

电力系统充裕度分析中的快速组合事件概率算法

徐林, 王秀丽, 王锡凡, 滕予非

(西安交通大学电气工程学院, 710049, 西安)

摘要: 提出了一种可应用于电力系统充裕度分析的快速组合事件概率求取方法. 该方法根据二值决策图的基本原理, 采用二叉树作为基本数据结构, 在电力系统充裕度计算所有可行状态已知的情况下, 只需一次回代即可求出相应的系统充裕度指标, 具有原理清晰、算法简单、易于实现的特点. 理论分析和实际算例表明, 其时间和空间复杂度都是问题规模的线性函数, 有着比经典 inclusion-exclusion 法则更好的计算效率, IEEE RTS-24 节点系统算例的计算表明了该算法在实际应用中的正确性和有效性.

关键词: 电力系统充裕度; 组合事件概率; 二叉树; 二值决策图

中图分类号: TM732 **文献标志码:** A **文章编号:** 0253-987X(2008)06-0713-05

Fast Composite-Event Probability Algorithm in Power System Adequacy Analysis

XU Lin, WANG Xiuli, WANG Xifan, TENG Yufei

(School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Fast composite-event probability algorithm with both linear time and space complexity is proposed. Based on the binary decision diagrams, the binary tree serves as the essential data structure in the presented algorithm, where the system adequacy index is obtained simply once if all the valid system states are known. The complexity of the presented algorithm depends linearly on the problem scale, which is endowed with a better performance than the classical inclusion-exclusion principle. A numerical example of IEEE RTS-24 test system confirms the validity.

Keywords: power system adequacy; composite-event probability; binary tree; binary decision diagram

随着电力系统规模的扩大和电力市场的引入, 电力系统可靠性越来越引起人们的关注^[1]. 基于充裕度的分析方法, 需要由已知的可行状态形成相应的系统指标, 实质是进行大规模组合事件概率计算. 因此, 如何提高其效率, 是充裕度分析应用于大规模系统的重要前提之一, 也是本文所研究的内容.

常见的组合事件概率算法有以下几种^[2~4]: 穷举法使用 inclusion-exclusion 法则^[5] 直接对所有事件进行概率求和计算, 计算量较大, 不适用于大规模系统; 概率图法将事件状态表示在一个表格内, 可对具有相同项的事件进行合并; 全概率分解法可对概率计算公式进行分解, 降低问题规模, 在复杂系统中常被应用; 最小路/最小割集法是最常用的组合事件概

率算法, 通过找出关键元件的组合, 能减少事件枚举的状态; 不变化布尔代数法通过不变化处理, 可以把组合事件概率的计算化为各事件概率简单相加的形式; 网络变换法则通过将复杂网络简化为简单的串并联网络来简化计算.

二值决策图^[6] (BDD) 是一种表示布尔代数的有效方法, 在系统设计和可靠性分析方面得到了广泛的应用^[7~8]. 本文根据其基本思想, 提出了一种基于二叉树的组合事件概率算法, 可以用于电力系统充裕度指标形成中组合事件概率的计算. 在可行状态已知的前提下, 该方法只需一次回代即可形成相应的系统指标. 理论分析和具体算例表明了其时间和空间复杂度都是问题规模的线性函数, 具有较好的

运算效率,可望在大规模电力系统可靠性分析中得以应用。

1 原理与流程

1.1 充裕度分析中系统指标的形成

假设系统中每个元件都是两状态元件,并有

$$a_i = \begin{cases} 1 & \text{元件 } i \text{ 运行} \\ 0 & \text{元件 } i \text{ 停运} \end{cases} \quad i = 1, 2, \dots, s \quad (1)$$

式中: a_i 为第 i 个元件所对应的状态; s 为系统元件总数。则系统的一个状态可记为

$$A = a_s a_{s-1} \cdots a_3 a_2 a_1 \quad (2)$$

设 $\Phi(s)$ 为系统的状态空间, 对其中的一个子集 $\phi \subseteq \Phi(s)$, 所有状态都满足某一指标 D , 则全系统满足指标 D 的概率为 ϕ 中所有状态的概率和, 即

$$R_D = P(\bigcup A_i) \quad \forall A_i \in \phi \subseteq \Phi \quad (3)$$

对于式(3)常用 inclusion-exclusion 法则来计算, 即

$$\begin{aligned} P(\bigcup_{i=1}^m A_i) &= \sum_{i=1}^m P(A_i) = \sum_{i=1}^m \sum_{j>i}^m P(A_i A_j) + \\ &\quad \sum_{i=1}^m \sum_{j>i}^m \sum_{k>j}^m P(A_i A_j A_k) + \cdots - \\ &\quad (-1)^m P(A_1 A_2 A_3 \cdots A_{m-1} A_m) \end{aligned} \quad (4)$$

式中: m 为满足指标 D 的状态总数。

式(4)应用于大规模系统常会遇到维数灾问题, 很难直接应用, 因此需寻找更为有效的方法。

1.2 系统状态树的形成

ϕ 中元素互斥时, 式(4)可化简为简单求和形式, 即

$$R_D = P(\bigcup_{i=1}^m A_i) = \sum_{i=1}^m P(A_i) \quad (5)$$

这时时间复杂度降为 $O(ms)$, 为了做进一步简化, 采用二叉树对系统状态进行存储。存放原则如下:

- (1) 二叉树第 i 层代表系统第 i 个元件的状态;
- (2) 每层左节点表示状态 0, 右节点表示状态 1。

按上述原则对系统状态的编码串进行存放的二叉树称为状态树, 状态树的具体形成步骤为:

(1) 进入状态树第 0 层(第 0 层为虚拟元件, 对应树的根节点, 永远都只有一个状态);

(2) 对状态树第 i 层, 如果元件 $i+1$ 的状态为 0/1, 则判断该层是否已经存在左/右子节点, 如果是, 按照原则(2), 转到步骤(3), 否则转到步骤(4);

(3) 沿左/右子节点移动到下一层;

(4) 按原则(2)生成新的左/右子节点, 并沿该子节点移动到下层, 这时所插入的是一个新状态;

(5) 继续步骤(2), 直到检查完所有元件为止。

图 1 为 3 元件系统状态树的形成过程, 假设已存在系统状态(0,0,1), 欲插入一个新状态(0,1,1), 系统状态树的形成流程如图 2 所示。

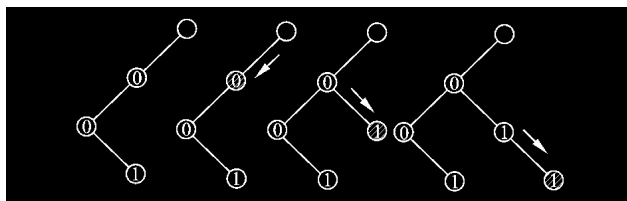


图 1 系统状态树形成过程示意图

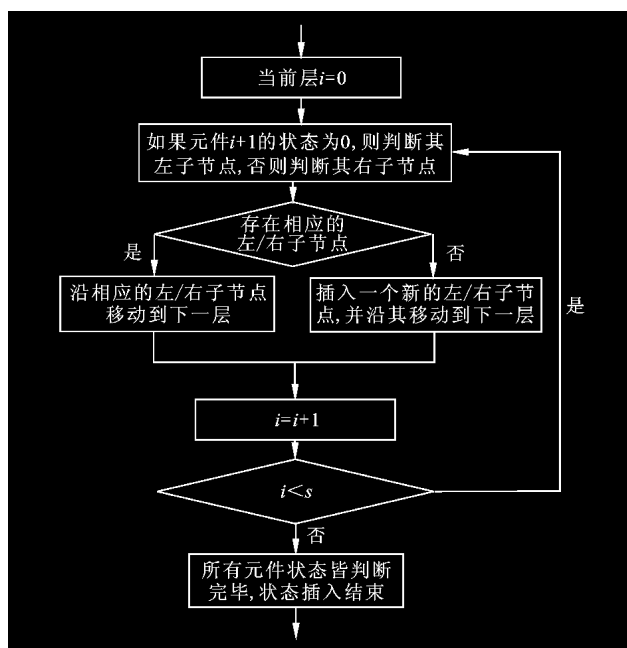


图 2 状态树的生成算法流程

1.3 组合事件概率的计算

假设状态树中每个节点都对应一个累加概率 P , 初值为

$$P = \begin{cases} 0 & \text{该节点非叶子节点} \\ 1 & \text{该节点为叶子节点} \end{cases} \quad (6)$$

按如下步骤形成系统指标:

- (1) 由叶子节点, 逐层向上计算;
- (2) 对第 i 层所有节点, 若为左节点, 将其累加概率乘以 a_i 的停运概率, 反之则乘以 a_i 的运行概率, 并将结果累加到其父节点的累加概率上去;
- (3) 重复步骤(2), 直到算到虚拟的根节点为止;
- (4) 这时根节点的累加概率即为状态树中所有可行状态所对应的组合概率。

设 p_i 为元件 i 的运行概率, $\bar{p}_i$ 为停运概率, 并用 x_i 表示其对应的概率值, 即

$$x_i = \begin{cases} p_i & \text{该元件运行} \\ \bar{p}_i & \text{该元件停运} \end{cases} \quad (7)$$

则式(5)可写为

$$R_D = \sum_{i=1}^m x_{i1} x_{i2} \cdots x_{is} \quad (8)$$

对该求和形式,可将与 $x_1 = p_1$ 相乘的项和与 $x_1 = \bar{p}_1$ 相乘的项分开,即

$$R_D = p_1 \sum_{i=1}^{m_1} x_{i2} \cdots x_{is} + \bar{p}_1 \sum_{i=1}^{m_2} x_{i2} \cdots x_{is} \quad (9)$$

式中: m_1 表示含 p_1 项的个数; m_2 表示含 $\bar{p}_1$ 项的个数. 将式(9)的两个求和项中分别和 $x_2 = p_2$ 及 $x_2 = \bar{p}_2$ 相乘的项分开,重复该过程,直到所有的元件全被分开为止. 将这一过程反演,即为图3所示过程,因此两者是等效的.

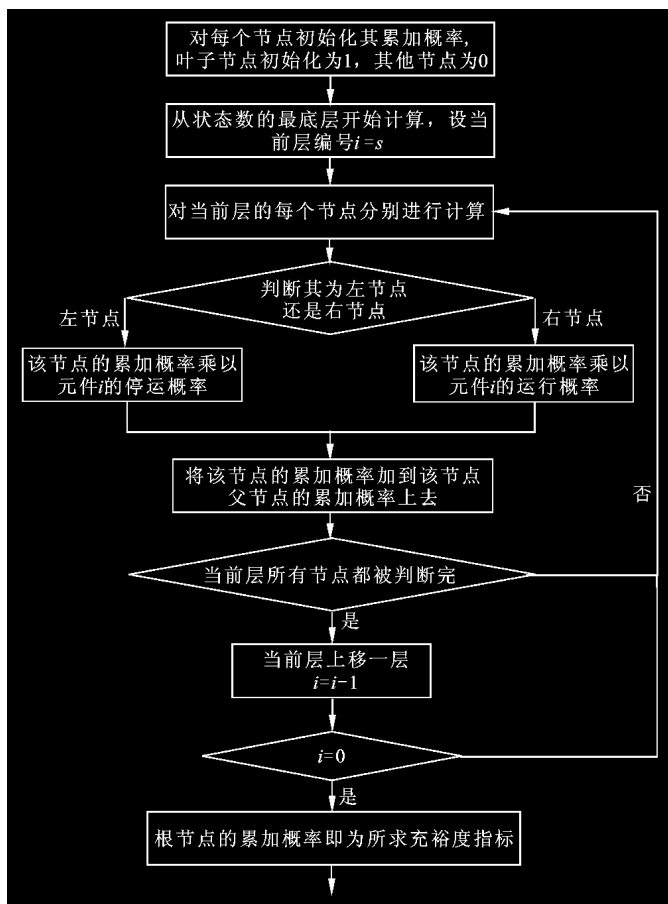


图3 组合事件概率求取流程

2 算法复杂度分析

2.1 复杂度证明

设系统有 s 个元件,其每一个状态都如式(2)所

示,元件从右侧开始依次编号为1到 s .

构造系统状态集 ϕ ,使其具有 2^k 个状态.可以证明,如果第 s 到 $k+1$ 个元件状态相同,第 k 到第1个元件取所有可能的状态组合,则在所有 ϕ 中,表示该状态集所需的二叉树节点数目最少

$$1 + s - k + \sum_{i=1}^k 2^i = s + 2^{k+1} - k - 1 \quad (10)$$

同样构造状态集 ϕ ,使其有 2^{k+1} 个状态.如果第 s 到 $s-k$ 个元件状态取所有可能的状态组合,其余元件状态相同,则在所有 ϕ 中,表示这样的状态集所需的二叉树节点数目最多,为

$$1 + \sum_{i=1}^{k+1} 2^i + 2^{k+1}(s-k-1) = (s+1-k)2^{k+1} - 1 \quad (11)$$

对于一般的状态集 ϕ ,设其有状态数 m ,则由式(10)得这些状态所需节点数下界的一个偏大估计为

$$O(s+m-\text{lb}m) \quad (12)$$

由式(11)得所需节点数上界的一个偏大估计为

$$O(ms - m\text{lb}m + 2m) \quad (13)$$

因本文算法对每个节点都要进行一次乘法计算,故上两式可以作为算法时间和空间复杂度估计的上、下界.

2.2 对比与结论

由以上分析,将本文算法复杂度的上、下界及与经典算法的对比列于表1,并得出以下结论.

(1)对系统元件数 s 的复杂度.设有效状态数 m 不变,本文算法复杂度介于 $O(s+m-\text{lb}m)$ 和 $O(ms+(2-\text{lb}m)m)$ 之间,经典方法为 $O(ms)$,可见两种方法复杂度都是 s 的线性函数.

本文算法增长速度介于1到 m 之间,小于经典方法.一般情况下, m 要远远大于 s ,因此本文算法复杂度的增长速度基本和经典法相同,由于上界中截距的存在,在数值上要好于经典方法.

(2)对有效状态数 m 的复杂度.设系统元件数 s 不变,考虑到一般 m 要远远大于 s ,本文算法介于 $O(m-\text{lb}m)$ 和 $O((s-\text{lb}m+2)m)$ 之间,经典法为 $O(sm)$,故两种算法都是线性复杂度的.

表1 本文方法与经典方法复杂度的对比

方法	时间复杂度			空间复杂度	
	插入新状态	全概率事件	一般情况	全概率事件	一般情况
经典方法	$O(sk/2)$	$O(m\text{lb}m)$	$O(ms)$	$O(m\text{lb}m)$	$O(ms)$
本文方法	$O(s)$	$O(2m)$	下限 $O(s+m-\text{lb}m)$	$O(2m)$	下限 $O(s+m-\text{lb}m)$
			上限 $O(ms-m\text{lb}m+2m)$		上限 $O(ms-m\text{lb}m+2m)$

本文算法增长速度介于1到 s 之间,小于经典法,且随 m 的增大,这一速度还在减小.当 $m=2^s$ 时,本文算法增长速度仅介于1和2之间,具有很好的效率.

(3) m 和 s 相关变化时的复杂度.实际情况下, m 和 s 往往不相互独立.定义抽样百分比 k 如下

$$k = \frac{m}{2^s} \times 100\% \quad 0 \leq k \leq 1 \quad (14)$$

那么,经典方法复杂度为

$$O(mlbm - mlbk) \quad (15)$$

本文方法下界为

$$O(m - lbk) \quad (16)$$

本文方法上界为

$$O((2 - lbk)m) \quad (17)$$

可见,此时经典法复杂度为超线性,本文方法仅为线性,因此在实际问题中本文方法更具有效率.

3 算例对比与分析

3.1 算法参数对性能的影响

这里不涉及具体系统,只是统计不同参数下性能的变化.系统元件数 s 可以反映相应系统规模的大小.这里假设所有的有效状态都是已知的和互斥的^[9],只应用本文方法和经典法对其进行组合概率的计算.本节比较两种算法中调用的乘法次数来说明其计算时间的差异,每组参数下计算100次并取平均值.

(1)系统元件数 s 对算法性能的影响.令有效状态数 $m=1\,024$ 不变,取 s 在10~100间变化,得统计结果如图4所示.

由图4可见,经典法和本文方法都是 s 的线性函数,具有几乎相同的增长速率,本文方法在数值上要优于经典法.这和2.2节的结论1是一致的.

(2)有效状态数 m 对算法性能的影响.令系统元件数 $s=15$ 不变, $m=2^{15}k$,取 k 在0到1.0间变化,得统计结果如图5所示.

由图5可见,经典法为线性复杂度,本文方法为斜率逐渐减少的线性复杂度,且增长速度始终小于经典法,这和2.2节的结论2是一致的.

特别指出,当 $k=1$,即全概率事件时,本文算法的复杂度为 $O(2m)$,这和使用全概率事件特点所得分析结果相一致,进一步验证了本文分析的正确性.

(3) m 和 s 相关变化对算法性能的影响.为了和实际情况相符,令 s 从11到20变化, $m=2^k$,随 s 相关变化,但 $k=50\%$ 不变,得统计结果如图6

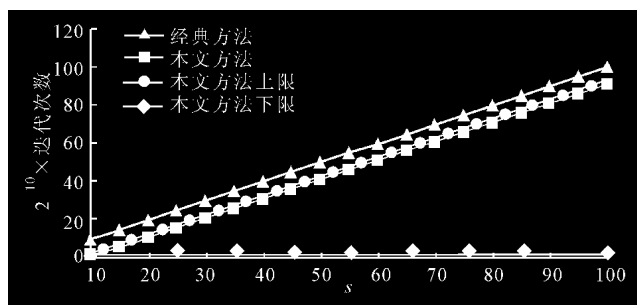


图4 系统元件数 s 对算法性能的影响

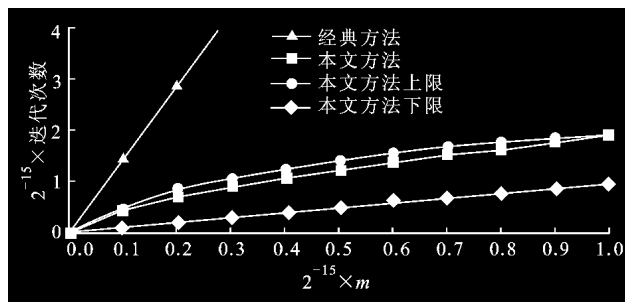


图5 有效状态数 m 对算法性能的影响

所示.

由图6可见,本文算法的复杂度是 m 的线性函数,而经典法为 m 的超线性函数,说明本文算法在实际系统中可能会比经典方法具有更好的效率,此结论与2.2节的结论3一致.应当指出,图6的横轴为元件总数 s ,而 m 正比于 2^s ,所以应视为关于 m 的半对数坐标系.

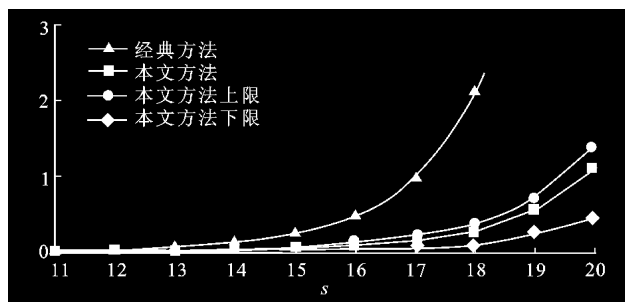


图6 m 和 s 相关变化对算法性能的影响

通过以上3种情况的计算和对比,显示了本文算法在任何情况下都具有线性复杂度,并且都要优于经典算法,这和上一节的理论分析是一致的,从而验证了本文算法的正确性和有效性.

3.2 IEEE RTS-24 系统算例

由于发电机组往往被模拟成多态元件,故只考虑线路故障,并假设故障率均为0.05.用两种算法分别计算线路发生3阶及以下故障时系统能满足一定负荷水平供电要求的概率^[9],结果如表2所示.

表2 IEEE RTS-24 系统算例比较

故障阶数	能满足负荷的故障方式数	经典方法		本文方法	
		所用时间/ms	满足负荷要求概率	所用时间/ms	满足负荷要求概率
0	1		0.174 825		0.174 825
1	34	< 1	0.312 844	< 1	0.312 844
2	551	318.91	0.266 838	25.525	0.266 838
3	5 658	34 075	0.144 213	307.960	0.144 213
总和	6 244	34 393	0.898 720	333.485	0.898 720

本例中,系统负荷水平取峰荷 2 850 MW. 表 2 中仍能满足负荷的故障方式数是指在相应的故障条件下系统仍能满足负荷供电要求的可行运行方式的总数,可以用来比较不同的可行状态规模对两种算法计算时间的影响.

由表 2 可见,两种算法得到了完全相同的结果,本文算法所用时间要少得多,因此本文算法是正确和有效的. 应当指出,此处数值偏小是因为没有考虑 3 阶以上的故障,这不属于本文讨论的范围. 另外,虽然此处仅举了形成系统满足负荷供电要求概率这一个指标的例子,但对系统其他充裕度指标的形成也可类似得出.

4 结 语

本文提出了一种使用二叉树的组合事件概率求取算法,可以用于电力系统充裕度分析中由可行状态求取系统指标的计算. 该方法原理清晰,实现简单,在系统所有可行状态已知的前提下,只需一次回代即可求出相应的系统指标. 理论分析和实际算例都表明,本文算法具有线性时间和空间复杂度,在效率上要优于经典方法. IEEE RTS-24 系统算例计算表明,本文算法更适用于大规模系统的可靠性分析.

本文算法的前提是要求所有的系统可行状态已知且两两互斥,这可能会对系统状态枚举和评估算法提出较严格的要求,这是本文算法的局限所在. 同时,因为本文假设所有元件都为两状态元件,因此采用了二叉树. 如果采用多叉树,则很容易实现多态元件系统的分析,这是未来可能的改进方向之一.

参考文献:

[1] US-Canada Power System Outage Task Force. Final report on the august 14, 2003 blackout in the United States and Canada: causes and recommendations [EB/OL]. (2004-10-12)[2006-08-22]. <https://reports.energy.gov/BlackoutFinal-Web.pdf>.

[2] 郭余庆,王岩. 系统可靠性理论及应用[M]. 北京:煤炭工业出版社,1991.

[3] RAI S, AGGARWAL K K. An efficient method for reliability evaluation of a general network[J]. IEEE Trans on Reliability,1978,R-27:206-211.

[4] 韩可琦. 矿业系统可靠性[M]. 徐州:中国矿业大学出版社,2002.

[5] ROSS S. Introduction to probability models[M]. New York,USA:Academic Press,1985.

[6] SINNAMON R M, ANDREWS J D. Fault tree analysis and binary decision diagrams[C]// Proceedings of IEEE Annual Reliability and Maintainability Symposium. Piscataway, NJ, USA: IEEE, 1996:22-25.

[7] YOSHIKAWA N, TAGO H, YONEYAMA K. A new design approach for RSFQ logic circuits based on the binary decision diagram[J]. IEEE Transactions on Applied Superconductivity,1999,9(2):3161-3164.

[8] ANDREWS J D, DUNNETT S J. Event-tree analysis using binary decision diagrams[J]. IEEE Transactions on Reliability,2000,49(2):230-238.

[9] 王锡凡. 电网可靠性评估的随机网流模型[J]. 电力系统自动化,2006,30(12):1-6.

WANG Xifan. Probabilistic network-flow models for reliability evaluation of power networks[J]. Automation of Electric Power Systems,2006,30(12):1-6.

(编辑 杜秀杰)