

配电网拓扑分析与三相潮流计算的实现

车仁飞, 李仁俊, 白树忠

(山东大学电气工程学院, 山东 济南 250061)

摘要: 根据配电网三相潮流计算的基本原理, 按面向对象技术设计并实现了配电网拓扑分析和三相潮流计算算法。潮流计算算法采用实用高效的前推回推法, 拓扑分析则根据前推回推法的特点采用支路分层的方法, 两者结合有效地保证了算法的效率。在实现中, 按面向对象技术设计了支路、节点、层和馈线类, 并在这些类的基础上实现了拓扑分析和潮流计算。利用 33 节点、123 节点和 292 节点系统对实现的算法软件进行了性能测试, 结果表明, 算法软件具有很高的效率。

关键词: 配电网; 三相潮流; 面向对象技术

中图分类号: TM13 **文献标识码:** A **文章编号:** 1003-4897(2003)03-0033-04

1 引言

潮流计算是电力系统一项基本计算。近年来, 随着配电网自动化的发展, 配电网中的潮流计算问题越来越受到人们的重视。

配电网具有不同于输电网络的显著特点: 辐射状或接近辐射状结构; 三相参数不对称; 三相负荷不平衡; 支路参数 r 和 x 相差不大, 有时甚至会出现 r 大于 x 的情况; 支路数和节点数十分庞大等。根据配电网的这些特点, 许多学者进行了大量研究, 提出了很多适合于配电网特点的算法^[1~5]。前推回推法^[1,4]就是其中之一。

前推回推法是计算辐射状配电网潮流的一种实用高效的方法, 在实际中得以广泛应用。本文根据前推回推配电网三相潮流计算的基本原理, 按面向对象的思想设计并实现了配电网拓扑分析和三相潮流计算, 并在实际工程中得以成功应用。

2 配电网三相潮流计算的基本原理

回推前推法的核心内容是一个建立在支路分层基础上的迭代过程。

在支路分层的基础上, 按如下步骤进行计算, 在第 k 次迭代:

(a) 计算节点注入电流:

$$\begin{bmatrix} I_{ja} \\ I_{jb} \\ I_{jc} \end{bmatrix}^{(k)} = \begin{bmatrix} (S_{ja}/U_{ja}^{(k-1)})^* \\ (S_{jb}/U_{jb}^{(k-1)})^* \\ (S_{jc}/U_{jc}^{(k-1)})^* \end{bmatrix} + \begin{bmatrix} Y_{ja} & 0 & 0 \\ 0 & Y_{jb} & 0 \\ 0 & 0 & Y_{jc} \end{bmatrix} \begin{bmatrix} U_{ja} \\ U_{jb} \\ U_{jc} \end{bmatrix}^{(k-1)} \quad (1)$$

其中, I_{ja}, I_{jb}, I_{jc} 为与恒定负荷功率和并联元件

相对应的节点 j 的注入电流; S_{ja}, S_{jb}, S_{jc} 为节点 j 已知注入功率; U_{ja}, U_{jb}, U_{jc} 为节点 j 的电压;

Y_{ja}, Y_{jb}, Y_{jc} 为节点 j 的所有并联元件的导纳。

(b) 回推计算各支路电流: 从最后一层支路开始向根节点推进, 支路 l 的电流为:

$$\begin{bmatrix} I_{la} \\ I_{lb} \\ I_{lc} \end{bmatrix}^{(k)} = - \begin{bmatrix} I_{ja} \\ I_{jb} \\ I_{jc} \end{bmatrix}^{(k)} + \sum_{m \in M} \begin{bmatrix} I_{ma} \\ I_{mb} \\ I_{mc} \end{bmatrix}^{(k)} \quad (2)$$

其中, I_{la}, I_{lb}, I_{lc} 为支路 l 的电流; M 为与节点 j 直接相连的所有下层支路的集合。

(c) 前推求解节点电压: 从第一层开始向最后一层推进, 节点 j 的电压为:

$$\begin{bmatrix} U_{ja} \\ U_{jb} \\ U_{jc} \end{bmatrix}^{(k)} = \begin{bmatrix} U_{ia} \\ U_{ib} \\ U_{ic} \end{bmatrix}^{(k)} - \begin{bmatrix} Z_{aa,l} & Z_{ab,l} & Z_{ac,l} \\ Z_{ba,l} & Z_{bb,l} & Z_{bc,l} \\ Z_{ca,l} & Z_{cb,l} & Z_{cc,l} \end{bmatrix} \begin{bmatrix} I_{la} \\ I_{lb} \\ I_{lc} \end{bmatrix}^{(k)} \quad (3)$$

在一次迭代执行了以上三步之后, 每个节点上所有相的功率不平衡分量为:

$$\begin{cases} S_{ja}^{(k)} = U_{ja}^{(k)} (I_{ja}^{(k)})^* - Y_{ja} |U_{ja}^{(k)}|^2 - S_{ja} \\ S_{jb}^{(k)} = U_{jb}^{(k)} (I_{jb}^{(k)})^* - Y_{jb} |U_{jb}^{(k)}|^2 - S_{jb} \\ S_{jc}^{(k)} = U_{jc}^{(k)} (I_{jc}^{(k)})^* - Y_{jc} |U_{jc}^{(k)}|^2 - S_{jc} \end{cases} \quad (4)$$

如果(4)中功率不平衡分量中有任何一个实部或虚部(有功或无功)大于收敛判据, 则重复(a)(b)(c)直至收敛。

3 基于面向对象技术的配电网三相潮流计算的实现

3.1 类定义

3.1.1 支路类

支路类是对线路和变压器等元件的抽象,内容包括支路号、支路阻抗、支路所属层的指针、首末节点指针等。类的定义如下所示:

```
class CBranch
{
public:
    CBranch(int &nBranchNo) {m. nBranchNo =
        nBranchNo; nBranchNo++;} //构造函数
    ~CBranch() {} //析构函数
private:
    int m. nBranchNo; //支路号
    CComplex complexZaa; //a 相自阻抗
    CComplex complexZab; //ab 相互阻抗
    CComplex complexZac; //ac 相互阻抗
    CComplex complexZbb; //b 相自阻抗
    CComplex complexZbc; //bc 相互阻抗
    CComplex complexZcc; //c 相自阻抗
private:
    CLayer * m. pLayer; //层指针
    CNode * m. pNode1; //始端节点指针
    CNode * m. pNode2; //末端节点指针
public:
    CComplex m. complexCurrentA; //a 相支路电流
    CComplex m. complexCurrentB; //b 相支路电流
    CComplex m. complexCurrentC; //c 相支路电流
};
```

3.1.2 节点类

节点类是对支路之间连接点的抽象,内容包括节点号、所连支路、三相注入功率、注入电流、节点电压等,类的定义如下所示:

```
class CNode
{
public:
    CNode(int &nNodeNo) {m. nNodeNo = nNodeNo;
        nNodeNo++;} //构造函数
    ~CNode() {m. branchlist. RemoveAll();} //析构函数
private:
    int m. nNodeNo; //节点号
    CBranchList m. branchlist; //所连支路链表
public:
    CComplex m. complexSInputA; //a 相注入功率
    CComplex m. complexSInputB; //b 相注入功率
    CComplex m. complexSInputC; //c 相注入功率
    CComplex m. complexCurrentInputA; //a 相注入电
    流
```

```
    CComplex m. complexCurrentInputB; //b 相注入电
    流
    CComplex m. complexCurrentInputC; //c 相注入电
    流;
    CComplex m. complexVoltageA; //a 相节点电压
    CComplex m. complexVoltageB; //b 相节点电压
    CComplex m. complexVoltageC; //c 相节点电压
};
```

3.1.3 层类

层类是对支路分层的抽象,每一层定义为一个层对象,内容包括层号、所包含支路链表等,类的定义如下所示:

```
class CLayer
{
public:
    CLayer(int &nLayerNo) {m. nLayerNo = nLayerNo;
        nLayerNo++;} //构造函数
    ~CLayer() {m. branchlist. RemoveAll();} //析构
    函数
private:
    int m. nLayerNo; //层号
    CBranchList m. branchlist; //包含支路链表
};
```

3.1.4 馈线类

馈线类是将馈线抽象形成的一个类,其内容包括:节点链表、支路链表、层链表以及一些在拓扑分析中用到的函数。类的定义如下所示:

```
Class CFeeder
{
public:
    CFeeder();
    ~CFeeder();
protected:
    CNodeList m. NodeList; //节点链表
    CBranchList m. BranchList; //支路链表
    CLayerList m. LayerList; //层链表
};
```

3.2 基于支路分层的馈线拓扑分析

拓扑分析的前提基础是网络建模,网络建模可以通过人工输入或由绘图工具自动建立,具体方法这里不再介绍。下面介绍支路分层方法及步骤。

3.2.1 支路分层方法

由于配电网的辐射状结构的特点,各条馈线之间只在根节点(10kV 母线)有电气联系,因此各条馈

线可在根节点处解耦,从而可以以馈线为单位进行拓扑分析和潮流计算。

为了计算方便,需要对节点、支路进行编号和对支路进行分层,本文采用的方法是:从根节点开始,进行广度优先搜索,节点的编号由顶及下顺序编号,支路由左至右顺序编号,同时由顶及下将支路进行分层。如图 1 所示,是对一个简单馈线进行编号和分层后的示意图,为清楚起见,在支路编号前加了一个‘L’。由图可见,支路 L1 对应节点 1 和 2, L2 对应节点 2 和 3;而节点 1 的对应支路是 L1,节点 2 的对应支路为 L1, L2 和 L3;层 1 包含的支路为 L1,层 2 包含的支路有 L2 和 L3,层 3 包含的支路有 L4, L5, L6, ……。

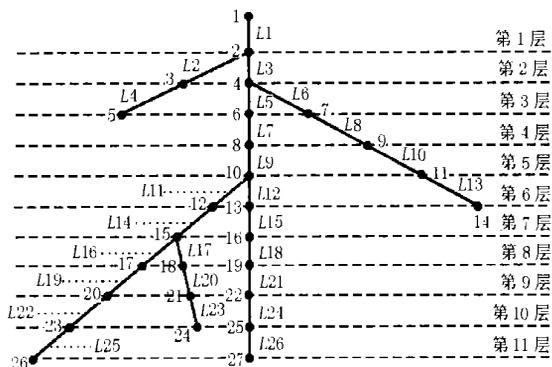


图 1 节点支路编号及支路分层示意图

Fig. 1 Schematic diagram of branch numbering and layering

3.2.2 支路分层步骤

基于支路分层的馈线拓扑分析的流程如图 2 所示,说明如下。

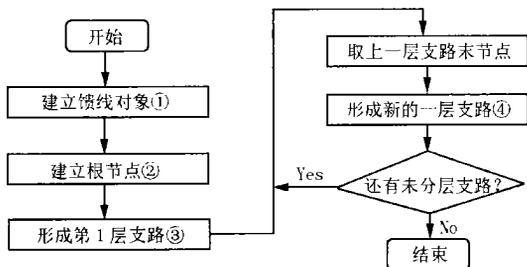


图 2 支路分层流程图

Fig. 2 The flowchart of branch layering

(a) 在框 中,建立馈线对象的同时,与之对应的节点链表、支路链表和层链表也建立了起来,初始状态为空。

(b) 在框 中,建立根节点时,将根节点对象添加到馈线对象的节点链表中。

(c) 在框 中,形成第 1 层支路,首先建立一个

新的层对象并将其添加到馈线对象的层链表中,然后按广度优先搜索与根节点相连的所有支路,这些支路形成了第 1 层支路,这些支路对象被顺序存放在馈线的支路链表和层的支路链表中,同样层对象的地址也被引用到每条支路的层指针变量中。每条支路的首节点对应根节点,末节点则新建一个节点对象与之对应,首末节点的支路链表中都增加对应支路的地址,新建的末节点对象按顺序添加到馈线的节点链表中。

(d) 在框 中,形成新的支路层,首先建立一个新的层对象并将其添加到馈线对象的层链表中,然后依次取上层支路的末节点,按广度优先搜索与末节点相连的支路(除上层支路外),所有这些支路形成了新的支路层,这些支路对象被顺序存放在馈线的支路链表和层的支路链表中,同样层对象的地址也被引用到每条支路的层指针变量中。每条支路的首节点对应上一层对应支路的末节点,每条支路的末节点则新建一个节点对象与之对应,首末节点的支路链表中都增加对应支路的地址,新建的末节点对象按顺序添加到馈线的节点链表中。

(e) 馈线拓扑分析结束后,馈线对象的节点链表、支路链表和层链表三个链表中存放的就是拓扑分析的结果,供后面的三相潮流计算使用。三个链表中存放对象之间的关系,可以用图 3 来说明。由图 3 可以看出,层链表、支路链表和节点链表中的对象都是按照建立或搜索的先后顺序来排列并编号的;每个层对应所包含的若干支路;每条支路对应某个层;每条支路对应首末两个节点;每个节点对应所连的若干支路。

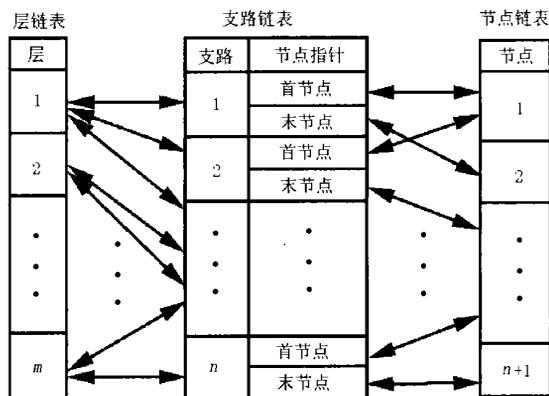


图 3 节点、支路和层之间关系图

Fig. 3 The relationship of node, branch and layer

3.3 三相潮流计算的实现

实现三相潮流计算总的步骤如下:

- (a) 按图 2 所示步骤进行支路分层。
 (b) 读入支路参数,初始化各节点电压。
 (c) 顺序访问节点链表中的各个节点,按式(1)计算各节点三相注入电流。
 (d) 逆序访问层链表中的各个层(即从最后一层回推),按式(2)计算每层中的各支路三相电流。
 (e) 顺序访问层链表中的各个层(即从第一层前推),按式(3)计算每层中的各支路末节点三相电压。
 (f) 顺序访问节点链表中的各个节点,按式(4)计算各节点三相功率的不平衡量,并检查各相不平衡量是否小于收敛值,如果是则退出,否则返回到(c)继续执行。

4 算例测试

配电网三相潮流计算是配电网高级应用软件的基本部分。作者利用 Visual C++ 实现了三相潮流计算软件,该软件可运行于 Windows 98、Windows NT 或 Windows 2000 操作系统之上。为了测试算法软件的收敛性能和运算效率,利用 33 节点系统^[6]、123 节点系统^[7]和 292 节点系统^[8]进行了测试,表 1 给出了性能测试结果。测试结果说明,实现的算法软件具有很高的效率,能够满足实际系统的需要。

表 1 性能测试结果

Tab. 1 Performance test result

测试系统	迭代次数	CPU 时间(s) (IBM PIII,主频 550MHz)
33 节点	4	0.005
123 节点	5	0.020
292 节点	6	0.045

5 结论

实现配电网潮流计算,是进行配电网分析的基本内容。本文按照面向对象技术设计实现的配电网拓扑分析和三相潮流计算,具有简单、实用、效率高等特点。在实现中,将拓扑分析的分层方法与潮流计算的前推回推法有效地结合起来,提高了计

算效率。该潮流计算软件已成功地应用到潍坊配电网自动化系统中。

参考文献:

- [1] D. Shirmohammadi, H. W. Hong, et al. A Compensation Based Power Flow Method for Weakly Meshed Distribution and Transmission Networks [J]. IEEE Transaction on Power Systems, 1998, 3(2): 753-762.
- [2] G. X. Luo, A. Demlyen. Efficient Load Flow for Large Weakly Meshed Networks [J]. IEEE Transaction on Power Systems, 1990, 5(4): 1309-1316.
- [3] Tsai-Hsiang Chen, Mo-Shing Chen, et al. Distribution System Power Flow Analysis-A Rigid Approach [J]. IEEE Transaction on Power Delivery, 1991, 6(3): 1146-1152.
- [4] Carol S. Cheng, Dariush Shirmohammadi. A Three-Phase Power Flow Method for Real-time Distribution System Analysis [J]. IEEE Transaction on Power Systems, 1995, 10(2): 671-679.
- [5] Ray D. Zimmerman, Hsiao-Dong Chiang. Fast Decoupled Power Flow for Unbalanced Radial Distribution Systems [J]. IEEE Transaction on Power Systems, 1995, 10(4): 2045-2051.
- [6] Mesut E. Baran, Flix F. Wu. Network Reconfiguration in Distribution Systems for Loss Reduction and Load Balancing [J]. IEEE Transaction on Power Delivery, 1989, 4(2): 1401-1407.
- [7] IEEE Distribution Planning Working Group Report. Radial Distribution Test Feeders [J]. IEEE Transaction on Power Systems, 1991, 6(3): 975-985.
- [8] Jirn-Cheng Wang, Hsiao-Dong Chiang, Gary R. Darling. An Efficient Algorithm for Real-Time Network Reconfiguration in Large Scale Unbalanced Distribution Systems [J]. IEEE Transaction on Power Systems, 1996, 11(1): 511-524.

收稿日期: 2002-05-10

作者简介:

车仁飞(1971-),男,讲师,博士研究生,主要从事配电网自动化方面的研究;

李仁俊(1939-),男,教授,博士生导师,主要从事电力系统运行与控制 and 配电网自动化方面的研究;

白树忠(1965-),男,讲师,主要从事配电网自动化方面的研究。

Implementation of topology analysis and three-phase power flow for distribution systems

CHE Ren-fei, LI Ren-jun, BAI Shu-zhong

(School of Electric Engineering, Shandong University, Jinan 250061, China)

Abstract: According to the principle of three-phase power flow for distribution network, the distribution network topology analysis and three phase power flow are designed and implemented in this paper, and the implementation is based on the object-oriented technique. The backward/forward method is adopted in power flow calculation, and the branch layering method is used in the topology analysis. The combination of two methods guarantees the efficiency of power flow algorithm. Moreover, the object classes based on the object-oriented technique are designed to implement the power flow calculation. The performance test results of 33 nodes system and 123 nodes system and 292 nodes system are also offered in this paper, and the results indicate that the realized program has high efficiency.

Key words: distribution system; three-phase power flow; object-oriented technique