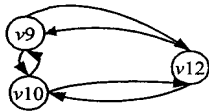


图 4 G_4 的约简结果

Fig. 4 The reduced result of G_4

可收缩弧序列为(10, 12), (10, 11, 12). 展开得 G_4 的所有基本回路: (9, 10, 9), (9, 12, 9), (10, 12, 10), (10, 11, 12, 10), (9, 10, 12, 9), (9, 10, 11, 12, 9), (9, 12, 10, 9).

(4) G 的所有基本回路即 SD 模型的所有反馈环为: (7, 8, 7); (5, 4, 5), (3, 1, 3), (3, 2, 1, 3), (3, 5, 3), (3, 2, 5, 3), (3, 4, 5, 3), (3, 1, 4, 5, 3), (3, 2, 1, 4, 5, 3); (9, 10, 9), (9, 12, 9), (10, 12, 10), (10, 11, 12, 10), (9, 10, 12, 9), (9, 10, 11, 12, 9), (9, 12, 10, 9).

4 复杂性分析

算法的(1)、(2)步均可在多项式时间内完成. 算法第(3)步生成数组 A 和将数组 B 展开的时间复杂度较难计算. 当 A 含 i 个元素时, 如欲加入第 $i+1$ 个元素, 设其含 $j(j \leq 2)$ 条可收缩弧, 则第 $i+1$ 个元素最多可扩充到的个数为 $(i/j)^j$, 这不是多项式算法. 对每个非空的 G'_i , 算法第(3)步的组合生成算法的运算量为 $\sum_{j=2}^{n(i)} C_{n(i)}^j$; 对每个 j , 其排列生成算法的运算量为 $(j-1)!$ 因此每个非空的 G'_i 的运算量为

$$\sum_{j=2}^{n(i)} [C_{n(i)}^j (j-1)!],$$

总的运算量为

$$\sum_{i=1}^m \{ \sum_{j=2}^{n(i)} [C_{n(i)}^j (j-1)!] \}.$$

本算法的时间复杂度为

$$O(\sum_{i=1}^m \{ \sum_{j=2}^{n(i)} [C_{n(i)}^j (j-1)!] \}),$$

优于文献[10]中的行列式算法的时间复杂度 $O(n!)$, 其中 n 是原图的核的顶点数, 易知 $n \geq \sum_{i=1}^m n(i)$. 若以本算法的运算量为分子, 以文献[10]中算法的运算量为分母, 并求极限, 有

$$\lim_{n \rightarrow \infty} \frac{\sum_{i=1}^m \sum_{j=2}^{n(i)} [C_{n(i)}^j (j-1)!]}{n!} \leq \lim_{n \rightarrow \infty} \frac{\sum_{j=2}^n [C_n^j (j-1)!]}{n!} = 0.$$

可见, 本算法的时间复杂度小于文献[10]中算法的时间复杂度, 且随 n 增大, 本算法的优势更加明显.

5 结 论

提出了简单有向图的强核概念, 并设计了求简单有向图的全部基本回路的算法. 本算法基于强核概念, 采用分而治之的方法, 运用分裂原理, 将高阶问题低阶化, 降低了问题的复杂度, 并运用排列生成算法和组合生成算法进一步简化计算, 提高了算法的效率. 本算法的时间复杂度为

$$O(\sum_{i=1}^m \{ \sum_{j=2}^{n(i)} [C_{n(i)}^j (j-1)!] \}),$$

与文献[10]相比降低了时间代价.

参考文献:

[1] 贾仁安, 胡玲, 丁荣华等. SD 简化流率基本入树模型及其应用[J]. 系统工程理论与实践, 2001, 21(10): 137-144.

[2] 胡玲, 贾仁安. 强简化流率基本入树模型与枝向量矩阵反馈环分析法[J]. 系统工程理论与实践, 2001, (11): 83-88.

[3] 贾仁安, 丁荣华. 系统动力学——反馈动态性复杂分析[M]. 北京: 高等教育出版社, 2002. 65.

[4] 王其藩. 高级系统动力学[M]. 北京: 清华大学出版社, 1995. 94-95.

[5] Anany Levitin. 算法设计与分析基础(影印版)[M]. 北京: 清华大学出版社, 2003. 351-353.

[6] 卢开澄, 卢华明. 组合数学(第三版)[M]. 北京: 清华大学出版社, 2002. 484, 23-31.

[7] 卢开澄. 计算机算法导引——设计与分析[M]. 北京: 清华大学出版社, 2001. IX, 193-199.

[8] 杨传杰, 李继红. C 语言编程实践及实例荟萃[M]. 安徽: 安徽科学技术出版社, 1995. 277-278.

[9] 徐兵, 贾仁安. 有向圈的矩阵算法及其有关性质[J]. 南昌大学学报(理科版), 2002, 26(1): 5-11.

[10] 徐兵, 贾仁安. 有向圈的行列式算法及 HAMILTON 图条件[J]. 数学的实践与认识, 2002, 32(4): 643-650.

[11] 程世娟, 何平. 一种计算网络两点间最小路集的方法[J]. 西南交通大学学报, 2002, 37(1): 95-98.

[12] 张羽成, 吕红霞, 王宝杰. 基于图论的列车运行径路分析与构造[J]. 西南交通大学学报, 2000, 35(3): 273-276.

[13] Zhu S N, Zhu Q. Network Decomposition and Maximum Independent Set[J]. Journal of Southwest Jiaotong University, 2003, 11(2): 103-121.