

Abstract

Recently, several foreign power system faults caused a lot of loss. It makes the study of power system fault diagnosis more urgent. Moreover, after the fault, the quick and accurate diagnosis of fault component is the precondition of repair and recover power. So the diagnosis of fault component became the most important in the power system fault diagnosis.

The key problem of power system fault diagnosis is the distinguish of fault equipments in fault area quickly and accurately. Base on protect and breaker information, this paper uses minimal deviation between expected value and actual value as objective function. Because the exponent form is prompt with the fact that the malfunction area inner waiting for increasing by diagnosing component quantity, target function calculation number of times presenting enhance, lead to adopt enumeration way or the tradition branch demarcation law can not satisfy the need searching for the optimum solution rapidly. This paper finds the solution with the target changing becoming 0-1 integer plan problems. Not only genetic algorithms does not to need to check knowledge regulation and to train a sample book directly, but also the number of times of search is the fewest, and the fault component could be recognized.

Base on the protection action principle, this paper sums up the generator, transformer, transline and busbar detailedly, and designs malfunction diagnose inheritance algorithm C++ procedure. Procedure is accomplished because C language procedure rewrites much according to current inheritance algorithm, the code multiplexing being able to not embody be geared to the needs of marriage partner programming waits for characteristic propert. This paper has designed that the inheritance algorithm kind and the dyeing individual kind have realized the data encapsulation , inherit and much state sex. For the problem such as overlaborate place operation and memory assignment resolving procedure middle, the function realizes the main body of a book owing to that STL has designed inheritance algorithm malfunction diagnose algorithm procedure data structure and inheritance algorithm, urge the C++ code is especially concise, high-effect and reliable and so on. The union electric system malfunction diagnose

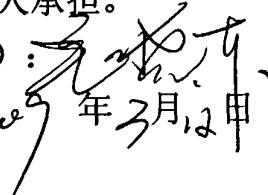
example and , the verification having carried out the malfunction diagnose model and the algorithm procedure, have confirmed accurate diagnose result reasonableness.

For satisfying the need that the power system diagnose studies and the effective storage and invoking about the fault information of power system, on the basis studying the vector artwork technology's, this paper defines coordinate system and mapping relation, realizes power equipment picture basic drawing and manage. The platform composes with the draw of power component and the operation of database. This paper realize the draw function of power component, achieves mapping between power component and database record, and satisfies the need of power system fault diagnosis. Meanwhile, this paper makes further verify about the rightness of common apply of the platform of graphic and database and fault diagnosis base on IEEE power system with five generater and fourteen nodes.

Key words: power system fault diagnosis; genetic algorithm; STL; graphic and database

哈尔滨工程大学学位论文授权使用声明模板
哈尔滨工程大学
学位论文原创性声明

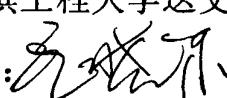
本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：
日期：2009年3月12日

哈尔滨工程大学
学位论文授权使用声明

本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名单位为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文在授予学位后即可 在授予学位12个月后
解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）： 导师（签字）：
日期：2009年3月12日 年 月

第1章 绪论

1.1 研究背景及意义

近年来美国、加拿大及欧洲等国电力系统发生的大停电事故，引起了世界范围的广泛关注。尽管我国多年来还没有出现过类似欧美国家发生的大规模停电事故，但绝不能说我们的电网稳定性已经达到了很高的水平。相反，随着经济快速增长，电力系统也向大机组、大电网、高电压和远距离输电发展。目前我国正在加紧进行各大区域电力系统联网，在实现“西电东送”、“南北互供”合理利用资源的同时，也增大了停电事故的风险，给电力系统安全运行带来了一些新的问题。保证电力系统能在正常情况下安全、稳定、经济运行，发生事故时能迅速判断、处理事故，使电网尽快恢复正常供电，是对国家电网和各大区电网调度运行人员提出的新要求。显然，调度员必须具有很高的业务水平，极强的责任心，同时还应该给调度员配备高水平的调度自动化分析软件，以协助调度员对电力系统进行调整、控制和操作。有助于调度人员面对复杂的实时信息，快速做出判断决策。由于电力系统故障分析处理是调度运行的重要内容，所以电力系统故障诊断系统也必然成为决策的重要部分，负责在电力系统事故状态下，快速、准确的对故障进行分析处理，确定故障元件、故障类型和故障位置等。

1.2 国内外研究状况

电力系统故障诊断是近年来十分活跃的研究课题之一。主要包括系统故障诊断和元件故障诊断两个方向^[1]，系统级故障诊断是指通过分析电网中各级各类保护装置产生的报警信息、断路器的状态变位信息以及电压电流等电气量测量的特征，根据保护、断路器动作的逻辑和运行人员的经验来推断可能的故障元件和故障类型的过程。诊断重点在于故障元件的识别，是对电力系统故障进行分析的重要环节。元件级故障诊断则指对系统运行中发生故障的电气元件进行的故障性质的分析，主要有线路故障、变压器故障等。本文主要研究系统级故障诊断。

在进行故障诊断时，必须依据一定的信息，按进行诊断判断所依据的信

息源，电力系统故障诊断有以下几种：

- (1) 利用SCADA系统/故障信息系统提供的开关变位信息进行诊断^[2]。
- (2) 利用SCADA/EMS系统/故障信息系统提供的开关和部分（或完全）保护信息进行诊断^{[3][4][5][6]}。
- (3) 利用SCADA/EMS/故障信息系统提供的开关、保护信息和故障录波器的录波信息进行分层诊断^{[7][8][9][10]}。

其中开关变位信息主要包括由继电保护动作引起的开关跳合闸信息，以及自动重合闸装置动作引起的开关闭合信息。保护信息事故发生后保护动作信息包括保护跳闸报告、保护软压板状态报告、保护录波报告（部分保护有）、保护自检报告等。

在进行电力系统故障诊断时，除了必要的判断信息以外，还要有适应的方法，所以还要进行诊断方法的研究。

传统电力系统故障诊断研究是在建立被诊断系统网络拓扑结构模型的基础上，根据发生故障时系统结构和参数变化，导致系统潮流的变化，进而根据潮流计算的变化判断出故障，多用传统的数学方法，采用单一的集中求解，因系统规模、复杂程度和不确定因素等的限制难以适应目前电力系统的这样一个日趋复杂的分布式大系统的故障诊断问题发展趋势，系统故障诊断难以达到理想的效果。因此，目前研究电力系统故障诊断的方法主要是采用的智能化方法。近几十年来，故障诊断技术得到了深入广泛的研究，提出了众多可行的方法，概括起来可分为三大类^[11]：

(1) 基于解析模型的方法，在了解诊断对象数学模型的基础上，按一定的数学方法对被测信息进行处理判断，可分为状态估计法、等价空间法和参数估计法等。基于解析模型的故障诊断方法是最早发展起来的，其主要思想是通过构造观测器估计出系统输出，然后将它与输出的测量值作比较，从中获得故障信息。由于建模的困难与模型本身的误差以及各种不可预见的因素（如系统发生故障时，不仅可能引起模型参数的变化，还可能引起模型结构的变化，而且这种变化是不确定的），大大地影响了其诊断的准确性。

(2) 基于信号处理的方法，利用信号模型，如相关函数、频谱、自回归滑动平均、小波变换等，直接分析可测信号，提取诸如方差、幅值、频率等特征值，从而检测出故障。这种方法由于不需要建立对象的解析数学模型，

实现简单，在工程上具有广泛的应用，但这种方法只有当故障发展到相当程度并影响到外部特征时才有效，而且只能对故障范围做出粗略的判断，大多数情况下不能直接定位故障。

(3) 基于知识的诊断方法。这类方法的主要优点是不依赖于具体的数学模型，而且克服了基于信号处理故障诊断方法的缺点，引入了诊断对象的许多信息，具有较为丰富的、灵活的知识表达和问题求解能力，它可以充分发挥人类专家在诊断中根据感觉、知识、经验所进行的推理判断的能力，并可适合于各种场合的故障判别。

其中，基于知识的故障诊断方法由于其本身具有的优点已经成为故障诊断领域中的一个主要方法，它不仅可以进行离线的故障诊断，还可以用于在线的故障诊断与故障处理。在电力系统中，这类方法的开发研究也是最为引人瞩目的一类课题，国内外也有大量的文献介绍基于知识的诊断方法在电网故障诊断中的应用。近年来，一般报警信息的处理和常见简单故障的诊断问题已经得到较好的解决。主要的方法有：基于专家系统的方法ES (Expert System)、基于人工神经网络的方法ANN (Artificial Neural Network)、基于模糊理论的方法FT (Fuzzy Theory)、基于遗传算法的方法GA (Genetic Algorithms)、基于Petri网的方法等。

专家系统 (Expert System) 是发展最早，也是比较成熟的一种人工智能技术。一般地说，它是一个具有大量专业知识与经验的程序系统，根据某个领域的专家提供的知识和经验进行推理和判断，模拟专家的决策过程，以解决那些需要专家才能决策的复杂问题。

专家系统可以进一步划分为：基于规则系统 (Rule-Based System，简称RBS)、基于模型系统 (Model-Based System，简称MBS) 和基于逻辑系统 (Logic-Based System，简称MBS) 等三种类型。

电网故障诊断专家系统大多基于规则系统。在这类系统中保护、断路器的动作逻辑及运行人员的诊断经验通过规则表示出来，形成故障诊断知识库，进而根据报警信息对知识库进行推理，获得故障诊断的结果。有两种基本的推理方式：前向推理和后项推理。前向推理是一种数据驱动推理的方式，而后向推理是一种目标驱动推理方式。对于故障征兆有限而故障原因很多的问题，可以采用前向推理技术，否则，后项推理技术更为合适。在求解某些复

杂问题时，可以综合使用前向推理和后向推理技术。

基于模型的专家系统优点在于提供了深层知识及推理描述功能，从而提高了专家系统的问题求解与解释能力。模型库中建立线路、变压器、母线等一次设备和保护装置的物理关系模型。当系统发生故障时，专家系统将接收到一系列来源于故障录波器、保护装置或者SCADA/EMS的报警和事件信息，专家系统将在模型库中进行模拟，以搜索出所获得的报警或事件协调一致的故障原因。

基于逻辑的专家系统是根据保护系统隔离故障的原理和逻辑进行推理的，系统中表达出保护和一次设备的关系、主保护和各段后备保护的配合逻辑，以及重合闸原理等。系统不但表达正常工作逻辑，而且还表达保护、断路器在误动、拒动等情况下的逻辑，以便处理这些装置的不正确动作。在系统发生故障并接收到故障报警或者事件后，根据故障征兆做出多种故障假设，并分别进行分析，最终利用评价标准给出每条假设成立的可信度，并把具有最高可信度的假设作为可能的故障诊断结论提供给用户。

人工神经元网络（Artificial Neural Network，简称ANN）是模拟人类神经系统传输、处理信息过程的一种人工智能技术。自1943年McCulloch和Pitts第一次提出模拟神经元以来，它迅速发展成为与专家系统并列的人工智能技术的另一个重要分支。与专家系统相比，人工神经网络最大的特点是采用神经元及它们之间的有向权重连接来隐含处理问题的知识，具有自我学习的能力和容错能力比较强的特点。文献[2]介绍了一种利用人工神经网络实现变电站故障诊断的方法，该方法充分利用人工神经网络所具有的、强大的学习能力及高度的容错性特点，实现对变电站故障元件的诊断。仿真结果表明，该方法不仅能准确地诊断出保护、开关正确动作时的故障元件，也可有效地诊断出因保护或开关拒动，越级故障时的故障元件。

基于模糊理论（Fuzzy Sets Theory，简称FST）是将经典集合理论模糊化，并引入语言变量和近似推理的模糊逻辑，具有完整推理体系的智能技术。一般模糊系统的结构与专家系统的机构类似，由模糊知识库、模糊推理机和人机界面等几部分组成。其模糊知识库使用语言变量来表述专家的经验，更接近人的表达习惯，能够得到问题的多个、可能的解决方案，并可以根据这些方案的模糊度高低进行优先程序排序等。

模糊理论在电力系统故障诊断的应用中分两类情况：第一类认为诊断所依据的信息正确，但故障与对应的动作保护装置和断路器状态之间存在不确定的关联关系，用模糊隶属度来对这种可能性进行描述性的度量；另一类则是认为诊断所依据的报警信息的可信度不为1，而根据系统网络拓扑与故障所发生动作保护、断路器状态赋予报警信息的可信度，再由ES或ANN给出故障诊断结果的模糊输出。

随着电力系统建设的发展、计算机技术和网络技术以及数学和智能科学理论的发展，不断有新的电力系统故障诊断方法出现，但是从电力系统故障诊断理论与方法研究和应用的深度、广度来看，这些研究仍停留在理论和模型的探索阶段，实用化方面需要进一步的探索。

1.3 论文主要工作

(1) 探讨了电力系统故障分类、故障征兆信息以及电力系统故障处理方式。在研究电力系统继电保护原理及故障分析过程的基础上，介绍了利用保护和断路器动作信息识别发生故障元件的电力系统故障诊断模型。通过实例，归纳总结了包括发电机故障、变压器、母线和输电线路等四类电力设备的保护动作原理。

(2) 将电力系统故障诊断问题转化为0-1整数规划的求解问题，并针对电力系统故障诊断的快速性要求，介绍了利用遗传算法快速获取最优解，确定故障元件的方法。鉴于传统电力系统故障诊断软件对算法程序的要求^[20]，文中采用面向对象方式和泛型算法设计电力系统故障诊断遗传算法诊断程序，不但代码量缩短到传统程序设计的三分之一，摆脱了烦琐的内存管理和指针处理过程，提高了代码的可读性、健壮性和可扩展性。文中通过线路系统的两个故障实例验证了诊断模型和算法软件的正确性。

(3) 为了实现故障信息的有效存储和调用，开发了具有Ribbon界面风格的电力图形数据库平台，用以支持故障诊断算法程序的应用研究。平台实现了绘制电力设备图形及对设备图形的拖拽、缩放、复制和粘贴等功能。电力设备与数据库记录实现对应的绑定存储。为了进一步验证故障诊断应用程序与图形数据库的性能，通过IEEE中5机14结点系统的故障实例，对故障诊断

过程进行了验证，验证结果表明了数据库信息和算法程序配合正确，诊断结果合理。

第2章 电力系统故障信息及处理

电力系统故障诊断过程就是从被诊断系统的某些检测量中得到故障征兆信息，通过对这些征兆信息的分析处理，判断故障源的过程。本章在介绍电力系统的常见故障情况的基础上，对电力系统故障信息类型和特征给予了分析，并介绍了电力系统发生故障后的常规分析处理过程。

2.1 常见的电力系统故障

电力系统故障是指设备不能按照预期的指标工作的一种状态，也可以说是设备未达到其应该达到的功能^[12]。电力系统可能发生的故障类型比较多，其中短路故障是电力系统中最常见、危害最严重的故障。短路是指电力系统正常运行情况以外的一切相与相之间或相与地之间的短接。在电力系统正常运行时，除中性点外，相与相或相与地之间是绝缘的，如果由于某种原因使其绝缘破坏而构成了通路，就称电力系统发生了短路故障。产生短路的主要原因是电气设备载流部分的绝缘损坏。电力系统的运行经验表明，短路故障主要是单相接地短路。

故障状态：电气元件发生短路、断线时的状态均为故障状态。最常见且最危险的故障状态是各种类型的短路。

电力系统在运行中，所有元件均可能出现各种故障状态和不正常运行状态，这时若是处理不当，将引起电力系统的事故。故障状态主要可以划分为输电线路故障，变压器故障，母线故障及全厂、全所停电等。

2.1.1 输电线路故障

输电线路的故障，通常指由雷电引起的绝缘子表面闪络，大风引起的短时碰线，树枝等物掉落在导线上引起的短路等等。当线路被继电保护迅速动作控制断路器跳闸后，故障点的绝缘平衡可自行恢复，故障随即消失。此时，如果把跳闸的线路断路器重新合上，就能够恢复正常供电。另外也有“永久性故障”，例如，由于线路倒杆、断线、绝缘子击穿或损坏等引起的故障，在线路被断开之后，它们依然是存在的。常见的输电线路故障包括以下4种。

(1) 单相接地短路故障

单相接地短路是最常见的故障，约占全部故障的80%以上。对于中性点直接接地系统，发生单相接地时，要求迅速切除故障点。对于中性点不接地或中性点经消弧线圈接地的系统，发生单相接地时，允许短时间带电运行，但要求尽快寻找接地点，将接地部分退出运行并进行处理。

(2) 两相接地短路故障

两相接地短路一般不会超过全部故障机率的10%。在中性点直接接地系统中，这种故障多在同一地点发生；在中性点非直接接地系统中，常见情况是先发生一点接地，而后其他两相对地电压升高，在绝缘薄弱处形成第二接地点，此两点多数不在同一点。

(3) 两相短路及三相短路故障

两相短路及三相短路相对较少，一般不超过全部故障机率的5%，但这种故障比较严重，故障发生后要求更迅速的切除。

(4) 断相故障

断相故障是指电力系统一相断开或两相断开的情况。这种故障也属于不对称性故障，此时电力系统处于非全相运行状态。断相的故障机率更少，约为1%，这种故障造成系统非全相运行，一般不允许长期存在，应由继电保护装置自动切除，或由工作人员手动断开其他非故障相。

2.1.2 变压器故障

变压器故障涉及的面很广，例如，油浸式变压器的故障常被分为内部故障和外部故障两种，故障诊断的重点是针对内部故障。从性质上来分，内部故障一般可以分成热故障和电故障两大类。热故障通常为变压器内部局部过热、温度升高，根据其严重程度，热故障常被分为轻度过热（一般低于150℃）、低温过热（150~300℃）、中温过热（300~700℃）、高温过热（一般高于700℃）四种故障情况。电故障通常指变压器内部在高电场强度的作用下，造成绝缘性能下降或劣化的故障，根据放电的能量密度的不同，电故障又分为局部放电、火花放电和高能电弧放电几种故障类型。

如果从回路划分变压器故障，主要有电路故障、磁路故障和油路故障。

从变压器主体结构上来划分主要可分为绕组故障、铁心故障、油质故障和附件故障。工程人员习惯上根据变压器故障易发生的区域进行划分，如绝缘故障、铁心故障、分接开关故障等等。

(1) 绝缘故障

大多数变压器的损坏和故障都是因为绝缘系统的损坏而造成。根据统计，因为各种类型的绝缘故障而产生的事故约占全部变压器故障的85%以上。在油浸式变压器中，绝缘系统就是由液体绝缘介质和固体绝缘介质组成的。液体绝缘介质就是变压器油，固体绝缘介质就是绝缘纸、绝缘板、绝缘垫等，其主要成分是纤维素。对变压器绝缘系统的故障有着重要影响的因素有：温度、湿度、油保护方式和过电压等。

(2) 铁心故障

变压器的器身主要绕组和铁心构成，他们是变压器传递、交换电磁能量的主要部件，铁心不仅要求质量好，还必须有可靠的一点接地。铁心只有一点接地时，变压器才能正常运行，当出现两点及以上的接地，为多点接地。多点接地会导致的后果有：在铁心中产生涡流，铁耗增加，铁心局部过热；长时间未处理，会导致油及绕组过热，使油纸老化；长时间的多点接地，会使得变压器油劣化分解并产生可燃气体，使气体继电器动作，造成停电事故；严重的多点接地会使得接地线烧断，使变压器失去了正常的一点接地，后果不堪设想。

(3) 出口短路故障

出口短路故障指运行的变压器由于受到出口短路故障的影响而遭到的破坏。变压器出口短路时，其高、低压绕组可能同时通过数十倍于额定值的短路电流，同时产生很大的热量，使变压器严重发热，损坏绝缘。另外，变压器受短路冲击时，如果短路电流小，保护及时动作，绕组会发生轻微变形；如果短路电流大，保护延时动作，甚至保护未动，绕组就会出现严重变形。一台绕组已经变形的变压器如果继续运行，又会引起多种故障和事故。

(4) 绕组故障

各类变压器的绕组均是由带绝缘层的绕组导线按一定排列规则和绕向，经绕制、整形、浸烘、套装而成。由于在绕组的生产不当、运输中受伤、受潮、各类过电压及过电流冲击等，致使绕组绝缘层受到损伤、老化、劣化，

造成绕组的短路、断路、变形等故障。这些故障会导致变压器出现过热、局部放电和电弧放电等故障情况。

2.1.3 母线故障及全厂、全所停电

在电力系统中，枢纽变电所的母线发生故障时，可能会使系统用户停电，使相关联络线过载，严重的还可能引起系统稳定的破坏，造成严重的后果。发生母线故障的通常包括母线短路或母线保护拒动、误动作；送出线故障引起的越级跳闸；发电厂厂内事故又引起系统联络线跳闸，或者反之，系统联络线跳闸引起全厂停电；单电源供电的降压变电所受电线路故障。

2.2 电力系统故障征兆及特征

2.2.1 电力系统故障征兆信息

电力系统故障诊断以故障后所表现的征兆信息为基础^[14]。当电力系统发生故障时，首先反映的是电力系统各节点电压、支路电流（或功率）等电气量的变化，之后是保护装置依据电气量信息按照整定值对故障的判断生成保护动作信息，然后由保护跳开相应的开关来隔离故障，形成断路器动作信息。

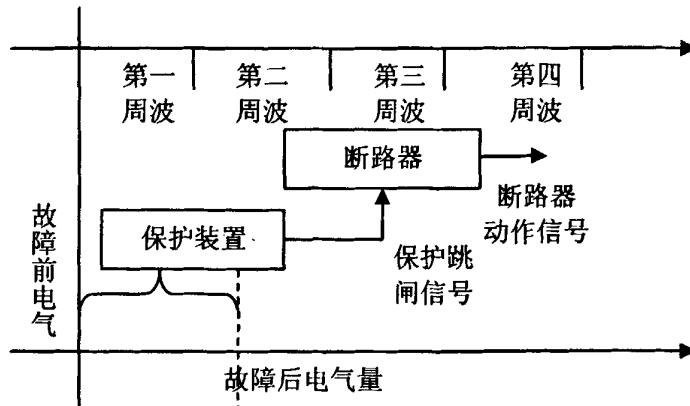


图 2.1 故障征兆信息

换句话说，故障诊断求解可以利用的信息包括：电气量信息、开关量信息、保护信息、开关和保护所带时标的时间顺序记录信息，参见图2.1。

2.2.2 电力系统故障信息特征

分层性：故障征兆信息具有一定分层特性，一层征兆信息是在另一层征兆信息基础上产生的。电力系统发生故障后，首先反映的是电力系统各节点电压、支路电流（或功率）等电气量的变化，之后是保护装置的动作信号，最后是断路器动作信息。

冗余性：从征兆信息分层角度来看，一层征兆信息可以由前一层征兆信息推出，如断路器动作信息可以由保护动作信息推出。所以，对于同一故障，不同层次的故障征兆信息，虽然具备不同形式，但从征兆信息所反映的故障内容来说具有一定冗余性。从一般诊断理论来说，尽量不使信息冗余。但对于电力系统故障诊断问题，在复杂故障情况下以及保护设备和断路器有异常工作或信息丢失时，要想对故障元件进行详细诊断定位，可利用征兆信息的冗余特性，也就是利用关于故障征兆信息间关系的知识，对故障元件进行诊断。

间断性：继电保护和故障录波装置记录的是电力系统故障的暂态数据，电力系统故障时它们将会在短时间内采集到大量数据，电力系统故障时，若这些数据都要快速、准确的传送到调度中心，信道将十分繁忙，而在大多数的时间，信道是空闲的，即具有间断性。

不规范性：从目前设备运行情况来看，各继电器和故障录波器的数据通讯都是封闭的，不对外开放。每个制造厂或某一型号的装置都有一套自己的通讯、分析软件，采用不同的通讯规约和数据格式，相互间无法进行数据共享。欲充分利用全网各种自动装置采集的信息，需要进行规约转换，转换成自己需要的规范统一格式，这将是一份十分繁琐的工作。

2.3 故障信息系统

故障信息系统^[13]能够实现故障信息的整合功能，对变电站内不同厂商，不同类型的继电保护装置和故障录波器的故障信息进行汇集、分类、规约转换，经电力通信网将信息上送到电网调度中心或接受电网调度中心下达的信息调用，在电网调度中心的主站系统实现故障信息的综合利用分析。

2.3.1 信息接入

随着众多现代信息科技的发展，计算机技术、微电子技术和网络技术促进了电力系统故障诊断技术的发展。例如：

数字式保护：数字式保护采集交流电流、电压信息，经A/D转换成低电平的数字信息，保护装置的CPU根据保护动作原理，执行数据处理、数值运算、逻辑判断等一系列功能，在满足保护内部整定值的情况下给出保护动作的结果，同时提供反映保护基本情况的硬接点信息（如装置报警、出口动作跳闸等），和反映保护动作信息的数据输出信息（如电流、电压信息，装置内部整定信息，装置各元件动作信息等）。

故障录波器：作为电力系统故障或异常情况记录的“黑匣子”，故障录波器能够提供比事故顺序记录仪更详尽的信息。能够记录电力系统大扰动、震荡、频率崩溃、电压崩溃等事件发生后的相关信息，比如系统电参量变化过程及继电保护与安全自动装置的动作行为。

测控单元/RTU：不但能完成交流量信息采集，还可以对原始数据进行计算，同时具备SOE记录器的功能，能够记录断路器状态和各种保护情况，成为故障录波器信息完整性的重要补充。

为了将涉及电网故障的各种信息有机关联并综合利用，在厂站端设置一套具有数据接口、数据通信、数据处理及存储功能的设备。以完成数字式保护和故障录波器的信息接入，实现不同设备的规约转换，按寻访机制实现对接入设备的定时访问，获取新的事件信息，实现对信息的智能预处理，同时对重要信息主动上送，或者等待主站信息调用。

接入故障信息系统的信息主要有以下几类^[13]：

- (1) 安装在电网中各运行厂站的数字式继电保护装置的动作信号；
- (2) 安装在电网中各运行厂站的数字式故障录波器的模拟量信号；
- (3) 经故障录波器事件量转接的断路器分/合闸信号；
- (4) 经故障录波器事件量转接的非数字式保护等。

2.3.2 数据传输

可靠的电力通信网络构成了故障信息系统的重要环节，是实现故障信息系统各种应用的基础。一般通过电力通信网络进行数据传输业务的方式有拨号方式、专线方式、数据网方式，基础传输网络为微波或者光纤传输网。

通常拨号方式不能实现对通道状况的实时监视，只能适合离线数据调用，这样会极大的降低故障信息系统作为调度决策支持的辅助手段的效果。因此故障信息系统数据传输通道优先选择数据网络方式，在数据网络不具备的情况下可以选择专线方式。

2.3.3 综合应用

故障信息系统应用价值体现在“在适当的时机，将合适的信息传递给最需要的人”^[19]。因此，在子站信息接入、预处理、传输的基础上，故障信息系统的效用主要通过主站端的应用软件所实现具体的功能。

通常电力故障信息主站建立了信息收集、管理平台，产生了对每次事故所收集信息的累积效应，对各种数据的有效管理是发挥系统效用的重要基础。主站端对各种经筛选的、主动上送的信息或事后详细调用的信息进行综合管理，并以此为基础，利用一些工具进行信息处理，同时针对不同的应用需求提供相应的、经处理的信息服务。

2.4 电力系统故障分析及处理过程

2.4.1 故障分析过程

当地区电力系统发生断路器跳闸时，调度员对系统故障的分析过程一般如下：首先，根据断路器跳闸的情况，调出系统中相关厂站的单线图画面并对照模拟盘开关变位的情况，确定系统是否确已发生了事故。现场人员未做出汇报前，观察以下几项重要的故障现象：

(1) 频率是否有较大的波动，与跳闸线路相应的潮流是否为零或接近于零，相关厂站的母线电压是否变为零。

(2) 有时在系统发生事故时，相关厂站的远动数据可能会变“死”，这时可以通过临近厂站联络线潮流的是否突然增大，来判断系统是否确实发生了事故。

(3) 当系统出事故时，其临近厂站也会有冲击，这个冲击的大小以及范围的远近也可以作为确认故障大小以及事故恢复时的一个重要参考因素。

然后，当确认系统确已发生事故后，通过故障信息系统或者SCADA上的有关信息，来搜索出当前事故过程中，停电的有关设备，根据这些设备进行故障的初步判断。

其次，若事故过程中只有一个设备停电，则该设备就为故障设备。若有多项设备停电则进行下一步的分析。

最后，若有多项设备停电，则调度员根据保护设备配合的原理，分析每个设备停电的可能原因（如：是否符合主保护动作后开关跳闸的特征、后备保护动作跳闸的特征、失灵保护动作后开关跳闸的特征）。通过以上分析，来初步判断故障设备。

当现场人员检查保护动作信号，做出汇报后，根据保护动作的信号，最终确认故障点的具体位置。

2. 4. 2 故障处理的一般原则

在发生事故时，当值人员要迅速正确查明情况并快速做出记录，报告上级调度和有关负责人员，迅速正确地执行调度命令及运行负责人的指示，按照有关规程规定正确处理^[14]。

(1) 迅速限制事故发展，消除事故根源，并解除对人身和设备的威胁；

(2) 用一切可能的方法坚持设备继续运行，以保持对用户和线路的供电正常；

(3) 尽快对停电的用户和线路恢复供电。

发生事故时，只允许与事故处理有关的领导和工作人员留在控制室，其余人员应自觉离开，无关的工作班组暂停工作，离开现场。其余留在控制室的人员尽量保持肃静，以免妨碍值班人员处理事故。

处理事故时要保持头脑清醒，在当值运行值班长的统一领导下进行。处

理事故过程中，应当与上级调度保持紧密联系，随时执行调度的命令。当事故告一段落时，应迅速向有关领导汇报。事故处理完毕后应详细记录事故情况及处理过程，并保留所有电话录音备查。

变电所的技术人员应定期整理事故档案，并集中讨论事故处理步骤的正确与否，结合事故预想、反事故演习等培训工作，对职工进行安全教育，提高值班人员事故处理的素质。

2.5 本章小结

本章介绍了电力系统常见的几种故障情况，探讨了电力系统故障信息特征及故障诊断的数据信息获取方式，以及电力系统故障处理的原则与方法。

第3章 电力系统故障诊断的原理及方法

3.1 引言

电力系统是由生产、输送、分配和消费电能的发电机、变压器、电力线路和电力用户组成的整体，是将一次能源转换成电能并输送和分配至用户的一个统一系统。由于电力系统在国民经济中的重要地位，电力系统故障诊断原理和方法的研究一直受到人们的重视。

电力系统故障诊断就是根据设备运行状态查找故障源。虽然故障录波器能够记录开关和保护动作顺序以及各个电气量的波形，对分析事故很有用处，但是，目前依然多用于事故后离线分析。然而，随着电力系统调度自动化的发展，在调度中心可以得到完整的开关和保护动作信息，从而能够确保电力系统故障诊断的准确性，所以作为实时电力系统故障诊断，国内外通常都是基于保护和断路器信息进行诊断研究。

本章首先从理论上阐述电力系统继电保护的原理，随后在介绍电力系统故障诊断过程和方法过程中，引入电力系统故障诊断的数学描述，建立由保护和断路器信息构成的电力系统故障诊断模型，对电力系统故障后保护和断路器动作逻辑进行详细分析与总结。

3.2 继电保护原理

3.2.1 继电保护概念及特性

当电力系统中的电力元件（如发电机、线路等）或电力系统本身发生了故障，危及电力系统安全运行时，继电保护装置能够反应电气设备的不正常工作情况，并根据不正常工作情况和设备运行维护条件的不同（例如有无经常值班人员）发出信号，以便值班人员进行处理，或者直接向所控制的断路器发出跳闸命令以终止这些事件发展的一种自动化措施和设备，一般通称为继电保护装置^[15]。

因为继电保护在其发展过程中曾主要用有触点的继电器来保护电力系统及其元件（发电机、变压器、输电线路、母线等）使之免遭损害，所以称为

继电保护。

继电保护装置主要利用电力系统中元件发生短路或异常情况时的电气量（电流、电压、功率、频率等）的变化，构成继电保护动作的原理。具有选择性、灵敏性、快速性、可靠性等特点，这“四性”之间紧密联系，既矛盾又统一。

(1) 可靠性是指保护该动作时应可靠动作，不该动作时应可靠不动作。可靠性是对继电保护装置性能的最根本的要求。

(2) 选择性是指首先由故障设备或线路本身的保护切除故障，当故障设备或线路本身的保护或断路器拒动时，才允许由相邻设备保护、线路保护或断路器失灵保护切除故障。为保证对相邻设备和线路有配合要求的保护，或同一保护内有配合要求的两元件（如启动与跳闸元件或闭锁与动作元件）的选择性，其灵敏系数及动作时间，在一般情况下应相互配合。

(3) 灵敏性是指在设备或线路的被保护范围内发生金属性短路时，保护装置应具有必要的灵敏系数，各类保护的最小灵敏系数在规程中有具体规定。选择性和灵敏性的要求，通过继电保护的整定实现。

(4) 速动性是指保护装置应尽快地切除短路故障，其目的是提高系统稳定性，减轻故障设备和线路的损坏程度，缩小故障波及范围，提高自动重合闸、备用电源或备用设备自动投入的效果。一般从装设速动保护（如高频保护、差动保护）、充分发挥零序接地瞬时速断保护及相间速断保护的作用、减少继电器固有动作时间和开关跳闸时间等方面入手来提高速动性。

3.2.2 继电保护分类

继电保护可按以下4种方式分类^[16]:

(1) 按被保护对象分类：有输电线保护和主设备保护（如发电机、变压器、母线、电抗器、电容器等保护）。

(2) 按保护功能分类，有短路故障保护和异常运行保护。前者可分为主保护、后备保护和辅助保护；后者可分为过负荷保护、失磁保护、失步保护、低频保护、非全相运行保护等。

(3) 按保护装置进行比较和运算处理的信号量分类，有模拟式保护和数

字式保护。一切机电型、整流型、晶体管型和集成电路型（运算放大器）保护装置，直接反映输入信号的连续模拟量，均属模拟式保护；采用微处理机和微型计算机的保护装置，反映的是将模拟量采样和模/数转换后的离散数字量，这是数字式保护。

(4) 按保护动作原理分类，有过电流保护、低电压保护、过电压保护、功率方向保护、距离保护、差动保护、高频（载波）保护等。

(5) 按保护层次分类：主保护和后备保护。主保护是满足系统稳定和设备安全要求，能以最快速度有选择地切除被保护设备和线路故障的保护。

后备保护是主保护或断路器拒动时，用来切除故障的保护。后备保护可分为近后备保护（本文用第一后备保护）和远后备保护（本文用第二后备保护）两种，本文的主保护、第一后备保护和本文用第二后备保护仅仅是对设备发生短路、或者相临区域发生故障后，故障元件所配备的几套保护装置的动作顺序或者方式，是一种统计划分，不明确指明是电流保护、差动保护、过负荷保护、失磁保护等等，具体由系统实际配备保护装置情况确定。

第一后备保护是当主保护拒动时，由本电力设备或线路的另一套保护来实现后备的保护；当断路器拒动时，由断路器失灵保护来实现后备保护。第二后备保护是当主保护或断路器拒动时，由相邻电力设备或线路的保护来实现的后备保护。

3.3 电力系统故障诊断过程

电力系统故障诊断过程就是要从系统中确定有待诊断的故障区域，尽可能的缩小待诊断范围。然后，从被诊断区域的某些检测量中得到故障征兆信息，经过对这些征兆信息的分析处理，判断故障位置。

电力系统中通常只有少数元件同时发生故障，而且在事故发生之后比较短的时间之内保护将动作切断相应的断路器，将故障元件与非故障网络隔离以免事故的进一步扩大。因此，当故障平息后，故障元件将处于一个或几个孤立的子网络之中，通常利用断路器实时信息，采用实时结线分析方法来识别故障前后的系统拓扑结构，形成故障后所谓的故障区域。在形成了与故障元件相关的故障区域之后，故障元件的识别就可以局限于这些故障区域中所

包含的元件。

通常利用断路器的实时信息，采用实时结线分析方法来识别故障诊断前与故障平息后系统拓扑结构，之后找出这两个拓扑结构的差异，便可以识别出故障区域。对于一些简单故障，甚至可以直接识别出故障元件。

通过实时网络结线分析，将电力系统划分为一个个连通的子系统，这些子系统可以分为以下几种情况：

- (1) 含源无故障区域：包含有电源无故障区域。
- (2) 正常停运区域：故障发生前就停运的区域。
- (3) 被波及的区域：因上一级元件故障被切除而导致停电的区域。
- (4) 故障区域：包含有故障元件的区域。

根据故障前后开关变位的情况，最终排除前三种区域，将诊断区域缩小至故障区域内。

利用开关信息进行故障区域识别的步骤如下：

- (1) 根据变位开关的信息进行判断，如果故障前后区域内没有开关变位，则可以判断该区域为正常停运区域；
- (2) 在剩余的连通子系统中遍历每一个电源点，排除含源正常运行区域。至此可以得到因故障切除形成的各个孤立无源子系统。
- (3) 进一步利用保护动作信息与开关相互关联关系确定相邻上一级元件所波及的区域。

3.4 电力系统故障诊断模型

3.4.1 电力系统故障诊断的数学描述

如果将保护和断路器动作信息看作是由故障元件引起的结果，即故障是原因，继电保护动作信息与断路器跳闸信息是结果，那么整个过程可以抽象地看成一种状态经过一种变换，达到了另一种状态，可以用式(3-1)描述：

$$S_s = E(S_f) \quad (3-1)$$

其中， S_s ：保护和断路器动作信息状态集

S_f ：故障元件集

E：因果转换关系

对式(3-1)进行逆变换得到式(3-2)：

$$S_F = E^{-1}(S_S) \quad (3-2)$$

式(3-1)、式(3-2)描述的是两个集合之间的变换，从故障元件集到保护和断路器动作信息状态集之间的映射是多对一的变换关系。这种变换关系可以电力系统设备与保护配置信息、断路器关联信息中搜索得到，即这种变换关系是确定的、可预知的。

3.4.2 电力系统故障诊断模型的建立

诊断故障区域内发生故障的元件，就是要找出最能解释保护和断路器动作的信号的故障假说。而基于优化技术的故障诊断方法，基本思想就是根据保护动作原理将故障诊断问题表示为0-1整数规划问题，然后用优化方法求解。文献[2]将电网故障诊断问题表示为使如下目标函数式(3-3)最小化的问题。

$$E(S) = \sum_{k=1}^{n_r} |r_k - r_k^*(S)| + \sum_{j=1}^{n_c} |c_j - c_j^*(S, R)| \quad (3-3)$$

其中， n_r 为保护总数目， n_c 为断路器总数目。

S 为 n 维向量，表示系统中元件的状态(n 为系统中元件的数目)。 S 中的第*i*个元素 S_i 表示第*i*个元件的状态， $s_i=0$ 或1分别表示第*i*个元件的正常或故障状态；

R 为 n_r 维向量，表示 n_r 个保护的实际的状态。 R 中的第*k*个元素 R_k 表示第*k*个保护的实际状态： $r_k=0$ 或1分别表示第*k*个保护的未动作或动作状态；

$R^*(S)$ 为 n_r 维向量，表示 n_r 个保护的期望的状态。 $R^*(S)$ 中的第*k*个元素 $r_k^*(S)$ 表示第*k*个保护的期望的状态。如果第*k*个保护应该动作， $r_k^*(S)=1$ ，否则为0。 $R^*(S)$ 由 S 的状态决定；

C 为 n_c 维向量，表示 n_c 个断路器的实际状态。 C 中第*j*个元素 c_j 表示第*j*个断路器的实际状态， $c_j=0$ 或1分别表示第*j*个断路器的未跳闸或跳闸状态；

$C^*(S, R)$ 为 n_c 维向量，表示 n_c 个断路器的期望的状态。 $C^*(S, R)$ 中的第*j*个元素 $C_j^*(S, R)$ 表示第*j*个断路器的期望的状态。如果第*j*个断路器应该跳闸， $C_j^*(S, R)=1$ ，否则为0。 $C_j^*(S, R)$ 的状态由 S 和 R 的状态决定。

3.5 保护动作原理

电力系统故障诊断利用保护和断路器的动作信息识别发生故障的元件和误动作的保护与断路器，其理论依据就于保护动作逻辑，本文详细分析并总结了母线、线路、发电机和变压器等设备的保护动作原理：

(1) 母线故障时，母线的主保护应动作，跳开与该母线直接相连的所有断路器，将故障母线从线路中切除。由于在双侧电源网络中，考虑作为主保护的母线差动保护的主保护能达到保护的“四性”要求，所以参考文献[4]通常不设置母线的第一、二后备保护。如果母线与输电线路直接相连，当母线故障但母线主保护拒动时，由与其相连线路的第二后备保护动作跳开线路与母线相对一侧的断路器，防止故障蔓延；或者当母线与变压器直接相连时，由于变压器设备昂贵，出于保护变压器的目的，由与该故障母线相连的变压器第二后备动作跳开该母线和变压器之间的断路器。同样由于母线没有设置后备保护，如果与母线相连的设备发生故障且该设备主保护、第一后备保护拒动时，故障会从母线蔓延到与该母线直接相连的线路。

(2) 变压器发生故障时，变压器主保护应动作，跳开其两端的断路器，将发生故障的变压器从线路中切除；当主保护拒动时，对应的第一后备保护应动作跳开其两端的断路器切除发生故障的变压器；变压器的第二后备保护用于相邻区域发生故障，而该区域保护未动作时保护变压器^[18]。

(3) 线路主保护用于线路发生故障时跳开该线路两端的断路器，将发生故障的线路切除；当其主保护拒动时，对应的第一后备保护动作跳开其两端的断路器切除该线路；线路的第二后备保护亦用于相邻区域发生故障，且该区域保护未动作，为了防止故障沿线路蔓延，该保护动作跳开故障区域对侧断路器^[35]。

(4) 发电机发生故障时，发电机主保护应动作，跳开发电机出口断路器，将发生故障的发电机从线路中切除；当主保护拒动时，对应的第一后备保护应动作跳开发电机出口断路器，切除发生故障的发电机；发电机的第二后备保护用于相邻区域发生故障，而该区域保护未动作或者断路器拒动时保护发电机。

一般情况下故障范围不会蔓延到两个厂站（变电站）外，这样，对于变

压器，我们只在它所在的变电站内进行故障搜索。同时，以上这种各类保护动作分析仅仅是一种理想的或预期（或最优）情况，实际的电力系统未必按此设定的逻辑工作，除了可能出现该动而未动的情况外，还可能出现不该动而误动的情况。

3.6 电力系统故障诊断目标函数的形成

为了形成电力系统故障诊断目标函数，必须对电力系统的线路系统进行处理，现用如图3.1所示的线路系统做进一步的解释。

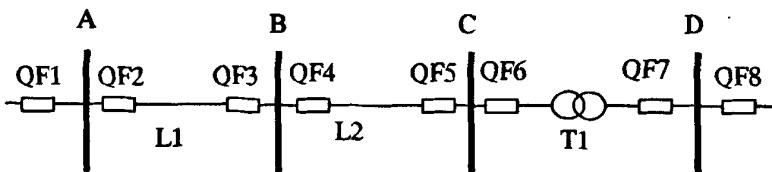


图 3.1 保护动作原理实例图

系统中共有7个元件，即母线A、B、C、D，线路L1、L2，变压器T1，在此依次用 S_1 、 S_2 、 S_3 、 S_4 、 S_5 、 S_6 、 S_7 表示。

8个断路器QF1、QF2、QF3、QF4、QF5、QF6、QF7、QF8依次用 C_1 、 C_2 、 C_3 、 C_4 、 C_5 、 C_6 、 C_7 、 C_8 表示。

表 3.1 主保护动作原理

| 保护名 | 动作原理 |
|--------|-------------------------------|
| A_m | A发生故障时， A_m 动作跳开QF1和QF2 |
| B_m | B发生故障时， B_m 动作跳开QF3和QF4 |
| C_m | C发生故障时， C_m 动作跳开QF5和QF6 |
| D_m | D发生故障时， D_m 动作跳开QF7和QF8 |
| $T1m$ | $T1$ 发生故障时， $T1m$ 动作跳开QF6和QF7 |
| $L1Am$ | $L1$ 发生故障时， $L1Am$ 动作跳开QF2 |
| $L1Bm$ | $L1$ 发生故障时， $L1Bm$ 动作跳开QF3 |
| $L2Bm$ | $L2$ 发生故障时， $L2Bm$ 动作跳开QF4 |
| $L2Cm$ | $L2$ 发生故障时， $L2Cm$ 动作跳开QF5 |

19个保护，即 A_m 、 B_m 、 C_m 、 D_m 、 $T1m$ 、 $L1Am$ 、 $L1Bm$ 、 $L2Bm$ 、 $L2Cm$ 、 $L1Ap$ 、 $L1Bp$ 、 $L2Bp$ 、 $L2Cp$ 、 $T1p$ 、 $L1As$ 、 $L1Bs$ 、 $L2Bs$ 、 $L2Cs$ 、 $T1s$ ，依次用 r_1 、 r_2 、 r_3 、 r_4 、 r_5 、 r_6 、 r_7 、 r_8 、 r_9 、 r_{10} 、 r_{11} 、 r_{12} 、 r_{13} 、 r_{14} 、 r_{15} 、 r_{16} 、 r_{17} 、 r_{18} 、 r_{19} 表示。其中m表示主保护、p表示第一后备保护，s表示第二后备保护。

对于图3.1所示的系统，其主保护、第一后备保护以及第二后备保护的动作原理列于表3.1、表3.2表和表3.3中：

表 3.2 第一后备保护动作原理

| 保护名 | 动作原理 |
|------|---------------------------------------------------|
| L1Ap | L1 故障时, QF2 在主保护动作时段未跳开时, L1Ap 动作跳开 QF2 |
| L1Bp | L1 故障时, QF3 在主保护动作时段未跳开时, L1Bp 动作跳开 QF3 |
| L2Bp | L2 故障时, QF4 在主保护动作时段未跳开时, L2Bp 动作跳开 QF4 |
| L2Cp | L2 故障时, QF5 在主保护动作时段未跳开时, L2Cp 动作跳开 QF5 |
| T1p | T1 故障时, QF6 和 QF7 在主保护动作时段未跳开, T1p 动作跳开 QF6 和 QF7 |

为了形成实际优化的具体目标函数公式。文中引入“关联元件”和“关联路径”概念。这里的“关联元件”意义较广，泛指有联系的元件与保护、保护与断路器、元件与元件间的关系^[35]。

“元件——保护”间的关联，指该保护能划为该元件的主保护、第一后备保护、第二后备保护中的一种；

“保护——断路器”间的关联，指一旦该保护动作，原理上能使这个断路器动作；

“元件1——元件2”间的关联，指元件1的第二后备保护范围能够包括元件2，同时称元件2为元件1的关联元件。例如L1As能保护到B和L2，称L1的关联元件为B和L2。

“关联路径”指从系统结线图上来看，一个元件到与其关联的另一个元件，经过的一个或多个断路器的排列。如L1和L2为关联元件，L2是通过QF3和QF4连到L1的，故L1到L2的关联路径为QF3和QF4。为便于描述，表3.4中给出的例子中的关联元件表。

结合保护动作原理及关联分析，形成如下实际优化的具体目标函数公式。

图中共7个元件（A、B、C、D、L1、L2、T1）依次编号为(s₁~s₇），8个断路器（QF1~QF8）依次编号为(c₁~c₈），19个保护（Am、Bm、Cm、Dm、L1Am、L1Bm、L2Bm、L2Cm、T1m、L1Ap、L1Bp、L2Bp、L2Cp、

T1p、L1As、L1Bs、L2Bs、L2Cs、T1s) 依次编号为 (r₁~r₁₉)

表 3.3 第二后备保护动作原理

| 保护名 | 动作原理 |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L1As | B 故障且 QF3 在 B 的主保护、第一后备保护动作时段未跳开，或者 L2 故障且 QF4 在 L2 的主保护、第一后备保护动作时段未跳开，此时为防止故障沿 L1 蔓延，由 L1As 动作跳开 QF2。 |
| L1Bs | A 发生故障时，QF2 在 A 的主保护、第一后备保护动作时段未跳开，此时为防止故障沿 L1 蔓延至 B，由 L1Bs 动作跳开 QF3。 |
| L2Bs | C 发生故障且 QF5 在 C 的主保护、第一后备保护动作时段未跳开，或者 T1 故障且 QF6 在 T1 的主保护、第一后备保护动作时段未跳开，此时为防止故障沿 L2 蔓延至 B，由 L1Bs 动作跳开 QF4。 |
| L2Cs | B 发生故障且 QF4 在 B 的主保护、第一后备保护动作时段未跳开，或者 L1 故障且 QF3 在 L1 的主保护、第一后备保护动作时段未跳开，此时为防止故障沿 L2 蔓延，由 L2Cs 动作跳开 QF5。 |
| T1s | C 发生故障且 QF6 在 C 的主保护、第一后备保护动作时段未跳开，或者 L2 故障且 QF5 在 L2 的主保护、第一后备保护动作时段未跳开，此时出于保护变压器的目的，T1s 动作跳开 QF6；再者当 D 故障且 QF7 在 D 的主保护、第一后备保护动作时段未跳开，此时出于保护变压器的目的 T1s 动作跳开 QF7。 |

表 3.4 关联元件及关联路径综合表

| 元件名称 | 保护名 | 关联方向/故障方向 | 关联元件 | 跳开的断路器 | 关联路径 |
|------|------|-----------|------|--------|---------|
| L1 | L1Bs | L1 左侧 | A | QF3 | QF2 |
| | L1As | L1 右侧 | L2 | QF2 | QF3、QF4 |
| | | | B | | QF3 |
| L2 | L2Cs | L2 左侧 | L1 | QF5 | QF3、QF4 |
| | | | B | | QF4 |
| | L2Bs | L2 右侧 | C | QF4 | QF5 |
| | | | T1 | | QF5、QF6 |
| T1 | T1s | T1 左侧 | C | QF6 | QF6 |
| | | | L2 | | QF5、QF6 |
| | | T1 右侧 | D | QF7 | QF7 |

(1) 主保护的期望状态 = 被诊断元件的实际状态, 如式 (3-2) 所示。

$$\left[\begin{array}{l} r_1^*(S) = s_1 \\ r_2^*(S) = s_2 \\ r_3^*(S) = s_3 \\ r_4^*(S) = s_4 \\ r_5^*(S) = s_5 \\ r_6^*(S) = s_5 \\ r_7^*(S) = s_6 \\ r_8^*(S) = s_6 \\ r_9^*(S) = s_7 \end{array} \right] \quad (3-2)$$

(2) 第一后备保护的期望状态反映“主保护未动作, 第一后备应该动作”这一动作原理, 即第一后备保护期望状态=被保护元件状态×(1—相应主保护状态), 如式 (3-3) 所示。

$$\left[\begin{array}{l} r_{10}^*(S) = s_5(1-r_5) \\ r_{11}^*(S) = s_5(1-r_6) \\ r_{12}^*(S) = s_6(1-r_7) \\ r_{13}^*(S) = s_6(1-r_8) \\ r_{14}^*(S) = s_7(1-r_9) \end{array} \right] \quad (3-3)$$

(3) 第二后备保护的期望状态反映只在关联元件的主保护和第一后备保护都拒动时才能够动作这一特性。在原理实现上, 可以表示为第二后备保护期望状态=1-[1-关联元件1的状态值×(1-关联路径上断路器1的状态值)×(1-关联路径上断路器2的状态值)×…]×[1-关联元件2的状态值×(1-关联路径上断路器1的状态值)×(1-关联路径上断路器2的状态值)×…]×[1-关联元件3的状态值×(1-关联路径上断路器1的状态值)×…]×…, 如式 (3-4) 所示:

$$\left. \begin{array}{l} r_{15}^*(S) = 1 - [1 - s_2(1 - c_3)][1 - s_6(1 - c_3)(1 - c_4)] \\ r_{16}^*(S) = 1 - [1 - s_1(1 - c_2)] \\ r_{17}^*(S) = 1 - [1 - s_3(1 - c_5)][1 - s_7(1 - c_5)(1 - c_6)] \\ r_{18}^*(S) = 1 - [1 - s_2(1 - c_4)][1 - s_5(1 - c_3)(1 - c_4)] \\ r_{19}^*(S) = 1 - [1 - s_3(1 - c_6)][1 - s_6(1 - c_5)(1 - c_6)][1 - s_4(1 - c_7)] \end{array} \right\} \quad (3-4)$$

(4)由保护触发的断路器的期望状态与所有能够激发该断路器的保护有关, 等于这些保护的期望状态函数值乘以该保护实际状态值中所得的最大值。即断路器期望值= $\max\{\text{与该断路器相关的保护的期望状态函数值} \times \text{该保护实际状态值}\}=\max\{(\text{关联主保护1的状态值} \times \text{相应元件的状态值}), (\text{关联主保护2的状态值} \times \text{相应元件的状态值}) \dots, (\text{关联第一后备保护1的状态值} \times \text{相应元件的状态值} \times (1-\text{相应主保护的状态值})), (\text{关联第一后备保护2的状态值} \times \text{相应元件的状态值} \times (1-\text{相应的主保护的状态值})) \dots, (\text{关联第二后备保护1的状态值} \times [1-\text{直接相连元件状态值} \times (1-\text{关联断路器的状态值})] \times [1-\text{间接相连元件状态值} \times (1-\text{关联断路器1的状态值}) \times (1-\text{关联断路器2的状态值})]) \dots\}$, 如式(3-5)所示:

$$\left. \begin{aligned} c_1^*(S, R) &= r_1^* r_1 \\ c_2^*(S, R) &= \max \{r_1^* r_1, r_5^* r_5, r_{10}^* r_{10}, r_{15}^* r_{15}\} \\ c_3^*(S, R) &= \max \{r_2^* r_2, r_6^* r_6, r_{11}^* r_{11}, r_{16}^* r_{16}\} \\ c_4^*(S, R) &= \max \{r_2^* r_2, r_7^* r_7, r_{12}^* r_{12}, r_{17}^* r_{17}\} \\ c_5^*(S, R) &= \max \{r_3^* r_3, r_8^* r_8, r_{13}^* r_{13}, r_{18}^* r_{18}\} \\ c_6^*(S, R) &= \max \{r_3^* r_3, r_9^* r_9, r_{14}^* r_{14}, r_{19}^* r_{19}\} \\ c_7^*(S, R) &= \max \{r_4^* r_4, r_9^* r_9, r_{14}^* r_{14}, r_{19}^* r_{19}\} \\ c_8^*(S, R) &= r_4^* r_4 \end{aligned} \right\} \quad (3-5)$$

至此, 所有保护和断路器的期望状态都可以得到, 根据得到的保护实际状态R和断路器实际状态C, 带入目标函数, 则E(s)可以表示为故障区域所有元件状态S的函数, 故障诊断的目的就是确定S的元素, 使目标函数取得最小值。

3.7 本章小结

本章在简单介绍继电保护原理的基础上, 阐述了电力系统故障诊断过程和方法, 给出了基于保护和断路器动作信息进行故障元件识别的的电力系统

故障诊断模型，并通过实例给出诊断模型所基于的保护动作原理和诊断目标函数的建立过程。

第4章 基于遗传算法故障诊断的实现

4.1 引言

电力系统故障诊断模型中，故障区域内待诊断元件存在正常和故障两种状态，所以能构成0-1整数规划求解问题^[12]。0-1整数规划问题是全部决策变量都必须取整数值0/1。如上一章所给诊断模型，每个待诊断元件都用一个变量表示（ $s_1 \sim s_7$ ），变量取0表示元件正常，取1表示设备发生故障。由于变量只取这两个值，所以人们很自然会想到用穷举法来解上一章中形成的目标函数，即排出全部7个变量取值为0或1的每一种组合共 2^7 个，通过上一章形成的目标函数求出这 2^7 个组合所对应的保护和断路器实际状态值与期望状态值之间的偏差，找出最大偏差和最小偏差，以此找到最优解，判定故障元件。

但是，当故障区域内待诊断的元件数量n很大时，计算量是相当大的，即要进行 2^n 次目标函数值的计算（例如，当n=20，则要计算 $2^{20}=1048576$ 个组合），才能求得最优解，此时不能满足电力系统故障诊断的快速性要求，难以在尽可能短的时间内判定故障元件，控制故障范围，减少损失以及为事故抢修争取时间。

对于0-1整数规划问题，传统算法采用分枝定界的方式。分枝定界法是一种隐性枚举法，虽然对枚举法有所改进，但终究不是一种有效的算法。而且传统的0-1整数规划问题求解方法，是一种确定性算法，其得到的结果是一个唯一的最优解，但实际问题中常存在多个最优解的情况。因此随着电力系统实际问题规模的增大和复杂度的提高，状态决策变量的数量随之增多，计算时间将极大增加，不能适应电力系统故障诊断实时性的要求，传统的求解方法不能奏效，很自然会寻求优化方法求解。其中遗传算法（GA）就是一种比较成熟且应用最广泛的进化优化算法，可以快速求解0-1整数规划问题，并进而给出诊断结果。

遗传算法（GA）是建立在Darwin自然选择和Mendel遗传学说基础上，通过模仿生物遗传和进化的进程，寻求对复杂问题的全局最优解的优化算法，同时以全局并行搜索优化群体中的最优个体，以求得满足要求的最优解。而且GA也不需要直接对知识规则和训练样本选择处理，这是它和基于ES、ANN

诊断系统相比的最大优势之处^[13]。GA从优化的角度解决故障诊断问题，它能够在诊断信息不完整的情况下给出全局最优或局部最优的多个可能的诊断结果。所以本文以GA算法辅助进行电力系统故障诊断。

4.2 遗传算法

4.2.1 遗传算法的发展

早在20世纪50年代，自然界生物进化的理论已被广泛接受。作为生物学家的Fraser，试图通过计算的方法来模拟生物界“遗传与选择”的进化过程，这便是遗传算法的最早雏形。直到20世纪60年代，Holland教授在他的著作《自然系统与人工系统中的适应性》中，首次明确提出遗传算法的概念。Holland创建的遗传算法，是基于二进制表达的概率搜索方法。在种群中，通过信息交换重新组合新串，根据评价条件，概率选择适应性好的串进入下一代，经过多代进化，种群最后稳定在适应性好的串上。Holland最初提出的遗传算法被认为是简单遗传算法的基础，也称为标准遗传算法。

遗传算法作为一种计算方法，对求解的问题本身一无所知，Holland最初研究的遗传算法也并不是为解决特定的问题而设计的。由于遗传算法的特点是种群搜索和种群中个体之间交换信息，搜索不依赖梯度信息和问题本身，具有鲁棒性和全局搜索能力，常常应用于优化和并行处理领域。

4.2.2 遗传算法的应用及特点

遗传算法是一种宏观意义上的仿生算法，它模仿的机制是一切生命与智能的产生与进化过程。它通过二进制字符串模拟人工染色体来表示某一优化问题的可行解，用随机方法产生一个可行解的集合，按照自然选择的原理，即群体中的人工染色体的适应值越高则它获得繁殖后代的机会越大，运用选择、交叉、变异等各种GA算子模拟进化，使整个群体不断优化并最终找到问题的全局最优解。作为一种随机的优化搜索方法，其有着鲜明的特点：

(1) 遗传算法的处理对象不是参数本身，而是对参数集进行了编码的个体。此操作使得遗传算法可直接对结构对象进行操作。

(2) 许多传统搜索算法都是单点搜索算法，容易陷入局部的最优解。遗

传算法同时处理群体中的多个个体，即对搜索空间中的多个解进行评估，减少了陷入局部最优解的风险，同时算法本身易于实现并行化。

(3) 遗传算法基本上不用搜索空间的知识或其它辅助信息，而仅用适应度函数值来评估个体，在此基础上进行遗传操作。适应度函数不仅不受连续可微的约束，而且其定义域可以任意设定。这一特点使得遗传算法的应用范围大大扩展。

(4) 遗传算法不是采用确定性规则，而是采用概率的变迁规则来指导它的搜索方向。

(5) 具有自组织、自适应和自学习性。遗传算法利用进化过程获得的信息自行组织搜索，适应度大的个体具有较高的生存概率，并获得更适应环境的基因结构。

4.2.3 遗传算法的步骤

遗传算法求解问题时，从初始化的种群出发，以适应度函数为依据，通过对群体的个体不断施加选择（使群体中的优良特性有更多的机会遗传至下一代）、交叉（体现了下一代自然界中群体内个体之间的信息交换）、变异（在群体中引入新的品种以确保群体中信息的多样性）等遗传操作，实现个体之间的信息交换和结构重组，使群体品质不断地得以提高并逐渐逼近全局最优解。标准遗传算法流程如图4.1所示。

(1) 编码

遗传算法求解问题时，首先确定问题的目标函数和变量，并对变量进行编码。由于遗传操作算子在迭代过程中直接对串进行操作，因此编码的策略对遗传操作，尤其是对交叉操作的功能有很大的影响。

编码方式可以分为二进制编码和实数编码，即十进制编码。

采用二进制编码方式，其优点在于它与计算机码制相一致，适于计算机应用，编码和解码操作简单，交叉、变异等操作便于实现；表示的变量范围大，适合表示离散变量，对于连续变量制要群体总数取的足够多，就能满足精度。十进制编码表示问题解的数字串会比用二进制表示的数字串短，相应的遗传操作算子的计算量也会减少，计算时间也相对减少。具体的编码方式

要根据实际优化问题确定，考虑变量编码表示方便及遗传操作简便等问题。

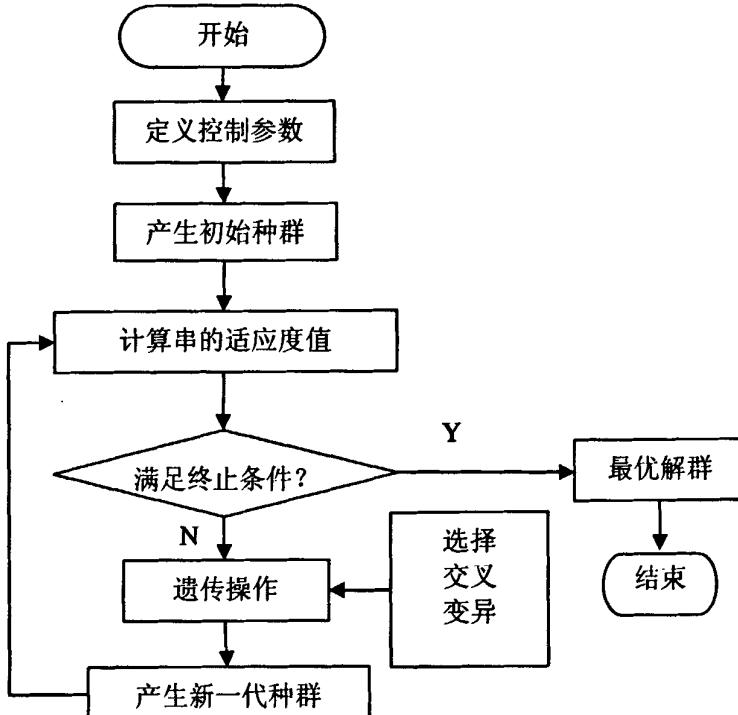


图 4.1 标准遗传算法流程图

(2) 形成初始种群

遗传算法是一种群体型的优化搜索方法，遗传操作是针对多个个体同时进行的，这些个体构成了群体。在编码设计后的任务就是初始种群的设定，并以其作为遗传计算的起点。遗传算法中通常使用随机发生的方法来产生初始种群。因为在优化计算之前，问题的最优解到底位于解空间何处是不得而知的，因此从原理上讲，这一作法可以使初始种群均匀分布于整个解空间，有利于扩大初始种群的覆盖范围。但如此产生的初始种群的总体品质往往较低，获得最优个体可能需要较长的搜索时间，因而经常是先随机生成一定数目的个体，然后从中选择最好的个体加入初始群体。如此反复迭代，直到初始种群中的个体数目达到预定的规模。

作为遗传算法的控制参数之一，种群规模的大小和交叉概率、变异概率一样，对遗传算法效能的发挥有很大影响。从原理上讲，群体规模越大，群体中个体的多样性程度也越高，算法陷于局部最优的风险也就越小。所以，从考虑群体的多样性出发，群体规模越大越好。但是群体规模太大也会带来

弊端：一是从计算效率着眼，群体规模越大，其适应度函数评估的次数也越多，计算效率受到影响；二是群体中个体的生存概率大多采用和适应度值成比例的方法来确定，当群体中个体过多时，往往只有少量适应度很高的个体被选择而生存下来，大多数个体却被淘汰，交叉操作开辟新的搜索区域性能受到影响。另一方面，群体规模太小，会使初始种群在搜索空间中的分布范围受到限制，因而有可能会导致遗传搜索停止在未成熟阶段，即发生未成熟收敛现象。实际应用中群体中个体数目的取值范围一般是几十到几百。

(3) 适应度函数

遗传在进化搜索中基本不利用外部信息，仅以适应度函数为依据，利用种群中每个个体的适应度值来进行搜索。因此适应度函数的选择至关重要，直接影响到遗传算法的收敛速度以及能否找到最优解。一般而言，适应度函数是由目标函数变换而成的。

若目标函数为最小化问题，则适应度函数可以表示为式(4-1)的形式。

$$Fit[f(x)] = \begin{cases} c_{\max} - f(x) & f(x) < c_{\max} \\ 0 & \text{其他} \end{cases} \quad (4-1)$$

(4) 遗传操作

遗传操作是模拟生物进化基因的操作，它根据个体适应值对染色体施加一定的操作，从而实现优胜劣汰的进化过程。从优化搜索角度而言，遗传操作可使得问题的解逐代优化，逼近最优解。

遗传操作包括三个基本的遗传算子：选择、交叉和变异。选择和交叉基本上完成了遗传算法的大部分功能，变异增强了遗传算法搜寻到接近最优解的能力。

选择操作是指从群体中选择优良个体并淘汰劣质个体的操作。从种群中选择性能优良的个体，参加下一代的繁殖，将其优良性能通过复制遗传到下一代。它建立在适应度评估的基础上，适应度越大的个体，被选择的可能性就越大，而适应度较小的个体被淘汰。被选中的个体被放入配对库中，随机进行配对，进行交叉操作。

简单的选择方法为轮盘赌法：通常以第*i*个个体入选种群的概率以及群体规模的上限来确定其生存与淘汰，这种方法称为轮盘赌法。轮盘赌法是一种

正比选择策略，能够根据与适应函数值成正比的概率选出新的种群。轮盘赌法由以下五步构成：

首先计算各染色体适应函数值 $f(x)$ ，然后计算种群中所有染色体的适应函数值的和，接下来计算各染色体的选择概率 $P(x_i)$ ，随后计算各染色体的累积概率，并产生一随机数，该随机数落入轮盘的哪个区域，就选择相应个体进行繁殖。

交叉操作模拟生物进化过程中的繁殖现象，通过两个个体的交换组合来创造新的优良个体。选择操作每次仅作用在一个个体上，而交叉操作每次作用在从配对库中随机选取的两个个体上。交叉操作产生两个子代个体，它们一般与其父代个体不同，并且彼此也不同，每个子代个体都包含两个父代个体的遗传物质。交叉操作分为单点交叉、多点交叉和均匀交叉等。

交叉操作分两步进行，首先按照某种方法，随机地从配对库中取出要交叉的一对个体，然后进行交叉，产生一对新个体。交叉的方法是，先根据个体数字串长度L，随机产生一个交叉位置，即[1, L-1]区间上的一个整数，后在这个位置上，将双亲的基因码链截断，最后互换尾部。

变异操作就是以变异概率 P_m 随机地改变群体中个体的某些基因值。遗传算法中使用变异操作具有局部的随机搜索能力，而且可以使遗传算法维持群体的多样性。变异操作一般在交换后进行。变异操作的对象是个体，旨在改变个体中的某些位的值。对于二进制编码的个体，其编码字符集为(0, 1)，变异操作就是将个体在变异点上的基因值取反，即由0变为1，或由1变为0。

4.3 基于遗传算法的电力系统故障诊断

电力系统故障诊断算法要依靠软件来实现，而且随着软件技术的发展，面对对象程序设计方法几乎成为了成功、先进、效率的代名词。使用面对对象的方法设计和实现一个软件几乎成为了开发者们的默认选择。

4.3.1 基于遗传算法的电力系统故障诊断操作设计

(1) 电力系统故障诊断被转化为0/1整数规划问题，设备状态变量用0/1表示正常/故障两种状态、保护和断路器状态变量也都被定义为0和1两个状态

值，所以对于电力系统故障诊断来说，适宜采用二进制进行编码。

(2) 适应度函数

电力系统故障诊断目标函数是式(4-1)所示的最小化问题，本文为了便于程序设计，将其转化为式(4-3)所示的适应度函数。

$$f(s) = 103 - \left\{ \sum_{k=1}^{n_k} |R_k - R_k^*(S)| + \sum_{j=1}^{n_j} |C_j - C_j^*(S, R)| \right\} \quad (4-2)$$

(3) 遗传算子的选择

选择算子采用轮盘赌法，由随机产生的(0, 1)之间的数作为选择指针，个体适应度除以累计适应度为选择概率。

交叉作为遗传算法中的最重要算子，寻优的搜索过程主要是通过交叉算子实现的，本文采用了单点交叉方式，交叉概率Pc=0.9。

变异可以使在运算过程中丢弃的某位重要数字予以恢复，但是变异算子应该取的比较小，一般应该在0.5以下，本文的变异概率Pm=0.01。

(4) 终止条件

根据故障区域中待诊断元件的数自动态设定最大允许迭代次数作为终止条件，一般选择10~20代。

4.3.2 基于遗传算法的电力系统故障诊断流程

(1) 从故障诊断图形数据库中读入系统数据，包括每个设备的运行状态和报道的警报信息(即保护动作和断路器跳闸的信息)。

(2) 识别停电(故障)区域。

(3) 识别故障区域中所包含的元件、保护及断路器，通过关系数据库内部设计的关联关系，确定元件、保护和断路器间的关联，以此建立故障区域适应度函数。

(4) 应用GA求解使适应度最大化的问题，直到达到给定的最大允许迭代次数。

4.4 电力系统故障诊断遗传算法程序设计

4.4.1 面向对象理论

对象（Object）是问题域或实现域中某些事物的一个抽象，能够反映此事物在系统中需要保存的信息和发挥的作用；是一组属性和有权对这些属性进行操作的一组服务的封装体。为对象分配职责、对象间如何协作、什么对象做什么事情，是系统设计的关键问题。

（1）面向对象分析

面向对象分析方法（Object-Oriented Analysis, OOA），是在一个系统的开发过程中进行了系统业务调查以后，按照面向对象的思想来分析问题。在问题领域内发现或描述对象或概念。主要基于抽象、封装、继承、分类、聚合、关联、通信和行为分析等原则，经过确定对象和类、结构、主题、属性和方法等步骤产生用例模型、静态模型和动态模型。

（2）面向对象设计

面向对象设计（Object-Oriented Design, OOD）方法是OO方法中一个中间过渡环节，在软件设计生命周期中发生于OOA之后期或者之后，其主要作用是对OOA分析的结果作进一步的规范化整理，以便能够被OOP直接接受。OOD是一种软件设计方法，是一种工程化的规范。设计过程中要根据面向对象的概念模型整理分析所确定的对象结构、属性、方法等内容，改正错误的内容，删去不必要和重复的内容等。并且进行分类整理，以便于下一步数据库设计和程序处理模块设计的需要。同时根据需求决定所需的类、类的操作以及类之间关联的过程。

（3）面向对象编程

面向对象编程（Object Oriented Programming, OOP，面向对象程序设计）是一种计算机编程架构。OOP的基本原则是计算机程序是由单个能够起到子程序作用的单元或对象组合而成。OOP达到了软件工程的三个主要目标：重用性、灵活性和扩展性。为了实现整体运算，每个对象都能够接收信息、处理数据和向其它对象发送信息。OOP主要有以下的概念：

抽象性：程序有能力忽略正在处理中信息的某些方面，即对信息主要方

面关注的能力。

封装：也叫做信息封装。确保组件不会以不可预期的方式改变其它组件的内部状态；只有在那些提供了内部状态改变方法的组件中，才可以访问其内部状态。每类组件都提供了一个与其它组件联系的接口，并规定了其它组件进行调用的方法。

多态性：组件的引用和类集会涉及到其它许多不同类型的组件，而且引用组件所产生的结果得依据实际调用的类型。

继承性：允许在现存的组件基础上创建子类组件，这统一并增强了多态性和封装性。典型地来说就是用类来对组件进行分组，而且还可以定义新类为现存的类的扩展，这样就可以将类组织成树形或网状结构，这体现了动作的通用性。

(4) UML及Rose

统一建模语言（UML）是通用的可视化建模语言，是“面向对象思想”的图形化展示，用来对软件进行描述、构造、可视化和文档编制。UML溶入了软件工程领域的思想、新方法和新技术，获得了工业界、科技界和应用界的广泛支持。融合以往建模技术Booch、OMT和OOSE的经验，UML成为当今最优秀的标准建模方法展示方式。UML通过三类图形描述系统模型（事物模型和关系模型）：用例（Use Case）图、静态结构图（对象类图、对象图、组件图、配置图）和动态行为图（顺序图、协同图、状态图、活动图）。

UML的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。在需求分析阶段，可以用用例来捕获用户需求。通过用例建模，描述对系统感兴趣的外部角色及其对系统（用例）的功能要求。分析阶段主要关心问题域中的主要概念（如抽象、类和对象等）和机制，需要识别这些类以及它们相互间的关系，并用UML类图来描述。为实现用例，类之间需要协作，这可以用UML动态模型来描述。在分析阶段，只对问题域的对象（现实世界的概念）建模，而不考虑定义软件系统中技术细节的类（如处理用户接口、数据库、通讯和并行性等问题的类）。

Rational Rose是IBM公司出品（Rational公司已经被IBM公司收购）的面向对象统一建模语言与软件开发集成环境，它提供了Java, C++, Visual Basic, ADA等等诸多软件的软件代码双向生成环境。

4.4.2 基于面向对象方法的遗传算法类设计

(1) 电力系统故障诊断遗传算法流程

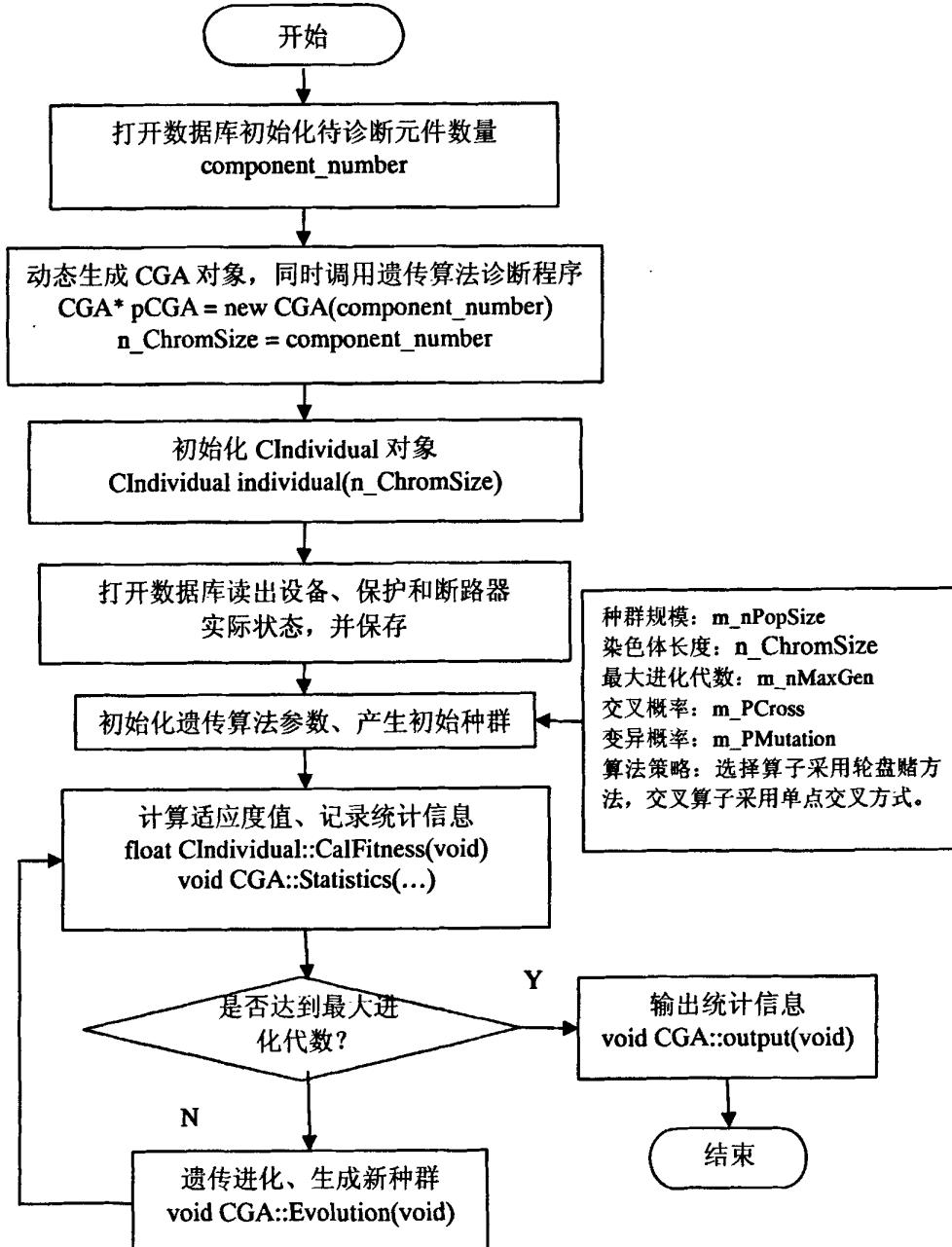


图 4.2 电力系统故障诊断遗传算法软件流程图

电力系统故障诊断遗传算法软件流程如图4.2所示。首先通过与数据库的

连接获取数据库待诊断元件数量用以初始化染色体基因序列的长度。随后动态生成遗传算法类对象进行电力系统故障诊断。

本程序的特点是充分利用了面向对象程序设计的原理，当动态生成CGA类对象的时候会自动调用CGA类构造函数进行初始化准备，本文借此原则在CGA类的构造函数中完成所有遗传算法初始化工作，包括初始化染色体对象、从数据库读取待诊断区域元件、保护和设备的实际状态、产生初始种群、计算初始种群的适应度值和种群统计信息。

随后CGA对象将进入遗传进化、产生新种群的迭代循环，当迭代次数达到设定值，程序会调用CGA对象的输出操作在软件输出窗口展示诊断结果。

(2) 故障诊断遗传算法Use Case图

所谓Use case是指系统的外部事物（活动者）与系统的交互，它表达了系统的功能，即系统所提供的服务。具体的说，Use case是关于系统的一组动作的说明，这些动作对一个或多个活动者给出所需要的结果。Use case图是一种描述Use case的可视化工具，是后续的工作，如交互图等的依据，也是系统测试的依据^[38]。

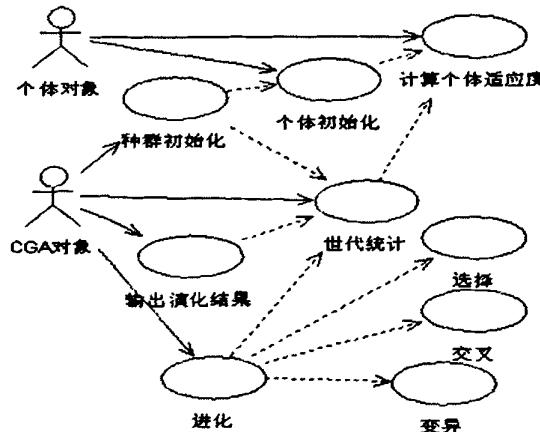


图 4.3 遗传算法模型的 Use case 图

如图4.3所示为遗传算法模型的Use case图，系统考虑的范围包括具体的遗传算法和染色体单体对象。活动者有两个，“故障诊断遗传算法CGA对象”，调用类库及个体对象为使用者提供服务。Use case有九个：“种群初始化”、“个体初始化”、“计算适应度”、“世代统计”、“进化”、“选择”、“交叉”、“变异”、“输出演化结果”。故障诊断系统通过直接或间接使

用这九个Use case完成遗传算法的演化运行。

“种群初始化”完成种群规模确定及初始代统计计算，并通过调用“个体初始化”设置染色体基因序列长度，计算适应度值；“进化”完成种群演化，它将依次调用“选择”、“交叉”、“变异”、“世代统计”；“输出演化结果”用于把当前群体的结果输出到指定格式的外部变量之中；“统计”完成对当前群体中目标函数值最大、最小等演化数据的统计。

其中“种群初始化”与“个体初始化”、“世代统计”，“个体初始化”与“计算个体适应度”，“进化”与“世代统计”、“选择”、“交叉”、“变异”，“世代统计”与“计算个体适应度”之间存在着使用关联。

(3) 故障诊断遗传算法对象类图

对象类简称类，是面向对象模型的最基本的模型元素，是对象的集合。对象类图表达一组类和它们之间的联系。在对象类图中，一方面描述各个类本身的组成，即类的属性、操作和对象的约束；另一方面描述系统中类之间的各种静态的联系。

如图4.4所示为故障诊断遗传算法的对象类图。矩形框“CGA”和“CIndividual”分别表示算法对象和个体对象。图中给出了对象的成员变量、方法，以及“CGA”类对“CIndividual”类的使用关系。

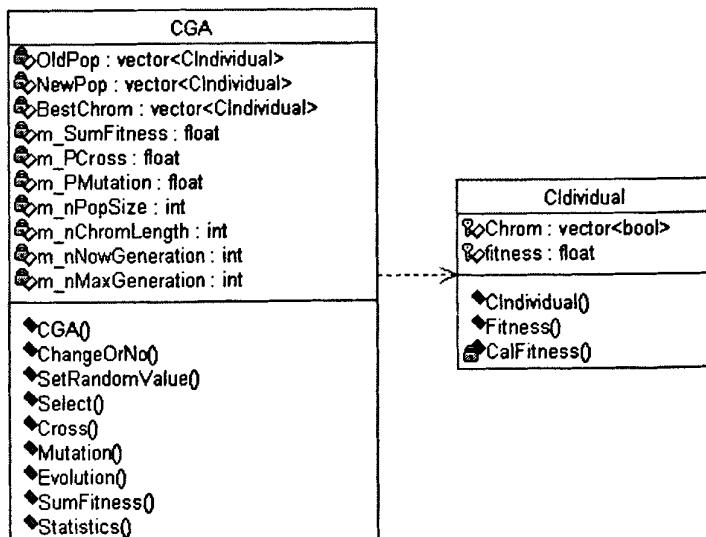


图 4.4 故障诊断遗传算法对象类图

4.4.3 基于标准模板库 STL 的遗传算法程序开发

本文开发的遗传算法基于C++ STL所提供的泛型数据结构，程序明显提高了程序可读性，简化了程序设计，将文献[18]的复杂且烦琐达300多条语句的源程序，缩短控制在100条语句内实现。

C++ STL通过模板技术，将常用的数据结构和算法，纳入到编译器的语言支持库中，极大地改善了程序的设计环境，不用为不同数据类型的数据结构和算法进行重复设计，C++ STL泛型库是由容器、迭代器、算法、内存分配器等部分组成^[40]。

(1) 遗传算法类的数据结构设计

C++ STL所提供的泛型数据结构用模板类给出，统称为容器类，包括vector、list、queue、set等。STL通过容器实现存放和管理其他对象等功能，当程序设计较为复杂的数据结构时，利用容器对众多复杂的对象进行管理，可以避免操纵大量独立的对象。容器仅按一定的数据结构存放和管理内部对象，而不会涉及到对象内部的数据。

本文定义了：

```
vector<bool> Chrom; // 染色体基因序列定义为bool值的向量类
```

```
vector<CIndividual> OldPop; // 种群定义为CIndividual类对象的向量类
```

针对每次电力系统故障中待诊断元件数量不能确定的特点，传统数据结构使用固定长度数组会导致超大内存占用，链表则会增大类型设计难度或者由于指针操作错误产生数据丢失。由于STL容器比数组更加强大灵活，程序在CIndividual类构造函数中使用vector算法push_back()轻松实现了根据待诊断元件的数目动态变化确定染色体基因序列长度值，不但降低的数据结构的复杂性，同时也增强了程序可靠性。

从严格意义上vector<bool>不是真正的容器，要成为STL容器就必须满足所有在C++标准中列出的容器必要条件^[41]。但是本文选择vector<bool>作为染色体对象类型，是因为在STL中对vector<bool>使用近似位域的方式压缩bool类型值的存储，对bool值打包并使每个在vector中保存的bool值仅占一个单独的位，最大程度的节省了空间，而且STL通过代理对象技术实现了下标访问，

解决了不能对位进行地址操作的问题。

(2) 遗传算法类算子设计

C++ STL泛型算法是以模板函数的方式提供出来的，可对具有泛型数据结构的容器进行数据处理。常用的排序、交换、查找和搜索功能，以泛化算法的形式定义到C++ STL库中，避免了不必要的重复设计。STL的算法函数有几十种之多，分为不修改容器数据的遍历/查找/搜索/计数/匹配等非变易算法、复制/替换/交换/反向排列等变易算法、排序算法和数值计算算法等。适当的使用正确的算法不但能够加快开发速度，而且能够减少逻辑错误。

比如为了动态设定染色体长度，程序使用的循环调用push_back()算法；在CGA类Cross()方法中通过区间交换算法swap_ranges()、元素复制算法copy()实现单点交叉算子。

文献[20]的程序使用了引入异或位变量标志设计变异算子，本文通过泛型算法flip()轻松解决，从而代码的可读性和可移植性都得到了增强。

(3) 代码安全性

虽然随着硬件技术的发展，算法实现的空间复杂度（空间复杂度是指算法在计算机内执行时所需存储空间的度量）已不再制约算法的实现，但内存使用的安全性却凸现出来。

而使用手工管理内存方式中，C语言的malloc/free只是对内存进行分配和释放，虽然C++的new/delete还负责完成了创建和销毁对象的任务，但是效率低，开辟的存储区不连续，从而减低访问速度，并且容易导致严重内存溢出错误，例如：内存分配虽然成功，但是尚未初始化就引用；内存分配成功并且已经初始化，但操作越过了内存的边界；忘记了释放内存，造成内存泄露；释放了内存却继续使用等等。

基于STL的C++程序设计中，常使用容器vector管理同一种类型的对象，vector对象（以及其他标准库容器）的重要属性就在于可以在运行时高效地添加元素，非特殊情况时，程序员不用专注内存的管理。其内存管理由内存配置器allocator负责，allocator是C++内存管理一个突破，使得内存管理从容器的实现中独立出来，便于STL使用者专注容器的使用和算法的设计，提高了内存管理的效率及可靠性。

4.5 基于遗传算法故障诊断的算例

如图4.5所示线路系统，共有28个元件、40个断路器和84个保护。

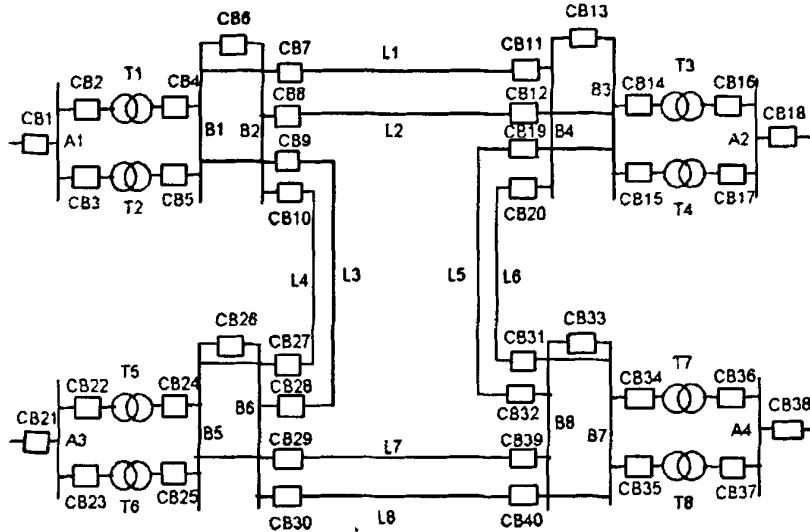


图 4.5 算例系统线路图

28个元件依次编号为 ($S_1 \sim S_{28}$) : A1, …, A4; B1, …, B8; T1, …, T8; L1, …, L8。

40个断路器依次编号为 (C₁~C₄₀) : QF1, QF2, …, QF40。

84个保护中，36个为主保护，48个为后备保护。

36个主保护依次编号为 (r₁~r₃₆) : A1m, …, A4m; B1m, …, B8m; T1m, …, T8m; L1Sm, L1Rm, …, L8Sm, L8Rm。

48个后备保护依次编号为 (r₃₇~r₈₄) : L1Sp, L1Rp, …, L8Sp, L8Rp, T1p, …, T8p, L1Ss, L1Rs, …, L8Ss, L8Rs, T1s, …, T8s。

在上述名称中，A和B表示母线，T表示变压器，L表示线路。S和R分别表示线路的发送端和接受端，下标m、p和s分别表示主保护、第一后备保护和第二后备保护。

本文故障诊断研究限定为故障区域已经明确，故障元件必定在此故障区域的情况。对算例系统发生的简单故障和复杂故障两种情况进行诊断。

4.5.1 故障诊断算例 1

(1) 识别故障区域

当系统发出警报信号：保护B1m、L2Rs、L4Rs动作，断路器QF4、QF5、QF7、QF9、QF12、QF27跳闸，形成如图4.6所示故障区域。

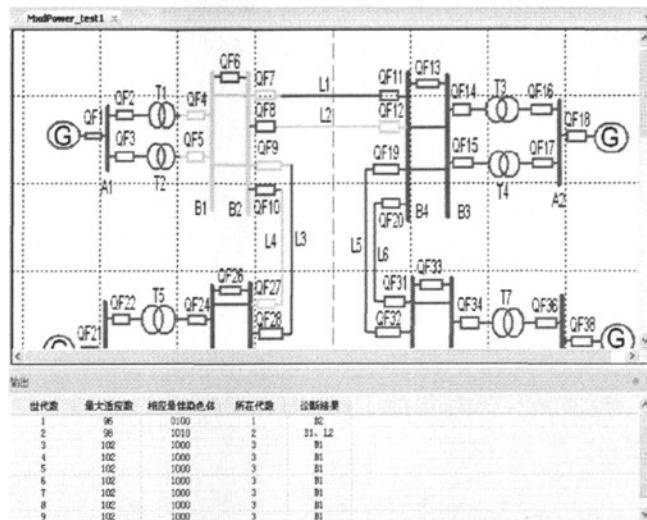


图 4.6 测试系统故障区域图

根据跳闸断路器形成的故障区域中，需要进行故障诊断的元件为B1、B2、L2、L4，依次编号为(s₁~s₄)，9个断路器(QF1~QF9)依次编号为(c₁~c₉)，14个保护(B1m、B2m、L2Sm、L2Rm、L4Sm、L4Rm、L2Sp、L2Rp、L4Sp、L4Rp、L2Ss、L2Rs、L4Ss、L4Rs)依次编号为(r₁~r₁₄)。

(2) 确定故障区域内保护和断路器动作逻辑

以下表4.1、表4.2和表4.3分别给出这四个元件的保护动作原理。

表 4.1 故障区域元件主保护动作原理表

| 保护名 | 动作原理 |
|------|------------------------------------|
| B1m | B1发生故障时，B1m动作跳开QF4、QF5、QF6、QF7、QF9 |
| B2m | B2发生故障时，B2m动作跳开QF6、QF8、QF10 |
| L2Sm | L2发生故障时，L2Sm动作跳开QF8 |
| L2Rm | L2发生故障时，L2Rm动作跳开QF12 |
| L4Sm | L4发生故障时，L4Sm动作跳开QF10 |
| L4Rm | L4发生故障时，L4Rm动作跳开QF27 |

表 4.2 故障区域元件第一后备保护动作原理表

| 保护名 | 动作原理 |
|------|------------------------------------------|
| L2Sp | L2 故障时, QF8 在主保护动作时段未跳开, L2Sp 动作跳开 QF8 |
| L2Rp | L2 故障时, QF12 在主保护动作时段未跳开, L2Rp 动作跳开 QF12 |
| L4Sp | L4 故障时, QF10 在主保护动作时段未跳开, L4Sp 动作跳开 QF10 |
| L4Rp | L4 故障时, QF27 在主保护动作时段未跳开, L4Rp 动作跳开 QF27 |

表 4.3 故障区域元件第二后备保护动作原理表

| 保护名 | 动作原理 |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| L2Ss | 在故障区域内无关联元件 |
| L2Rs | B1 故障且 QF6 在 B1 主保护动作时段内未跳开, 那么 L2Rs 动作跳开 QF12, 防止故障沿 L2 蔓延; 或者 B2 故障且 QF8 在 B2 的主保护动作时段未跳开, 此时为防止故障沿 L2 蔓延, 由 L2Rs 动作跳开 QF12; 或者 L4 故障且 QF10 在其主保护、第一后备保护动作时段内未跳开, 此时为防止故障沿 L2 蔓延, 由 L2Rs 动作跳开 QF12。 |
| L4Ss | 在故障区域内无关联元件 |
| L4Rs | B1 故障且 QF6 在 B1 主保护动作时段内未跳开, 那么 L4Rs 动作跳开 QF27, 防止故障沿 L4 蔓延; 或者 B2 故障且 QF10 在 B2 的主保护动作时段未跳开, 此时为防止故障沿 L4 蔓延, 由 L2Rs 动作跳开 QF27; 或者 L2 故障且 QF10 在其主保护、第一后备保护动作时段内未跳开, 此时为防止故障沿 L4 蔓延, 由 L2Rs 动作跳开 QF27。 |

(3) 确定各个保护和断路器实际状态向量

根据警报条件可以从故障诊断图形数据库中读出保护、断路器的实际状态向量。

断路器实际状态向量如式 (4-3) 所示:

$$C = [c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9] = [1, 1, 0, 1, 0, 1, 0, 1, 1] \quad (4-3)$$

保护实际状态向量如式4-4所示:

$$\begin{aligned} R &= [r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, r_{10}, r_{11}, r_{12}, r_{13}, r_{14}] \\ &= [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1] \end{aligned} \quad (4-4)$$

(4) 确定保护和断路器期望状态向量

结合保护动作原理及关联分析, 形成如下实际优化的具体目标函数公式。

主保护的期望状态如式(4-5)所示。

$$\begin{cases} r_1^*(S) = s_1 \\ r_2^*(S) = s_2 \\ r_3^*(S) = s_3 \\ r_4^*(S) = s_3 \\ r_5^*(S) = s_4 \\ r_6^*(S) = s_4 \end{cases} \quad (4-5)$$

第一后备保护的期望状态如式(4-6)所示。

$$\begin{cases} r_7^*(S) = s_3(1 - r_3) \\ r_8^*(S) = s_3(1 - r_4) \\ r_9^*(S) = s_4(1 - r_5) \\ r_{10}^*(S) = s_4(1 - r_6) \end{cases} \quad (4-6)$$

第二后备保护的期望状态如式(4-7)所示:

$$\begin{cases} r_{11}^*(S) = 0 \\ r_{12}^*(S) = 1 - [1 - s_2(1 - c_5)][1 - s_1(1 - c_3)(1 - c_5)][1 - s_4(1 - c_7)(1 - c_5)] \\ r_{13}^*(S) = 0 \\ r_{14}^*(S) = 1 - [1 - s_2(1 - c_7)][1 - s_1(1 - c_3)(1 - c_7)][1 - s_3(1 - c_7)(1 - c_5)] \end{cases} \quad (4-7)$$

由保护触发的断路器的期望状态如式(4-8)所示:

$$\left. \begin{array}{l}
 c_1^*(S, R) = r_1^* r_1 \\
 c_2^*(S, R) = r_1^* r_1 \\
 c_3^*(S, R) = \max \{r_1^* r_1, r_2^* r_2\} \\
 c_4^*(S, R) = r_1^* r_1 \\
 c_5^*(S, R) = \max \{r_2^* r_2, r_3^* r_3, r_7^* r_7, r_{11}^* r_{11}\} \\
 c_6^*(S, R) = r_1^* r_1 \\
 c_7^*(S, R) = \max \{r_2^* r_2, r_9^* r_9, r_5^* r_5, r_{13}^* r_{13}\} \\
 c_8^*(S, R) = \max \{r_4^* r_4, r_{12}^* r_{12}, r_8^* r_8\} \\
 c_9^*(S, R) = \max \{r_6^* r_6, r_{10}^* r_{10}, r_{14}^* r_{14}\}
 \end{array} \right\} \quad (4-8)$$

(5) 设置遗传算法参数

种群规模: `m_nPopSize = 40;`

染色体长度: `m_nChromLength = 4;`

最大进化代数: `m_nMaxGeneration = 10;`

交叉概率: `m_pCross = 0.9;`

变异概率: `m_pMutation = 0.01.`

算法策略: 选择算子采用轮盘赌方法, 随机产生一个[0, 1]之间的值, 作为进行轮盘赌选择的选择指针, 个体适应度除以累计适应度为选择概率。交叉算子采用单点交叉方式。

(6) 利用遗传算法求解

诊断结果为: 母线B1故障。

(7) 程序运行结果及分析

根据保护和断路器的报警信号以及诊断结果, 可以分析得知, 母线B1发生故障, B1主保护B1m动作, 发出指令跳开断路器QF4、QF5、QF6、QF7、QF9。断路器QF4、QF5、QF7、QF9顺序动作, 断路器QF6拒动, 故障波及母线B2、L2、L4, 此时由线路L2、L4第二后备保护L2Rs、L4Rs将L2、L4对侧区域切除, 避免事故进一步扩大。此中情况是保护正确动作, 但存在断路器拒动的情况。

4.5.2 故障诊断算例 2

(1) 识别故障区域

当系统发生复杂多重故障，发出警报信号：保护T7m、T8p、B7m、B8m、L5Sm、L5Rp、L6Ss、L7Sp、L7Rm、L8Ss动作，断路器QF19、QF20、QF29、QF30、QF32、QF33、QF34、QF35、QF36、QF37、QF39跳闸，形成如图4-7所示故障区域。

根据跳闸断路器形成的无源网络得到故障区域中，需要进行故障诊断的元件为L5、L6、L7、L8、B7、B8、T7、T8依次编号为(s₁~s₈)，13个断路器(QF19~QF40)依次编号为(c₁~c₁₃)，32个保护(L5Sm、L5Rm、L6Sm、L6Rm、L7Sm、L7Rm、L8Sm、L8Rm、B7m、B8m、T7m、T8m、L5Sp、L5Rp、L6Sp、L6Rp、L7Sp、L7Rp、L8Sp、L8Rp、T7p、T8p、L5Ss、L5Rs、L6Ss、L6Rs、L7Ss、L7Rs、L8Ss、L8Rs、T7s、T8s)依次编号为(r₁~r₃₂)，以下表4.4、表4.5和表4.6分别给出这八个元件的保护动作原理和关联关系。

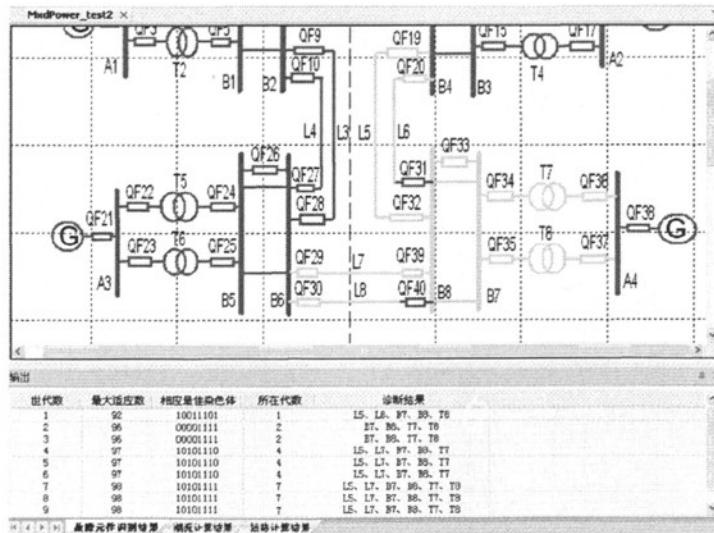


图 4.7 测试系统故障区域图

(2) 确定故障区域内保护和断路器动作逻辑

表 4.4 故障区域元件主保护动作原理表

| 保护名 | 动作原理 |
|------|-------------------------------------------|
| L5Sm | L5 故障时, L5Sm 动作跳开 QF19 |
| L5Rm | L5 故障时, L5Rm 动作跳开 QF32 |
| L6Sm | L6 故障时, L6Sm 动作跳开 QF20 |
| L6Rm | L6 故障时, L6Rm 动作跳开 QF31 |
| L7Sm | L7 故障时, L7Sm 动作跳开 QF29 |
| L7Rm | L7 故障时, L7Rm 动作跳开 QF39 |
| L8Rm | L8 故障时, L8Sm 动作跳开 QF30 |
| L8Sm | L8 故障时, L8Rm 动作跳开 QF40 |
| B7m | B7 故障时, B7m 动作跳开 QF31、QF33、QF34、QF35、QF40 |
| B8m | B8 故障时, B8m 动作跳开 QF32、QF33、QF39 |
| T7m | T7 故障时, T7m 动作跳开 QF34、QF36 |
| T8m | T8 故障时, T8m 动作跳开 QF35、QF37 |

表 4.5 故障区域元件第一后备保护动作原理表

| 保护名 | 动作原理 |
|------|---------------------------------------------------|
| L5Sp | L5 故障时, QF19 在主保护动作时段未跳开, L5Sp 动作跳开 QF19 |
| L5Rp | L5 故障时, QF32 在主保护动作时段未跳开, L5Rp 动作跳开 QF32 |
| L6Sp | L6 故障时, QF20 在主保护动作时段未跳开, L6Sp 动作跳开 QF20 |
| L6Rp | L6 故障时, QF31 在主保护动作时段未跳开, L6Rp 动作跳开 QF31 |
| L7Sp | L7 故障时, QF29 在主保护动作时段未跳开, L7Sp 动作跳开 QF29 |
| L7Rp | L7 故障时, QF39 在主保护动作时段未跳开, L7Rp 动作跳开 QF39 |
| L8Sp | L8 故障时, QF30 在主保护动作时段未跳开, L8Sp 动作跳开 QF30 |
| L8Rp | L8 故障时, QF40 在主保护动作时段未跳开, L8Rp 动作跳开 QF40 |
| T7p | T7 故障时, QF34、QF36 在主保护动作时段未跳开, T7p 动作跳开 QF34、QF36 |
| T8p | T8 故障时, QF35、QF37 在主保护动作时段未跳开, T8p 动作跳开 QF35、QF37 |

表 4.6 第二后备保护关联元件及关联路径综合表

| 元件名称 | 保护名 | 关联元件 | 跳开的断路器 | 关联路径 |
|------|------|------|--------|----------------|
| L5 | L5Ss | B7 | QF19 | QF32、QF33 |
| | | B8 | | QF32 |
| | | L7 | | QF32、QF39 |
| | | T7 | | QF32、QF33、QF34 |
| | | T8 | | QF32、QF33、QF35 |
| | | L6 | | QF32、QF33、QF31 |
| | | L8 | | QF32、QF33、QF40 |
| | | L5Rs | 无 | 无 |
| L6 | L5Ss | B7 | QF20 | QF31 |
| | | B8 | | QF31、QF33 |
| | | L5 | | QF31、QF32、QF33 |
| | | L7 | | QF31、QF33、QF39 |
| | | L8 | | QF31、QF40 |
| | | T7 | | QF31、QF34 |
| | | T8 | | QF31、QF35 |
| | | L6Rs | 无 | 无 |
| L7 | L7Ss | L5 | QF29 | QF39、QF32 |
| | | L6 | | QF39、QF33、QF31 |
| | | L8 | | QF39、QF33、QF40 |
| | | B7 | | QF39、QF33 |
| | | B8 | | QF39 |
| | | T7 | | QF39、QF33、QF34 |
| | | T8 | | QF39、QF33、QF35 |
| | | L7Rs | 无 | 无 |
| L8 | L8Ss | L5 | QF30 | QF40、QF32、QF33 |
| | | L6 | | QF40、QF31 |
| | | L7 | | QF40、QF33、QF39 |
| | | B7 | | QF40 |
| | | B8 | | QF40、QF33 |
| | | T7 | | QF40、QF34 |
| | | T8 | | QF40、QF35 |

| | L8Rs | 无 | 无 | 无 |
|----|------|----|--------------|----------------|
| T7 | T7s | L5 | QF34 QF36 | QF34、QF33、QF32 |
| | | L6 | | QF34、QF31 |
| | | L7 | | QF34、QF33、QF39 |
| | | L8 | | QF34、QF40 |
| | | B7 | | QF34 |
| | | B8 | | QF34、QF33 |
| T8 | T8s | L5 | QF35 QF37 | QF35、QF33、QF32 |
| | | L6 | | QF35、QF31 |
| | | L7 | | QF35、QF33、QF39 |
| | | L8 | | QF35、QF40 |
| | | B7 | | QF35 |
| | | B8 | | QF35、QF33 |

(3) 确定各个保护和断路器实际状态向量

根据警报条件可以从故障诊断图形数据库中读出保护、断路器的实际状态向量。断路器实际状态向量：

$$C = [c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}] = [1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0] \quad (4-9)$$

保护实际状态向量：

$$R = [r_1, r_2, \dots, r_{12}] = [1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0] \quad (4-10)$$

(4) 确定保护和断路器期望状态向量

结合保护动作原理及关联分析，形成如下实际优化的具体目标函数公式。

主保护的期望状态如式(4-11)所示。

$$\left[\begin{array}{l} \dot{r_1}(S) = s_1 \\ \dot{r_2}(S) = s_1 \\ \dot{r_3}(S) = s_2 \\ \dot{r_4}(S) = s_2 \\ \dot{r_5}(S) = s_3 \\ \dot{r_6}(S) = s_3 \\ \dot{r_7}(S) = s_4 \\ \dot{r_8}(S) = s_4 \\ \dot{r_9}(S) = s_5 \\ \dot{r_{10}}(S) = s_6 \\ \dot{r_{11}}(S) = s_7 \\ \dot{r_{12}}(S) = s_8 \end{array} \right] \quad (4-11)$$

第一后备保护的期望状态如式(4-12)所示。

$$\left[\begin{array}{l} r_{13}(S) = s_1(1-r_1) \\ r_{14}(S) = s_1(1-r_2) \\ r_{15}(S) = s_2(1-r_3) \\ r_{16}(S) = s_2(1-r_4) \\ r_{17}(S) = s_3(1-r_5) \\ r_{18}(S) = s_3(1-r_6) \\ r_{19}(S) = s_4(1-r_7) \\ r_{20}(S) = s_4(1-r_8) \\ r_{21}(S) = s_7(1-r_{11}) \\ r_{22}(S) = s_8(1-r_{12}) \end{array} \right] \quad (4-12)$$

第二后备保护的期望状态如式(4-13)所示:

$$\left[\begin{array}{l} r_{23}(S) = 1 - [1 - s_5(1 - c_6)(1 - c_7)][1 - s_3(1 - c_6)(1 - c_{12})] \\ \quad [1 - s_7(1 - c_6)(1 - c_7)(1 - c_8)][1 - s_8(1 - c_6)(1 - c_7)(1 - c_9)] \\ \quad [1 - s_2(1 - c_6)(1 - c_5)(1 - c_7)][1 - s_4(1 - c_6)(1 - c_7)(1 - c_{13})] \\ \quad [1 - s_6(1 - c_6)] \\ r_{25}(S) = 1 - [1 - s_6(1 - c_5)(1 - c_7)][1 - s_4(1 - c_5)(1 - c_{13})] \\ \quad [1 - s_1(1 - c_5)(1 - c_7)(1 - c_6)][1 - s_3(1 - c_5)(1 - c_7)(1 - c_{12})] \\ \quad [1 - s_7(1 - c_5)(1 - c_8)][1 - s_8(1 - c_5)(1 - c_9)][1 - s_5(1 - c_5)] \\ r_{27}(S) = 1 - [1 - s_6(1 - c_{12})][1 - s_1(1 - c_{12})(1 - c_{16})] \\ \quad [1 - s_2(1 - c_{12})(1 - c_7)(1 - c_5)][1 - s_4(1 - c_{12})(1 - c_7)(1 - c_{13})] \\ \quad [1 - s_7(1 - c_{12})(1 - c_7)(1 - c_8)][1 - s_8(1 - c_{12})(1 - c_7)(1 - c_9)] \\ \quad [1 - s_5(1 - c_{12})(1 - c_7)] \\ r_{29}(S) = 1 - [1 - s_5(1 - c_{13})][1 - s_2(1 - c_{13})(1 - c_5)] \\ \quad [1 - s_1(1 - c_{13})(1 - c_6)(1 - c_7)][1 - s_3(1 - c_{13})(1 - c_7)(1 - c_{12})] \\ \quad [1 - s_7(1 - c_{13})(1 - c_8)][1 - s_8(1 - c_{13})(1 - c_9)] \\ \quad [1 - s_6(1 - c_{13})(1 - c_7)] \\ r_{31}(S) = 1 - [1 - s_5(1 - c_8)][1 - s_2(1 - c_8)(1 - c_5)] \\ \quad [1 - s_1(1 - c_8)(1 - c_6)(1 - c_7)][1 - s_3(1 - c_8)(1 - c_7)(1 - c_{12})] \\ \quad [1 - s_6(1 - c_7)(1 - c_8)][1 - s_8(1 - c_8)(1 - c_9)] \\ \quad [1 - s_4(1 - c_8)(1 - c_{13})] \\ r_{32}(S) = 1 - [1 - s_5(1 - c_9)][1 - s_2(1 - c_9)(1 - c_5)] \\ \quad [1 - s_1(1 - c_9)(1 - c_6)(1 - c_7)][1 - s_3(1 - c_9)(1 - c_7)(1 - c_{12})] \\ \quad [1 - s_6(1 - c_9)(1 - c_7)][1 - s_7(1 - c_8)(1 - c_9)] \\ \quad [1 - s_7(1 - c_9)(1 - c_{13})] \end{array} \right] \quad (4-13)$$

其中 $r_{24}^*(S) = 0$, $r_{26}^*(S) = 0$, $r_{28}^*(S) = 0$, $r_{30}^*(S) = 0$ 。

由保护触发的断路器的期望状态如式(4-14)所示:

$$\left[\begin{array}{l} c_1^*(S, R) = \max \{r_1^* r_1, r_{13}^* r_{13}, r_{23}^* r_{23}\} \\ c_2^*(S, R) = \max \{r_1^* r_1, r_{13}^* r_{13}, r_{23}^* r_{23}\} \\ c_3^*(S, R) = \max \{r_5^* r_5, r_{17}^* r_{17}, r_{27}^* r_{27}\} \\ c_4^*(S, R) = \max \{r_7^* r_7, r_{19}^* r_{19}, r_{29}^* r_{29}\} \\ c_5^*(S, R) = \max \{r_4^* r_4, r_{16}^* r_{16}, r_{26}^* r_{26}, r_9^* r_9\} \\ c_6^*(S, R) = \max \{r_2^* r_2, r_{14}^* r_{14}, r_{24}^* r_{24}, r_{10}^* r_{10}\} \\ c_7^*(S, R) = \max \{r_9^* r_9, r_{10}^* r_{10}\} \\ c_8^*(S, R) = \max \{r_9^* r_9, r_{11}^* r_{11}, r_{21}^* r_{21}, r_{31}^* r_{31}\} \\ c_9^*(S, R) = \max \{r_9^* r_9, r_{12}^* r_{12}, r_{22}^* r_{22}, r_{32}^* r_{32}\} \\ c_{10}^*(S, R) = \max \{r_{11}^* r_{11}, r_{21}^* r_{21}, r_{31}^* r_{31}\} \\ c_{11}^*(S, R) = \max \{r_{12}^* r_{12}, r_{22}^* r_{22}, r_{32}^* r_{32}\} \\ c_{12}^*(S, R) = \max \{r_6^* r_6, r_{18}^* r_{18}, r_{28}^* r_{28}, r_{10}^* r_{10}\} \\ c_{13}^*(S, R) = \max \{r_8^* r_8, r_{20}^* r_{20}, r_{30}^* r_{30}, r_9^* r_9\} \end{array} \right] \quad (4-14)$$

(5) 设置遗传算法参数

种群规模: $m_nPopSize = 40$;

染色体长度: $m_nChromLength = 4$;

最大进化代数: $m_nMaxGeneration = 10$;

交叉概率: $m_PCross = 0.9$;

变异概率: $m_PMutation = 0.01$ 。

算法策略: 选择算子采用轮盘赌方法, 随机产生一个[0, 1]之间的值, 作为进行轮盘赌选择的选择指针, 个体适应度除以累计适应度为选择概率。交叉算子采用单点交叉方式。

(6) 利用遗传算法求解

得到如图4.7给出的诊断结果为: 线路L5、L7, 母线B7、B8和变压器T7、T8故障。

(7) 程序运行结果及分析

根据保护和断路器的报警信号以及诊断结果，可以分析得知，线路L5、L7，母线B7、B8和变压器T7、T8故障时，线路L5的S侧主保护L5Sm，线路L7的R侧主保护L7Rm动作，母线B7、B8的主保护B7m、B8m正确动作，变压器T7主保护T7m动作。由于线路L5的R侧主保护L5Rm未能正确动作，L5的R侧第一后备保护L5Rp动作，同理，L7的S侧第一后备保护L7Sp动作，变压器T8第一后备保护T8p动作。这些保护动作之后，断路器QF19、QF29、QF31、QF32、QF33、QF34、QF35、QF36、QF37、QF39、QF40应该跳闸，但是由于断路器QF31、QF40拒动，线路L6、L8S侧的第二后备保护L6Ss、L8Ss动作，将断路器QF20、QF30切断，避免事故进一步扩大。此中情况是主保护拒动和断路器拒动的多重复杂故障情况。

4.6 本章小结

为了适应电力系统故障诊断快速性的要求，本章引入遗传算法对故障诊断目标函数进行优化求解。在介绍遗传算法的基础上，开发了故障诊断遗传算法类，给出适合电力系统故障诊断应用的二进制编码方式、适应度函数以及遗传算子的选择。根据遗传算法求解电力系统故障诊断最优解的流程。最后，通过两个故障算例验证了诊断模型和诊断程序的正确性。

第5章 故障诊断的图形数据库平台实现

为了实现电力系统故障诊断设备、保护和断路器状态的存储与读取，满足复杂电力系统故障诊断的需求，还需要开发电力系统图形数据库平台。本章介绍电力设备图形的绘制与管理，给出了故障诊断数据库的设计方案，介绍了故障诊断数据库与图形绘制平台的映射实现。为了验证故障诊断程序与图形数据库平台，本文还给出了验证实例。

5.1 图形界面设计

微软公司最新发布的Visual C++2008英文版。专门针对MFC库进行两方面扩充：一方面是对ISO C++ Standard 2003标准的扩充（这在第4章使用STL算法已经有所体现）；另一方面就是是MFC的界面控件库。

图形用户界面的关键在于使用方便，界面友好。本文在研究Ribbon风格界面程序框架的基础上，开发了基于Office Ribbon风格的图形用户界面，如图5.1所示，实现了属性浮动窗口和输出浮动窗口。

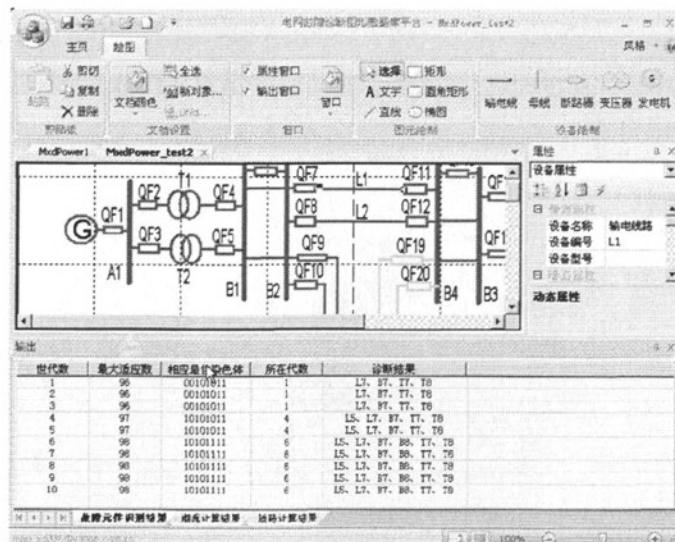


图 5.1 电力系统故障诊断图形数据库平台界面

5.2 电力系统图形数据库平台

5.2.1 概述

本论文出于电力系统故障诊断研究、搭建电力系统模型的需求，从电力系统图形和数据库一体化模式着手，研究了交互式CAD图形数据库平台的功能设计和具体开发的问题。论文通过面向对象的可视化方法，开发了图库一体化的人机交互平台，满足为本论文应用电力系统建立模型。

电力系统图库一体化技术是指把电力设备图元与该设备在电力系统数据库中对应的信息作为一个整体来进行处理^[44]。电力设备图形是数据库记录的一种表现形式，其特点是形象和直观。电力系统数据库是对各种电力设备图形元件所表示的信息进行存储、处理和管理的环境。

任何一个系统的建立，其实质都是对该系统所处理的数据信息进行数据结构和数据关系的建立。电力设备图元和其数据库信息必须按照一定的关系关联到一起才有实际意义，手工填写数据库记录的数据没有实际应用意义，所以必须通过操作图形，把数据库中的数据当作图形的属性来定义和修改，只要找到图形就可以找到与之对应的数据，电力图形数据库一体化技术是目前电力系统软件开发的主流技术。

电力系统图形数据库一体化的关键问题是实现面向电力系统设备的建模方式^[45]。其实质是设备图元对象与数据库记录一一对应，在定义一个设备图元对象的同时自动追加一条数据库记录。另一方面，要求设备图元的属性参数及其数据输入界面能够根据用户的要求进行修正，这样才能真正做到图形与数据库的一一对应，而且也能保证系统的兼容性。在定义设备图元对象的时候，根据图元设备的属性自动生成数据库及其它图元属性，并且能够自动绑定数据库，不仅省去了图形数据库的人工绑定的繁琐，而且减少了操作错误的可能性。

本文开发的电力系统图形数据库一体化平台应实现以下两个功能：

- (1) 具有绘制母线、线路、断路器、变压器和发电机图元的功能，并能够对这五类图元进行块操作，如复制、粘贴、移动等等。
- (2) 平台实现了设备图元文件与故障诊断数据库的自动对应关系，并且

能够在双击界面上电力设备图元后，进行相应图元的数据库信息的录入与读取。

5.2.2 坐标空间及变换

在使用MFC进行Windows图形应用程序设计时，根据绘图位置，使用绘图方法的不同通常会用到不同的坐标系统，绘图过程同时也伴随着不同坐标系统坐标之间的映射及转换关系。本文使用到以下四种坐标空间：

物理坐标空间：我们最终要绘制的目的地——屏幕，是物理设备空间。

逻辑坐标空间：设备描述表（CDC类的大多数成员函数，如MoveTo（50, 50））使用的坐标是逻辑坐标空间的逻辑坐标值。

设备坐标空间：常指视图窗口空间，大多数CWnd函数，如鼠标消息中的参数坐标是设备坐标。

实际坐标空间：本文建立了适合无极缩放矢量图形系统的，无坐标范围限制且坐标值为整形的坐标空间，原点取在画图板的中心位置，X轴向右为正，Y轴向上为正。

一旦建立了设备描述表，对大多数CDC成员函数来说，我们就可以以逻辑坐标作为其参数。但是，对于涉及到诸如鼠标捕捉、选中测试操作以及许多其它的MFC库函数，尤其是类CRect的成员函数，则只能在设备坐标下才能操作，因此必须解决不同坐标之间的转换。

设备坐标空间→物理坐标空间：视图窗口（设备坐标空间）可在屏幕（物理坐标空间）移动，映射关系由Windows自动完成，程序员不能控制此映射规则。

逻辑坐标空间→设备坐标空间：Windows通过一定的映射方式，把逻辑空间中的图形映射到设备坐标（视图窗口），映射方式由程序员选择。

实际坐标空间→逻辑坐标空间：程序根据画板实际大小、原点平移量进行实际坐标空间和逻辑坐标空间的转换。本文程序主要通过方法ClientToWorld()、WorldToClient()、ClientToWorld()、WorldToClient()分别实现了逻辑坐标转换为实际坐标、实际坐标转换为逻辑坐标、逻辑长度转换为实际长度、实际长度转换为逻辑长度。

5.2.3 电力设备的绘制与管理

(1) 设备图元类定义

电力系统设备图元基本属性是图形属性。本文将其定义为设备的图形固有属性，即设备元件的位置、颜色、名称等。以输电线路类为例，其固有属性包括图形宽、高、形状、颜色。

```
BOOL m_bPen;           //设备图形绘制
CRect m_position;     //设备图形位置
enum Shape { transline }; //设备图形选择定义
```

(2) 图元数据结构管理

考虑到所绘电力系统图元个数不定，并且在绘制及使用后会根据实际应用进行更改，不益使用数组直接管理，因此需要确定一种合理的图元管理数据结构。通常MFC提供了两种风格的数据管理图元对象的集合类，即基于模板和不基于模板的集合类。本文通过以下分析确定了图元管理数据结构：

基于模板的类具有更好的类型安全性，一个类型安全类只能同时保存同一种类型的数据；

集合类有三种形式^[49]：数组Array、链表List和映射Map。从它们的效率分析Map最快，链表次之，数组最慢，但映射类采用键值来映射数据内容，正如一本字典一样，将关键字和他的定义分别存放在不同的地方，而本图形数据库平台不使用数据字典，所以采用链表管理图形数据。

本文由CObject类派生出了图元类，在此情况下，有两个合适的MFC数据结构：CObArry类和COList类。CObArry类封装了一个指向CObject的指针，因此从CObject派生的图元类可以将指针存放在数组中。CObList类保存指针在双向链表中，允许双向搜索。

在CObArry类和COList类中选择，不同的数据结构具有不同的表现特点，提供不同的功能。我们需要一个链表数据结构，这样OnDraw可以迅速绘制出所有图形，我们还需要在链表中移动图形对象。采用链表数据结构使用插入、删除、移动对象更方便。因此CObList是相对较好的选择。

通过上面分析，本文定义如下类型定义：

```
typedef CTyedPtrList<COBList, CDrawObj*> CDrawObjList
```

CDrawObjList的基类是COBList，存放在CDrawObjList类型链表对象中的类型是电力系统设备图元对象的指针。

(3) 交互操作类设计

电力系统设备的绘制首先选择一种绘图工具，然后在绘图区域按下鼠标左键不放，拖动鼠标进行绘图，当鼠标运动时，绘出所选工具对应的图形，绘制结束时，释放鼠标按钮，图形既出现在视图区。

本文把与图形交互操作的鼠标消息处理过程，封装成为一个交互工具类CDrawTool，由这个基类派生出不同的图形工具类，比如母线绘制工具类CBusbarTool和断路器工具类CBreakerTool。如图5.2所示，给出了交互工具类UML对象类图，显示了交互工具类的继承关系。

由于采用类的机制后实现了数据的隐藏与封装，类的数据成员一般定义为私有成员，成员函数一般定义为公有的，以此提供类与外界间的通信接口。图元绘制工具类作为绘制图形的工具，要经常图形对象访问调用，所以程序中把图形绘制工具类定义为相应图形类对象的友元类，友元的作用就是提高了程序的运行效率（即减少了类型检查和安全性检查等都需要时间开销），访问类的私有成员。

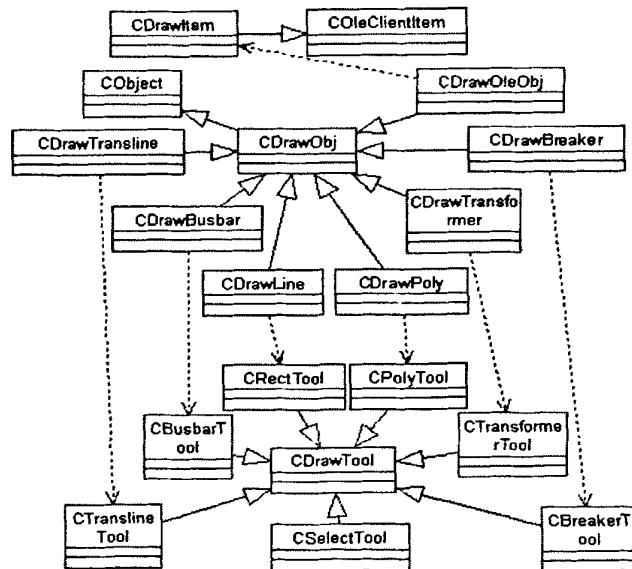


图 5.2 交互工具类 UML 对象类图

程序将具体的图形对象加入到定义好的CDrawObjList对象链表中，交互

工具类CDrawTool只定义了鼠标消息接口，当鼠标选择了某种工具的时候，就会通过CDrawTool::FindTool方法确定确定要绘制图形的相应图形工具类指针，通过该指针调用的方式响应图形工具的鼠标消息处理函数进而绘制图形。

5.3 基于故障诊断的数据库设计

5.3.1 数据库 E-R 图

本章基于遗传算法的故障诊断应用需求设计数据库结构。E-R图是一种用来在数据库数据过程中表示数据库系统结构的方法。它的主导思想是使用实体、实体的属性以及实体之间的关系。

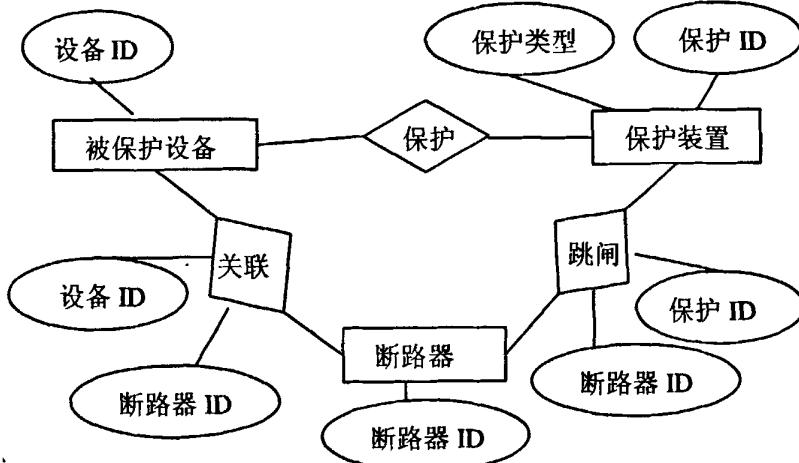


图 5.3 故障诊断数据库 E-R 图

如图5.3所示，共有三个实体：被保护设备、保护设备和断路器；三种联系：保护设备和被保护设备间的保护联系保护设备和断路器间的跳闸联系以及被保护设备与断路器之间的关联（关联元件、关联路径）关系。

(1) 其中保护设备和被保护设备间的保护联系属于n: 1联系，即一个被保护设备可以有若干套保护设备，而一个保护（通过保护ID区分）只属于一个设备，例如图中线路L2有主保护、第一后备保护和第二后备保护共三套保护，这三套保护设备用保护ID号区分共计五个保护装置，而这五个保护装置都只属于线路L2拥有的保护；

(2) 保护设备和断路器间的跳闸联系属于m: n联系：一个保护（通过保护ID区分）可以同时跳开几个断路器，而一个断路器也可能由几个不同的

保护跳开，例如图中母线B的主保护可以同时跳开两侧的断路器QF3和QF4，而断路器QF3既可以由Bm跳开，同时还可以由线路二的主保护、第一后备保护跳开；

(3) 被保护设备与断路器之间的关联（包括关联元件、关联路径）关系也属于m: n联系：通过对关联路径关联元件的分析可知，一个元件有若干关联元件，每个关联元件的关联路径中又包含若干断路器，同时一个断路器也可能属于不同元件的不同关联元件及其关联路径。

5.3.2 数据库关系模型

通常情况下从E-R图转换为数据库要遵循六条规则^[42]:

- (1) 一个实体转换为一个数据表。实体的属性转换为数据表的字段；
 - (2) 一对多的联系不转换为一个数据表。两个实体对应的数据表依靠外部关键字建立联系；
 - (3) 多对多的联系转换为一个数据表。该数据表的复合关键字为两个实体关键字；
 - (4) 三个或三个以上实体多对多的联系转换为一个数据表。该数据表的复合关键字为各个实体关键字，或引入单一字段（例如ID号）作为关键字，把各个实体的关键字作为外部关键字；
 - (5) 处理三个以上实体的联系时，先画出一对一对应的实体；
 - (6) 具有相同关键字的数据表可以合并为一个表。
- 因此，本文建立了五张表如下：
- 设备表（字段：设备ID、设备类型、设备状态）；
- 保护表（字段：保护ID、保护类型、被保护设备ID、送端或受端标志、保护动作情况）；
- 断路器表（字段：断路器ID、断路器状态）；
- 跳闸表（字段：保护ID、断路器ID）；
- 关联表（字段：设备ID、第二后备ID、关联元件ID、关联路径内断路器ID）。

如图5.-4所示为故障诊断数据库设备表截图

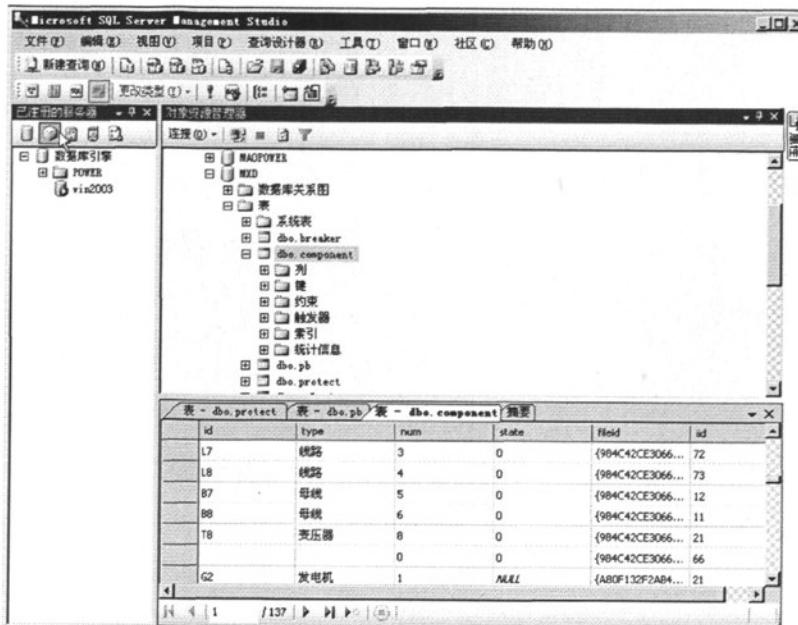


图 5.4 故障诊断数据库设备表截图

其中设备ID、保护ID及断路器ID分别是设备表、保护表及断路器表的主键，而跳闸表的外键约束为保护ID和断路器ID，关联表的外键约束为设备ID和断路器ID。

5.3.3 数据库访问

本文采用VC++的ADO (ActiveX Data Objects) 编程接口实现与数据库的连接。

ADO模型主要包括以下关键对象：

(1) 连接对象 (Connection)。在数据库应用里操作数据源都必须通过该对象，这是数据交换的环境。Connection对象代表了同数据源的一个会话，相当于同服务器的一次网络连接。

(2) 命令对象 (Command)。Command对象实现对数据源执行命令，可以查询数据库并返回一个Recordset对象，可以执行一个批量的数据操作，也可以操作数据库结构。

(3) 记录集对象 (Recordset)。如果执行的命令是一个查询并返回存放在表中的结果集，这些结果集将被保护在本地的存储区里，Recordset对象是

执行这种存储的ADO对象。

本文通过如下操作过程实现操作数据源里的数据：

- (1) 创建一个Connection对象。定义用于连接的字符串信息，包括数据源名称、用户ID、口令等。通过Connection建立与数据库的会话。
- (2) 打开数据源，建立与数据源的连接。
- (3) 执行SQL命令进行查询、存储操作，生成记录集，通过Recordset对象操作结果集。
- (4) 销毁连接对象，终止数据库连接。

5.3.4 图形数据库接口

电力系统设备除了具有图形属性外，还应该具备电力设备运行状态属性的特点。本文将电力设备运行状态属性也封装在电力设备类中。同时鉴于本文图形数据库的设计目标是承载电力系统故障诊断应用，所以采用了文件管理和数据库记录两种方式分别记录电力设备性质。

本文的设备图形属性由设备图元类管理，而设备的状态、故障诊断涉及到的设备关联关系由数据库来记录和体现。设备属性，设备属性通过文件中图形ID与数据库设备图形ID的记录对映绑定。

```
int g_draw_obj_id; //电力设备在文件中的编号
int iid;           //电力设备在数据库中的编号
```

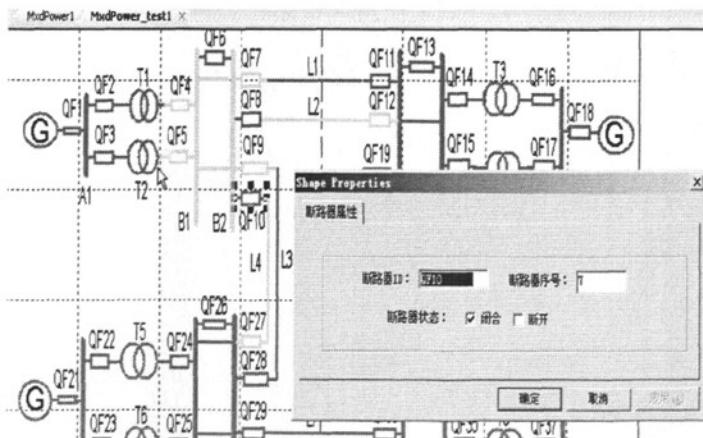


图 5.5 断路器属性与数据库交互

对电力设备的运行状态属性的操作通过设备属性状态类实现。例如对断路器的状态属性操作全部定义在断路器状态属性类中。如图5.5所示，通过对断路器设备图形属性窗口的调用，生成断路器属性对话框类的对象，并借此对象与数据库交互断路器状态信息。

5.4 平台测试算例

为了进一步验证故障诊断软件与图形数据库的配合效果及诊断的正确性。下面通过如图5.6所示IEEE中5机14结点系统，进行故障诊断验证。

将图5.6的五机系统转化为适合故障诊断的系统线路，如图5.7所示，共有37个元件、45个断路器和128个保护。

37个元件依次编号为 ($S_1 \sim S_{37}$) : G1, …, G5; A1, …, A14; T1, …, T3; L1, …, L15。

46个断路器依次编号为 ($C_1 \sim C_{46}$) : QF1, QF2, …, QF46。

127个保护中，51个为主保护，76个为后备保护。

51个主保护依次编号为 ($r_1 \sim r_{51}$) : G1m, …, G5m; A1m, …, A13m; ; T1m, …, T3m; L1Sm, L1Rm, …, L15A13m, L15A12m。

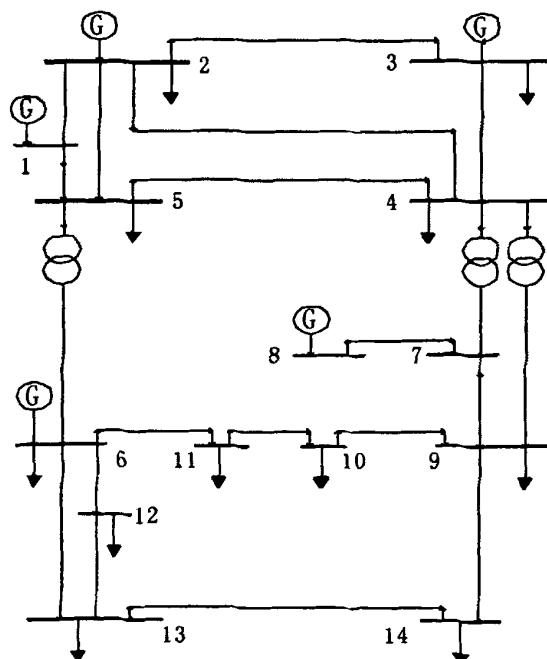


图5.6 IEEE五机系统图

76个后备保护依次编号为($r_{53} \sim r_{128}$)：G1p, …, G5p; T1p, …, T3p; L1A1p, L1A2p, …, L15A13p, L15A12p; G1s, …, G5s; T1s, …, T3s; L1A1s, L1A2s, …, L15A13s, L15A13s。当系统发出警报信号：保护G2m、A9m、L4A1s、L11A10s动作，断路器QF16、QF29、QF30、QF31跳闸，形成如图5.7所示故障区域。

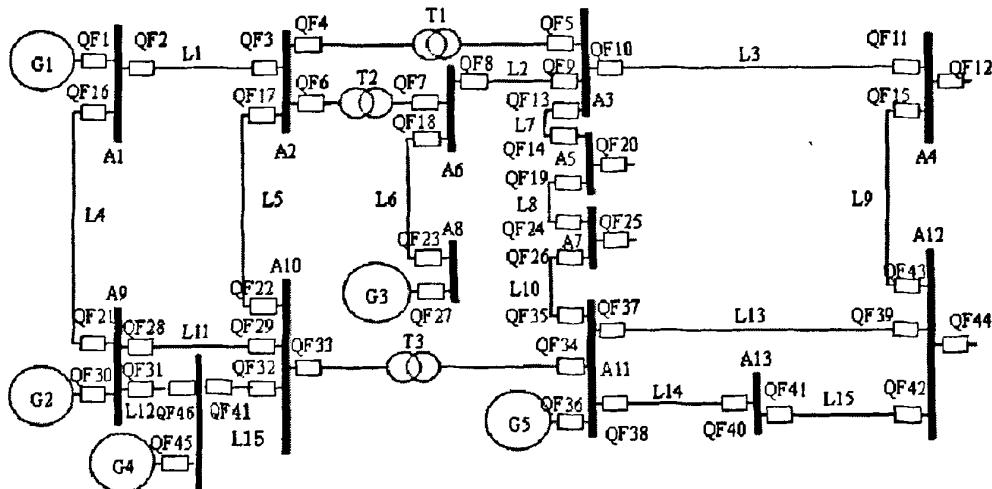


图 5.7 故障诊断线路系统

5.4.1 平台测试算例 1

(1) 绘制故障系统图，录入系统信息

电力系统故障诊断的设备、保护和断路器状态信息，以及他们之间的关联信息，均通过图形数据库平台双击设备或断路器所弹出的属性页进行设定，如图5.8和图5.9分别给出了母线和输电线路的属性录入界面，通过属性设定，在录入设备状态的同时，也确立了母线或者线路与其配套的保护，保护和与其关联的断路器的关联关系，以及与元件第二后备保护相关联的关联元件、关联路径的限定。

本文电力系统图形数据库平台是基于MFC多文档框架进行设计，如图5.10所示，给出了故障系统部分元件在数据库中的存储信息。数据库中字段fileid保存的了与图形文档的文件关联的信息，本算例的图形文件在数据库中的记录fileid为{A80F132F2AB44f2892A928A0BC5DBECE}，这个fileid是随机产生的，用于与其他图形文件的区分。

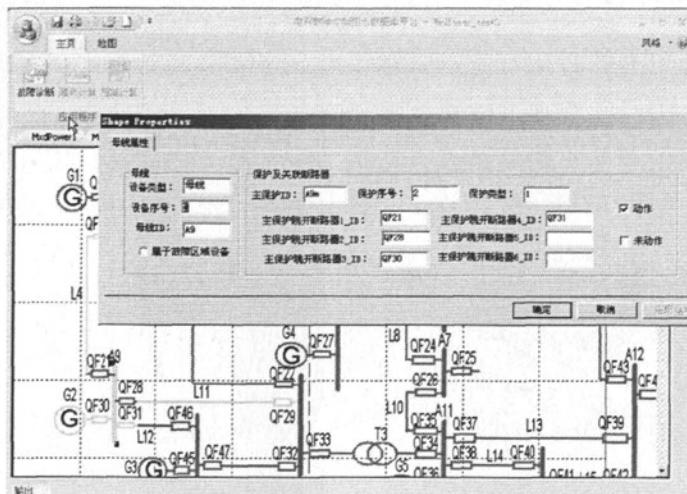


图 5.8 母线状态及关联属性录入窗口

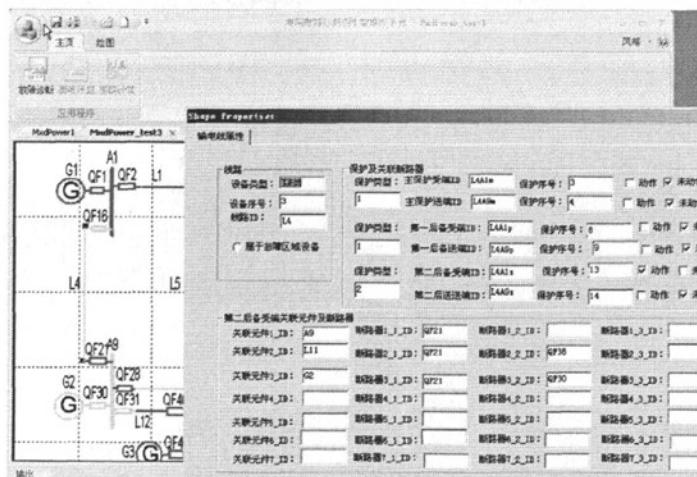


图 5.9 输电线路状态及关联属性录入窗口

(2) 识别故障区域

通过遍历故障信息数据库，获得断路器状态，根据跳闸断路器形成的故障区域中，需要进行故障诊断的元件为 G2、A9、L4、L11，依次编号为 ($s_1 \sim s_4$)，6个断路器 QF16、QF21、QF28、QF29、QF30、QF31 依次编号为 ($c_1 \sim c_6$)，16个保护 (G2m、A9m、L4A1m、L4A9m、L11A10m、L11A9m、G2p、L4A1p、L4A9p、L11A10p、L11A9p、G2s、L4A1s、L4A9s、L11A9s、L11A10s) 依次编号为 ($r_1 \sim r_{16}$)。

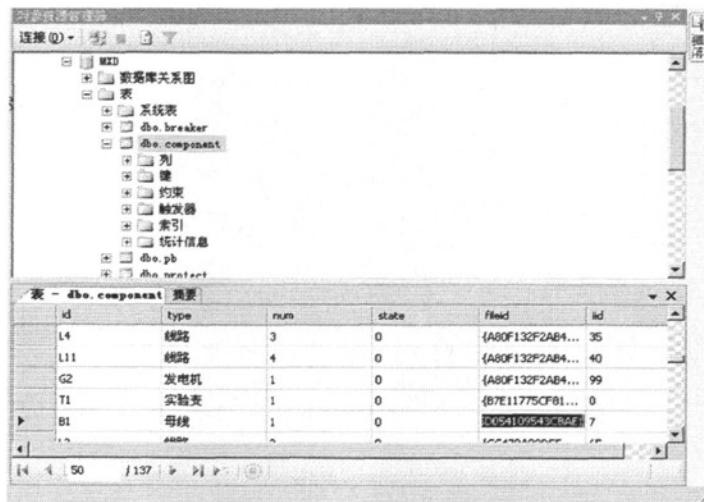


图 5.10 故障元件在数据库中的存储

(3) 确定故障区域内保护和断路器动作逻辑

以下表5.1、表5.2和表5.3分别给出这四个元件的保护动作原理。

表 5.1 故障区域元件主保护动作原理表

| 保护名 | 动作原理 |
|---------|----------------------------------------|
| G2m | G2 发生故障时, G2m 动作跳开 QF30 |
| A9m | A9 发生故障时, A9m 动作跳开 QF21、QF31、QF28、QF31 |
| L4A1m | L4 发生故障时, L4A1m 动作跳开 QF16 |
| L4A9m | L4 发生故障时, L4A9m 动作跳开 QF21 |
| L11A10m | L11 发生故障时, L11A10m 动作跳开 QF29 |
| L11A9m | L11 发生故障时, L11A9m 动作跳开 QF28 |

表 5.2 故障区域元件第一后备保护动作原理表

| 保护名 | 动作原理 |
|---------|----------------------------------------------|
| G2p | G2 故障时, QF30 在主保护动作时段未跳开, G2p 动作跳开 QF30 |
| L4A1p | L4 故障时, QF16 在主保护动作时段未跳开, L4A1p 动作跳开 QF16 |
| L4A9p | L4 故障时, QF21 在主保护动作时段未跳开, L4A9p 动作跳开 QF21 |
| L11A10p | L11 故障时, QF29 在主保护动作时段未跳开, L11A10p 动作跳开 QF29 |
| L11A9p | L11 故障时, QF28 在主保护动作时段未跳开, L11A9p 动作跳开 QF28 |

表 5.3 故障区域元件第二后备保护动作原理表

| 保护名 | 动作原理 |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| G2s | A9 故障且 QF30 在 A9 主保护动作时段内未跳开, 那么 G2s 动作跳开 QF30, 保护发电机 G2; 或者 L4 故障且 QF21 在 L4 的主保护、第一后备保护动作时段未跳开, 此时故障沿 A9 蔓延, 为保护发电机 G2, 由 G2s 动作跳开 QF30; 或者 L11 故障且 QF29 在 L11 的主保护、第一后备保护动作时段未跳开, 此时故障沿 A9 蔓延, 为保护发电机 G2, 由 G2s 动作跳开 QF30。 |
| L4A1s | A9 故障且 QF21 在 A9 主保护动作时段内未跳开, 那么 L4A1s 动作跳开 QF16, 防止故障沿 L4 蔓延; 或者 L11 故障且 QF29 在 L11 的主保护、第一后备保护动作时段未跳开, 此时故障沿 A9 蔓延, 波及 L4, 此时为防止故障沿 L4 蔓延, 由 L4A1s 动作跳开 QF16; 或者 G2 故障且 QF30 在其主保护、第一后备保护动作时段内未跳开, 此时故障沿 A9 蔓延, 波及 L4, 此时为防止故障沿 L4 蔓延, 由 L4A1s 动作跳开 QF16。 |
| L4A9s | 在故障区域内无关联元件。 |
| L11A10s | A9 故障且 QF28 在 A9 主保护动作时段内未跳开, 那么 L11A10s 动作跳开 QF29, 防止故障沿 L4 蔓延; 或者 L4 故障且 QF21 在 L4 的主保护、第一后备保护动作时段未跳开, 此时故障沿 A9 蔓延, 波及 L11, 此时为防止故障沿 L11 蔓延, 由 L11A10s 动作跳开 QF29; 或者 G2 故障且 QF30 在其主保护、第一后备保护动作时段内未跳开, 此时故障沿 A9 蔓延, 波及 L11, 此时为防止故障沿 L11 蔓延, 由 L11A10s 动作跳开 QF29。 |
| L11A9s | 在故障区域内无关联元件。 |

结合保护动作原理及关联分析, 形成如下实际优化的具体目标函数公式。

(4) 确定各个保护和断路器实际状态向量

根据警报条件可以从故障诊断图形数据库中读出保护、断路器的实际状

态向量。断路器实际状态向量如式(5-1)：

$$C = [c_1, c_2, c_3, c_4, c_5, c_6] = [1, 0, 0, 1, 1, 1] \quad (5-1)$$

保护实际状态向量如式(5-2)：

$$R = [r_1, r_2, \dots, r_{16}] = [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0] \quad (5-2)$$

(5) 确定保护和断路器期望状态向量

主保护的期望状态如式(5-3)所示。

$$\begin{bmatrix} r_1^*(S) = s_1 \\ r_2^*(S) = s_2 \\ r_3^*(S) = s_3 \\ r_4^*(S) = s_3 \\ r_5^*(S) = s_4 \\ r_6^*(S) = s_4 \end{bmatrix} \quad (5-3)$$

第一后备保护的期望状态如式(5-4)所示。

$$\begin{bmatrix} r_7^*(S) = s_1(1-r_1) \\ r_8^*(S) = s_3(1-r_3) \\ r_9^*(S) = s_3(1-r_4) \\ r_{10}^*(S) = s_4(1-r_5) \\ r_{11}^*(S) = s_4(1-r_6) \end{bmatrix} \quad (5-4)$$

第二后备保护的期望状态如式(5-5)所示：

$$\begin{bmatrix} r_{12}^*(S) = 1 - [1 - s_2(1 - c_5)][1 - s_4(1 - c_3)(1 - c_5)][1 - s_3(1 - c_2)(1 - c_5)] \\ r_{13}^*(S) = 1 - [1 - s_2(1 - c_2)][1 - s_4(1 - c_3)(1 - c_2)][1 - s_1(1 - c_2)(1 - c_5)] \\ r_{14}^*(S) = 0 \\ r_{15}^*(S) = 1 - [1 - s_2(1 - c_3)][1 - s_3(1 - c_3)(1 - c_2)][1 - s_1(1 - c_3)(1 - c_5)] \\ r_{16}^*(S) = 0 \end{bmatrix} \quad (5-5)$$

由保护触发的断路器的期望状态如式(5-6)所示：

$$\left. \begin{aligned} c_1^*(S, R) &= \max \{r_3^* r_3, r_8^* r_8, r_{13}^* r_{13}\} \\ c_2^*(S, R) &= \max \{r_2^* r_2, r_4^* r_4, r_9^* r_9, r_{14}^* r_{14}\} \\ c_3^*(S, R) &= \max \{r_2^* r_2, r_6^* r_6, r_{11}^* r_{11}, r_{16}^* r_{16}\} \\ c_4^*(S, R) &= \max \{r_5^* r_5, r_{10}^* r_{10}, r_{15}^* r_{15}\} \\ c_5^*(S, R) &= \max \{r_1^* r_1, r_2^* r_2, r_7^* r_7, r_{12}^* r_{12}\} \\ c_6^*(S, R) &= r_2^* r_2 \end{aligned} \right\} \quad (5-6)$$

(6) 设置遗传算法参数

种群规模: $m_nPopSize = 40$;

染色体长度: $m_nChromLength = 4$;

最大进化代数: $m_nMaxGeneration = 10$;

交叉概率: $m_PCross = 0.9$;

变异概率: $m_PMutation = 0.01$ 。

算法策略: 选择算子采用轮盘赌方法, 随机产生一个[0, 1]之间的值, 作为进行轮盘赌选择的选择指针, 个体适应度除以累计适应度为选择概率。交叉算子采用单点交叉方式。

(7) 利用遗传算法求解

得到如图5.11给出的诊断结果为: 发电机G2, 母线A9故障。

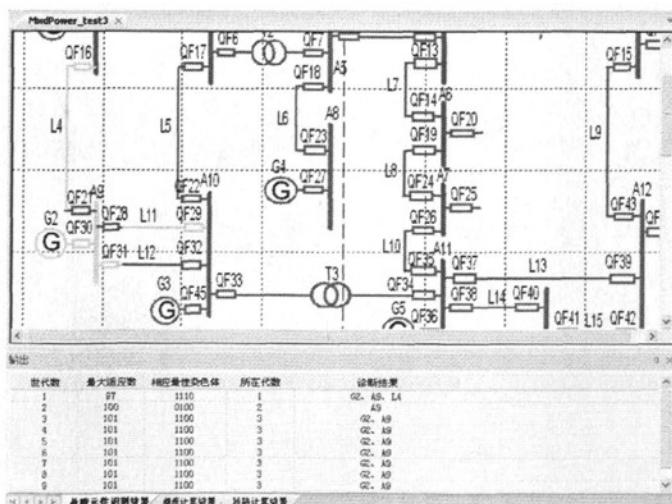


图 5.11 线路系统故障诊断图

(8) 程序运行结果及分析

根据保护和断路器的报警信号以及诊断结果，可以分析得知，发电机G2发生故障，G2主保护G2m动作，发出指令跳开断路器QF30，母线A9发生故障时，A9主保护A9m动作，发出指令跳开断路器QF21、QF28、QF30、QF31。但由于母线主保护未能正常跳开断路器QF21、QF28，导致故障蔓延至L4、L11，线路L4、L11第二后备保护L4A1s、L11A10s动作将断路器QF16、QF29跳闸，避免事故进一步扩大。此中情况是发电机、母线同时发生故障，并且断路器拒动故障情况。

5.4.2 平台测试算例2

验证发电机故障诊断的正确性，对发电机第二后备动作切除与故障区域的断路器连接情况进行了如下测算。

在线路系统图中，断路器QF21、QF28、QF30、QF31跳闸断路器形成的故障区域如图5.12所示，需要进行故障诊断的元件为G2、A9，依次编号为(S₁、S₂)，6个断路器QF21、QF28、QF30、QF31依次编号为(C₁~C₄)，4个保护(G2m、A9m、G2p、G2s)依次编号为(r₁~r₄)。表5.4给出了元件的保护动作原理。

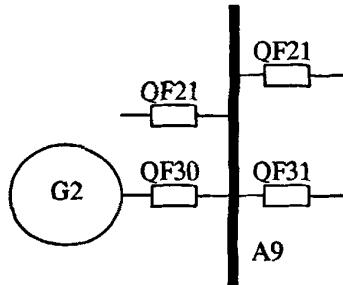


图5.12 测试系统故障区域图

表 5.4 故障区域元件保护动作原理表

| 保护名 | 动作原理 |
|-----|------------------------------------------------|
| G2m | G2发生故障时，G2m动作跳开QF30 |
| A9m | A9发生故障时，A9m动作跳开QF21、QF31、QF28、QF30 |
| G2p | G2故障时，QF30在主保护动作时段未跳开，G2p动作跳开QF30 |
| G2s | A9故障且QF30在A9主保护动作时段内未跳开，那么G2s动作跳开QF30，保护发电机G2； |

结合保护动作原理及关联分析，形成如下实际优化的具体目标函数公式。
保护的期望状态如式（5-7）所示。

$$\left[\begin{array}{l} r_1^*(S) = s_1 \\ r_2^*(S) = s_2 \\ r_3^*(S) = s_1(1-s_1) \\ r_4^*(S) = 1 - [1 - s_1(1 - c_3)] \end{array} \right] \quad (5-7)$$

由保护触发的断路器的期望状态如式（5-8）所示：

$$\left[\begin{array}{l} c_1^*(S, R) = r_2^* r_2 \\ c_2^*(S, R) = r_2^* r_2 \\ c_3^*(S, R) = \max \{r_1^* r_1, r_2^* r_2, r_3^* r_3, r_4^* r_4\} \\ c_4^*(S, R) = r_2^* r_2 \end{array} \right] \quad (5-8)$$

程序运行结果及分析：

本文通过元件的四种状态组合计算了目标函数值。

当 $(S_1, S_2) = (0, 0)$ 时，目标函数值为 100；

当 $(S_1, S_2) = (0, 1)$ 时，目标函数值为 102；

当 $(S_1, S_2) = (1, 0)$ 时，目标函数值为 99；

当 $(S_1, S_2) = (1, 1)$ 时，目标函数值为 100；

得出目标函数值最大情况是母线 A9 发生故障。从测算系统保护和断路器的报警信号以及诊断结果，皆可以分析得知，母线 A9 发生故障，A9 主保护 A9m 动作，正常情况下断路器 QF21、QF28、QF30、QF31 应该跳闸，但是由于 A9 与发电机 G2 之间的断路器 QF30 拒动，引起发电机 G2 第二后备保护 G2s 动作令 QF30 跳闸。此中情况为母线发生故障，并且断路器拒动情况下发电机第二后备保护动作，保护发电机。

5.5 本章小结

本章介绍了电力系统图形数据库平台的开发。探讨了设备图元类的设计与管理。给出了电力系统故障诊断信息数据库的 E-R 模型和表结构，介绍了图形与数据库记录关联方法。最后，通过 IEEE 的五机系统故障诊断过程，验证图形数据库与故障诊断程序的配合效果。

结 论

电力系统故障诊断应用技术要基于电力系统故障诊断模型与现代计算机应用技术。本文使用遗传算法来实现电力系统故障元件的快速诊断。在此过程中，研究了电力系统故障诊断解析模型，设计了基于遗传算法的诊断程序，开发了与故障诊断程序配合良好的电力系统图形数据库平台。主要研究结果如下：

(1) 总结了电力系统故障类型、研究了故障征兆信息，分析了电力系统故障处理方式。并在研究电力系统继电保护原理及故障分析过程的基础上，给出了利用保护和断路器动作信息识别发生故障元件的电力系统故障诊断模型。通过实例，归纳总结了包括发电机故障、变压器、母线和输电线路等四类电力设备的保护动作原理。

(2) 将电力系统故障诊断问题转化为0-1整数规划问题，确立了通过遗传算法搜索电力系统故障诊断函数最优解的方式。通过面向对象程序设计中的多态性和泛型算法的有效结合，设计了电力系统故障诊断程序。通过故障实例验证诊断模型和算法程序在对发电机、母线、线路和变压器共同进行故障识别的正确性。

(3) 开发了电力系统故障诊断图形数据库平台，实现了母线、断路器、发电机和输电线路等电气设备矢量图形的绘制和拖放等功能。设计了合理的电力系统故障诊断关系数据库，通过与图形平台的映射，实现了方便的数据关联。最后，采用了IEEE五机系统故障实例进行了诊断验证，获得了正确的结论。

由于时间和个人能力等原因，本文工作还存在一些不足，具体有：

在主站诊断模块的算法模型中，没有考虑自动重合闸情况等情况。其次针对不同的电力系统网络，还需要研究通过程序自动形成故障目标函数，实现大量的数据库数据关联信息查询，程序代码还需完善，实现关联矩阵进行运算。

这些工作有待在今后的工作中去完善，这不能说不是一个遗憾。

参考文献

- [1] 徐青山. 电力系统故障诊断及故障恢复[M]. 北京: 中国电力出版社, 2007
- [2] 文福拴, 韩祯祥, 田磊, 史觉玮, 张怀宇. 基于遗传算法的电力系统故障诊断的解析模型和方法 第一部分 模型与方法[J]. 电力系统及其自动化学报, 1998(10): 1~7 页
- [3] C.A.PorotPaPas, K.PPsaltiasr.An ExPert System for Substation Fault diagnosis and Alarm Processing[J].IEEE Trans Power Delivery,1991,06
- [4] Sun Y, Jinag H.Fault Snyhtetie Reeognition for an EHV Transmission Line Using a Group of Neuarl Netwokrs with a Time-space PorPerty[J], IEE Proeeedings: Geneartion, Transmission and Distribution, 145(3), 1998
- [5] 徐文, 王大忠, 周泽存等. 模糊辨识在电力设备故障诊断中的应用[J]. 电力系统自动化, 1996, 20(3): 15~22 页
- [6] 王蓓蓓, 郭基伟, 谢敬东. 基于模糊 Pertil 网的知识表示方法在变压器故障诊断专家系统中的应用[J]. 电力系统及其自动化学报, 2003, 5(11): 21~25 页
- [7] R.Khosla , Q.Li. Multi-layered Multi-Agent Architecture with Fuzzy Application in Electrical Power Systems.Proceedings of the 2002 IEEE International Conference on Fuzzy Systems.2002(1):209-214p
- [8] S.D.J.McArthur. Multi-Agent Systems for Diagnostic and Condition Monitoring Applications. Power Engineering Society General meeting.
- [9] 张明亮. 河南电网稳态监测及故障诊断专家系统[D]. 北京: 华北电力大学, 2001
- [10] 赵冬梅, 张东英, 徐开理.分层分布式电网故障诊断专家系统设计[J]. 现代电力, 2001, 18 (3) : 41~46 页
- [11] 毛一之.电力系统故障诊断[D]. 天津: 河北工业大学, 2007
- [12] 文福拴, 韩祯祥, 田磊, 史觉玮, 张怀宇. 基于遗传算法的电力系统故障诊断的解析模型和方法 第二部分 软件实现[J]. 电力系统及其自动

- 化学报, 1998(10): 8~14 页
- [13] 文福拴, 韩祯祥, 田磊, 史觉玮, 张怀宇. 基于遗传算法的电力系统故障诊断的解析模型和方法 第三部分 浙江电力系统 EMS 信息获取[J]. 电力系统及其自动化学报, 1998(11): 13~18 页
- [14] 孟祥萍, 刘春玲, 耿卫星. 利用保护、断路器状态信息进行电网故障的诊断[J]. 宁夏工程技术, 2006, 5 (3) : 261~264 页
- [15] 高翔, 张沛超, 章坚民. 电网故障信息系统应用技术[M]. 北京: 中国电力出版社, 2006
- [16] 张旭. 地区电网故障诊断系统的实用化研究[D]. 北京: 华北电力大学, 2004
- [17] 李丽娇, 齐云秋. 电力系统继电保护[M]. 北京: 中国电力出版社, 2005
- [18] 毕天姝, 杨奇逊. 人工智能技术在输电网络故障诊断中的应用述评[J]. 电力系统及其自动化学报, 2000, 24(2): 28~34 页
- [19] 巩文科, 李心广. 基于 BP 神经网络与专家系统的故障诊断系统[J]. 计算机工程, 2007, 33(8): 14~16 页
- [20] 王小平, 曹立明著. 遗传算法: 理论、应用与软件实现[M]. 陕西: 西安交通大学出版, 2002
- [21] 王广延, 吕继绍等. 电力系统继电保护原理与运行分析上、下册[M]. 北京: 中国电力出版社, 1998
- [22] Zhang Q, Han Z X, Wen F S. A New Approach for Fault Diagnosis in Power Systems Based on Rough Set Theory. Operation and Management(APSCOM'97).Hong Kong:1997:597-602p
- [23] Ogilvie T, Swidenbank E. Use of Data Mining Techniques in the Performance Monitoring and Optimization of a Thermal Power Plant. IEE Coloquium(Digest).1998
- [24] 桂强. 数据挖掘及信息融合技术在电网故障信息系统中的应用[D]. 上海: 上海交通大学硕士学位论文, 2007
- [25] 陈艳春. 决策支持系统设计与开发[M]. 北京: 中国铁道出版社, 2007.08
- [26] 蔡雨阳, 黄岩, 李农, 黄丽华. MIS、DSS 和 EIS 的数据集成[M]. 北京: 管理工程学报, 2001

- [27] 周晓峰, 王志坚. 基于多 Agent 的分布式异构数据源访问机制研究[J]. 北京: 计算机工程与设计, 2003, 24(4): 5~8 页
- [28] 蒋伟进, 王璞. 基于 MAS 的复杂系统分布式求解策略与推理研究[J]. 北京: 计算机研究与发展, 2006, 4(39): 1615-1623 页
- [29] 王强, 皮德常, 李维奇, 吕军. 基于 Agent 和数据挖掘的分布式信息审计平台[J]. 计算机技术与发展, 2006: 141~146 页
- [30] 丁士长, 翁汉勋, 毛鹏, 陈久林. 基于 MAS 的电网故障诊断系统的体系结构[J]. 继电器, 2006, 34(21): 24~28 页
- [31] 董泰福, 孙雅明. 可缩放矢量图形技术[J]. 北京: 华北电力大学学报, 2007, 34(4): 38~40 页
- [32] 王志海. 数据仓库 (原书第四版) [M]. 北京: 机械工业出版社, 2006
- [33] 于之虹, 郭志忠. 数据挖掘与电力系统[M]. 北京: 冶金工业出版社, 2003
- [34] 安淑芝. 数据仓库与数据挖掘[M]. 北京: 清华大学出版社, 2007
- [35] 布启敏, 舒利文. 数据仓库[M]. 北京: 清华大学出版社, 2007
- [36] 李志刚, 马刚著. 数据仓库与数据挖掘的原理及应用[M]. 北京: 高等教育出版社, 2008
- [37] 尹祝芳, 焦贤龙. 数据挖掘原理与应用: SQL Server 2005 数据库[M]. 北京: 清华大学出版社, 2007
- [38] 闫雷鸣, 冯飞. 数据仓库工具箱——面向 SQL Server 2005 和 Microsoft 商业智能工具集[M]. 北京: 清华大学出版社, 2007
- [39] 吕雪峰. 基于遗传算法的电力系统故障诊断[D]. 黑龙江: 大庆石油学院硕士学位论文, 2006
- [40] 冯博琴, 冯岚, 薛涛, 崔舒宁. 面向对象分析与设计 (原书第 2 版) [M]. 北京: 机械工业出版社, 2003
- [41] 张龙祥. UML 与系统分析设计[M]. 北京: 人民邮电出版社, 2007
- [42] 国刚, 周峰著. UML 与 Rational Rose2003 软件工程统一建模原理与实践[M]. 北京: 电子工业出版社, 2007
- [43] Stanley B.Lippman, Josee Lajoie, Barbara E.Moo. 李师贤, 蒋爱军, 梅晓勇, 林瑛. C++ Primer 中文版 (第 4 版) [M]. 北京: 人民邮电出版社,

2006

- [44] 叶至军. C++ STL 开发技术引导[M]. 北京: 人民邮电出版社, 2008
- [45] 潘爱民, 陈铭, 邹开红. Effective STL 中文版: 50 条有效使用 STL 的经验[M]. 北京: 清华大学出版社, 2007
- [46] 李春葆. 数据库原理与应用——基于 SQL Server 2005[M]. 北京: 清华大学出版社, 2008
- [47] 辛冬梅. 实现 Visual Studio 风格的窗口[J]. 电脑编程技巧与维护, 2003, 5: 18~20 页
- [48] 林济铿, 覃岭, 罗萍萍. 基于 Visual Graph 的电力图形系统开发[J]. 电力系统及其自动化, 2005, (29)15: 3~76 页
- [49] 董泰福, 文福拴, 廖志伟. 可缩放矢量图形技术在电力系统故障诊断软件包中的应用[J]. 华北电力大学学报, 2007, 34(4): 38~42 页
- [50] 蒋春勇. Windows 图形编程中的坐标系与 GDI 映射方式[J]. 北京: 电脑知识与技术: 学术交流, 2006
- [51] 陈建春. Visual C++ 开发 GIS 系统[M]. 北京: 电子工业出版社, 2001
- [52] 罗建新. 电网图库一体化平台研究与实现[M]. 武汉理工大学硕士学位论文, 2005
- [53] 孙鑫著. VC++深入详解[M]. 北京: 电子工业出版社, 2006
- [54] 杨秀章. COM 技术内幕——微软组件对象模型[M]. 北京: 清华大学出版社, 1999
- [55] 潘爱民. COM 原理与应用[M]. 北京: 清华大学出版社, 2001
- [56] 北京博彦科技发展有限责任公司. MFC Windows 程序设计[M]. 北京: 清华大学出版社, 2007
- [57] 张锦爱, 刘前进. 基于 Multi-Agent 的电网协同保护[J]. 高电压技术, 2007: 74~77 页
- [58] 张一倩, 石嘉川. 标准模板库 STL 及应用[J]. 科技信息, 2008, 4: 75~77 页
- [59] 曾祥辉, 宋玮, 邓健, 杨以涵. 面向对象的电力图形系统的分析和设计[J]. 继电器, 2004, 32(5): 36~39 页
- [60] 田保军. 基于 C++ 的 STL 编程思想[J]. 信息技术, 2007, 12: 112~113

页

- [61] 吴文传, 张伯明等. 支持 SCADA/PAS/DTS 一体化的图形系统[J]. 电力系统及其自动, 2001, 25(5): 45~48 页
- [62] 蔡士杰. 计算机图形学[M]. 北京: 电子工业出版社, 2005
- [63] 武森, 高学东. 数据仓库与数据挖掘[M]. 北京: 冶金工业出版社, 2003
- [64] 朱玉全, 杨鹤标, 孙蕾. 数据挖掘技术[M]. 南京: 东南大学出版社, 2006
- [65] 于之虹, 郭志忠. 数据挖掘与电力系统[M]. 北京: 冶金工业出版社, 2003
- [66] 杨宜东, 孙志挥. 基于时间序列的电力系统故障诊断规则挖掘[J]. 计算机工程与应用, 2003, 30(17): 223~224 页

攻读硕士学位期间发表的论文和取得的科研成果

- 1、 兰海, 毛晓东. STL 在电网故障诊断遗传算法中的应用研究. 2009. 应用科技. (已录用)

致 谢

经过两年多的学习和研究，我已经对电力系统这个原本陌生的领域有了一定程度的认知，特别是对故障诊断已经产生了浓厚的兴趣，也取得了一定的成果。这一切，除了自身的努力外，离不开周围的老师和同学对我的指导和帮助。在此，我对他们表示衷心的感谢！

首先要感谢我的导师兰海副教授。本论文是在导师的关怀指导下完成的，兰老师渊博知识、治学态度严谨、具有崇高的敬业精神，这些都给我留下了深刻的印象，使我受益匪浅。兰老师对我在学校的生活也给予了深切的关怀和爱护，使我十分感激。在此论文完成之际，我首先向导师表示最诚挚的谢意。

在课题研究过程中，张利军老师、黄曼磊老师、刘宏达老师和田凯老师给予了深切关怀和精心指导，在此，我向四位老师致以由衷的感谢。

我也衷心感谢实验室的同学们在课题研究过程中以及日常生活中给予的帮助和支持。

最后，我还要感谢家人对我的支持，感谢他们对我的鼓励与关心。

附录

个体类

```
class CIndividual
{
public:
    std::vector<bool> Chrom;
    int fitness;
    CMxdPowerDB m_mxpowerdb;
public:
    CIndividual(void);
    CIndividual(int n_ChromLength);
    ~CIndividual(void);
    int Fitness(void);
    int CalFitness(void);
};

CIndividual::CIndividual(int n_ChromLength)
{
    for (int n = 0; n != n_ChromLength; ++n)
    {
        Chrom.push_back( 0 );
    }
    fitness = 0;
}

int CIndividual::Fitness(void)
{
    return fitness;
}
```

遗传算法类

```
class CGA
{
private:
    std::vector<CIndividual> OldPop;
    std::vector<CIndividual> NewPop;
    std::vector<CIndividual> bestchrom;
    int m_SumFitness;
    float m_PCross;
    float m_PMutation;
    int m_nPopSize;
    int m_nChromLength;
    int m_nNowGeneration;
    int m_nMaxGeneration;
public:
    CGA(int n_ChromSize);
    ~CGA(void);
    void output(CIndividual best);
    inline bool ChangeOrNo(float Probability);
    inline float SetRandomValue(void);
```

```

void Sumfitness(std::vector<CIndividual>::iterator popbeg,
                std::vector<CIndividual>::iterator popend);
void Statistics(std::vector<CIndividual>::iterator popbeg,
                std::vector<CIndividual>::iterator popend);
int Select(void);
void Cross(CIndividual father, CIndividual mother,
            CIndividual &boy, CIndividual &girl);
void Mutation(CIndividual &individual);
void Evolution(void);
};

CGA:: CGA(int n_ChromSize)
{
    CIndividual individual(n_ChromSize);
    m_PMutation = 0.01;
    m_PCross = 0.9;
    m_nPopSize = 40;
    m_nMaxGeneration = 10;
    for (int m = 0; m != m_nPopSize; ++m)
    {
        OldPop.push_back(individual);
        NewPop.push_back(individual);
    }
    for (int m = 0; m != m_nMaxGeneration; ++m)
    {
        bestchrom.push_back(individual);
    }
    int i=1;
    srand((unsigned)time(NULL));
    for (std::vector<CIndividual>::iterator iter = OldPop.begin();
                     iter != OldPop.end(); ++iter)
    {
        for (std::vector<bool>::iterator biter = (*iter).Chrom.begin();
                         biter != (*iter).Chrom.end(); ++biter)
        {
            *biter = (bool)( rand() % 2 );
            (*iter).s[i++] = (int)(*biter);
        }
        (*iter).CalFitness();
    }
    m_SumFitness = 0;
    Statistics(OldPop.begin(), OldPop.end());
}
int CGA::Select(void)
{
    int index = 0;
    float PSum = 0;
    float SelectPoint = 0;
    SelectPoint = SetRandomValue();
    for (index = 0; (index != OldPop.size()) && (PSum < SelectPoint); ++index)
        PSum += (OldPop.at(index).Fitness()) / m_SumFitness;
    if(index == 0)

```

```

    {
        return (index);
    }
    else
    {
        return (index - 1);
    }
}
void CGA::Cross(CIndividual father, CIndividual mother,
                 CIndividual &boy,   CIndividual &girl)
{
    int nCrossSite = 0;
    if(ChangeOrNo(m_PCross))
    {
        nCrossSite = rand() % (m_nChromLength - 1);
        swap_ranges(father.Chrom.begin() + nCrossSite, father.Chrom.end(),
                    mother.Chrom.begin() + nCrossSite);
    }
    copy(father.Chrom.begin(), father.Chrom.end(), boy.Chrom.begin());
    copy(mother.Chrom.begin(), mother.Chrom.end(), girl.Chrom.begin());
}
void CGA::Mutation(CIndividual &individual)
{
    for (std::vector<bool>::size_type i = 0; i != individual.Chrom.size(); ++i )
    {
        if (ChangeOrNo(m_PMutation))
        {
            individual.Chrom.at(i).flip();
        }
    }
}
void CGA::Evolution(void)
{
    int nMate1 = 0;
    int nMate2 = 0;
    for (int i = 0; i != m_nPopSize; i += 2)
    {
        nMate1 = Select();
        nMate2 = Select();
        Cross(OldPop.at(nMate1), OldPop.at(nMate2), NewPop.at(i), NewPop.at(i + 1));
        Mutation(NewPop.at(i));
        for(int m=0; m < m_nChromLength; m++)
            NewPop.at(i).s[m+1] = (int)(NewPop.at(i).Chrom.at(m));
        NewPop.at(i).CalFitness();
        Mutation(NewPop.at(i + 1));
        for(int n=0; n < m_nChromLength; n++)
            NewPop.at(i).s[n+1] = (int)(NewPop.at(i).Chrom.at(n));
        NewPop.at(i + 1).CalFitness();
    }
    OldPop = NewPop;
    Statistics(OldPop.begin(), OldPop.end());
}

```

```

}

inline float CGA::SetRandomValue(void)
{
    float RandomProbability;
    RandomProbability = (float(rand() % 100)) / 100;
    return RandomProbability;
}

inline bool CGA::ChangeOrNo(float Probability)
{
    if(SetRandomValue() <= Probability)
    {
        return true;
    }
    else
    {
        return false;
    }
}

void CGA::SumFitness(std::vector<CIndividual>::iterator popbeg,
                     std::vector<CIndividual>::iterator popend)
{
    for(std::vector<CIndividual>::iterator iter = popbeg; iter != popend; ++iter)
    {
        m_SumFitness = m_SumFitness + (*iter).Fitness();
    }
}

void CGA::Statistics(std::vector<CIndividual>::iterator popbeg,
                     std::vector<CIndividual>::iterator popend)
{
    static int i = 0;
    SumFitness(pobeg, popend);
    bestchrom.at(i) = *popbeg;

    for(std::vector<CIndividual>::iterator iter = popbeg; iter != popend; ++iter)
    {
        if ((*iter).Fitness() > bestchrom.at(i).Fitness())
        {
            bestchrom.at(i) = *iter;
        }
    }
    output( bestchrom.at(i) );
    ++i;
}
}

```