电路 CAD 中 PCB 的自动布线算法研究

摘 要

首先介绍了计算机辅助设计的发展概况和电子电路 CAD 软件的基本类型和基本构成。接着阐述了自动布线技术的发展和分类，对自动布线理论进行综合分析研究。

给出了一种 MD 模型上的布线算法——智能识点法。首先，提出了 MD 模型上的广义线段的概念；其次，依据人工智能的思维给出绕障点的定义方法；最后，以绕障点和待布点组成布线点集合，以布线点集合中的所有点为顶点构造带边权值的完全图，将布线问题转化为在带边权值的完全图中求布线点间的最短路径，从而得到最优布线路径。实验证明算法的性能良好。

关键词： MD 布线模型，广义线段，绕障点，布线算法，完全图

STUDY ON PRINTED CIRCUIT BOARD AUTO ROUTING IN CIRCUIT CAD

ABSTRACT

This paper introduces the general developmental situation of the Computer-Aided Design (CAD) and the class and primary structure of the circuit CAD software. The developmental situation and class of the auto routing are also described, the theory of auto routing is studied synthetically.

A automatic routing algorithm——Intelligent Discern Points Algorithm is presented in Manhattan-diagonal(MD) model with tracks in horizontal,vertical,and ±45 directions.First,the concept of generalized line segment is presented in MD model.Second,by means of artificial intelligence theory,the way of deciding the position of detouring points is proposed.Finally,a set of point which is made up of detouring points and pins is formed.By using the points of the set as the vertices ,a weighted complete graph is formed.The question of routing is transformed into finding shortest-path problem in the weighted complete graph.Thereby,the optimization of routing is found. It is proved by experimentation that the algorithm performance is efficient.

Key Words:Manhattan-diagonal model; generalized line segment; detouring point; routing algorithm; complete graph

第一章 绪 论

1.1 电子电路CAD技术

1.1.1 电子电路CAD技术的概念及特点

计算机辅助设计（CAD：Computer Aided Design）是指以计算机系统作为主要技术手段来生成和运用各种数字信息与图形信息，帮助设计人员从事产品的开发、修改、分析、和优化设计的一门技术。CAD技术本身是一项综合性的、技术复杂的系统工程，涉及许多学科领域，如计算机科学与工程、计算数学、计算力学、几何造型、计算机图形学、数据结构和数据库、仿真技术、人工智能等。

电子电路CAD 技术是指以计算机硬件和系统软件为基本工作平台，继承和借鉴前人在电路和系统、图论、拓扑逻辑优化和人工智能理论等多学科的最新科技的成果而研制成的电子电路CAD通用支撑软件和应用软件包。其目的在于帮助电子设计工程师开发新的电子系统与电路、IC、PCB（印刷电路板）、FPGA（现场可编程门阵列）、CPLD（复杂可编程逻辑器件）等产品。实现在计算机上调用元器件库、连线画图、编制激励信号文件、确定跟踪点、调用参数库以及模拟程序等手段去设计电路

电子电路设计任务：电子电路的设计就是根据给定的功能和特性指标要求（设计要求），采取一定的方法来确定采用什么样的电路拓扑结构及电路中的各个元器件应该采用什么样的参数，进一步将设计好的电路转换成印刷电路板图设计。电子电路设计工作流程如图1.1所示。

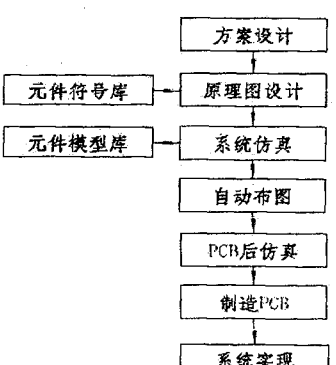


图1.1 电路设计工作流程图

电子电路设计中采用 CAD 技术有很多优点，比如缩短了设计周期，节省设计费用，提高设计质量，共享设备资源以及很强的数据处理能力。随着电子技术的发

第一章 绪 论

1.1 电子电路CAD技术

1.1.1 电子电路CAD技术的概念及特点

计算机辅助设计（CAD：Computer Aided Design）是指以计算机系统作为主要技术手段来生成和运用各种数字信息与图形信息，帮助设计人员从事产品的开发、修改、分析、和优化设计的一门技术。CAD技术本身是一项综合性的、技术复杂的系统工程，涉及许多学科领域，如计算机科学与工程、计算数学、计算力学、几何造型、计算机图形学、数据结构和数据库、仿真技术、人工智能等。

电子电路CAD 技术是指以计算机硬件和系统软件为基本工作平台，继承和借鉴前人在电路和系统、图论、拓扑逻辑优化和人工智能理论等多学科的最新科技的成果而研制成的电子电路CAD通用支撑软件和应用软件包。其目的在于帮助电子设计工程师开发新的电子系统与电路、IC、PCB（印刷电路板）、FPGA（现场可编程门阵列）、CPLD（复杂可编程逻辑器件）等产品。实现在计算机上调用元器件库、连线画图、编制激励信号文件、确定跟踪点、调用参数库以及模拟程序等手段去设计电路

电子电路设计任务：电子电路的设计就是根据给定的功能和特性指标要求（设计要求），采取一定的方法来确定采用什么样的电路拓扑结构及电路中的各个元器件应该采用什么样的参数，进一步将设计好的电路转换成印刷电路板图设计。电子电路设计工作流程如图1.1所示。

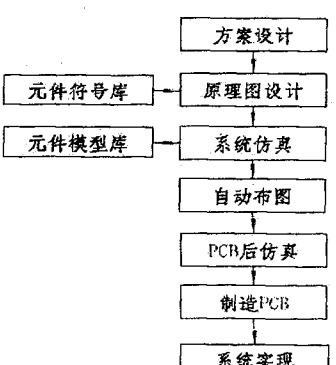


图1.1 电路设计工作流程图

电子电路设计中采用 CAD 技术有很多优点，比如缩短了设计周期，节省设计费用，提高设计质量，共享设备资源以及很强的数据处理能力。随着电子技术的发

展，需设计的电路越来越复杂，规模也越来越大，在这种情况下，离开 CAD 技术几乎无法完成现代的电子电路设计任务。

1.1.2 电子电路 CAD 软件系统的基本构成

根据各个电子电路 CAD 软件的用途和设计任务的要求，通常在设计过程中所用的 CAD 软件系统主要包括以下几类软件：

（1）通用逻辑模拟软件：用来对一般数字电路进行模拟验证。该软件可根据用户提供的数字电路结构和所用基本逻辑单元的特性，模拟分析该电路的逻辑功能、延迟特性以及电路中是否存在冒险竞争情况等。

（2）通用电路模拟软件：用来对一般电子电路进行模拟验证。该软件可根据给出的电路拓扑结构和电路中所用的元器件参数，模拟分析该电路的直流、交流和瞬态等各种特性，并进而进行灵敏度分析、成品率模拟和最坏情况分析等特殊类型的分析。

（3）专用电路设计软件：专门用于某些特定类型电路的设计软件。例如由 PAL（可编程阵列逻辑）和 FPGA（现场可编程门阵列）等可编程器件构成的电路。与通用模拟软件相比，这类软件适用面窄，但是在其适用范围内功能则更强，往往还具有优化设计的功能。

（3）印刷电路板（PCB：Printed Circuit Board）布局布线软件：该类软件具有自动布局布线功能，一般只需用户进行少量的人工干预就可完成印刷电路板的设计任务。

（4）电路图绘制和后处理软件：为了运行上述四类软件，需要提供电路的拓扑结构。目前，常用方法是利用绘制电路图的计算机软件将电路图送入计算机，然后再调用相应的后处理程序，生成能完全表征电路拓扑结构的连接网表文件，作为上述几类软件的输入。同时后处理程序还可对绘出的电路图进行电连接规则检验和各种统计报表生成等。

（5）数据库建库软件：绘制电路图和运行印刷电路板布局布线软件时需调用多种型号的器件符号，运行通用逻辑模拟和电路模拟软件时也需要使用不同型号的特性参数值。这些器件符号和元器件参数值都是存放在专用的数据库中提供给用户的。此外电子电路 CAD 软件还提供有数据库建库软件，以便于用户在使

展，需设计的电路越来越复杂，规模也越来越大，在这种情况下，离开 CAD 技术几乎无法完成现代的电子电路设计任务。

1.1.2 电子电路 CAD 软件系统的基本构成

根据各个电子电路 CAD 软件的用途和设计任务的要求，通常在设计过程中所用的 CAD 软件系统主要包括以下几类软件：

（1）通用逻辑模拟软件：用来对一般数字电路进行模拟验证。该软件可根据用户提供的数字电路结构和所用基本逻辑单元的特性，模拟分析该电路的逻辑功能、延迟特性以及电路中是否存在冒险竞争情况等。

（2）通用电路模拟软件：用来对一般电子电路进行模拟验证。该软件可根据给出的电路拓扑结构和电路中所用的元器件参数，模拟分析该电路的直流、交流和瞬态等各种特性，并进而进行灵敏度分析、成品率模拟和最坏情况分析等特殊类型的分析。

（3）专用电路设计软件：专门用于某些特定类型电路的设计软件。例如由 PAL（可编程阵列逻辑）和 FPGA（现场可编程门阵列）等可编程器件构成的电路。与通用模拟软件相比，这类软件适用面窄，但是在其适用范围内功能则更强，往往还具有优化设计的功能。

（3）印刷电路板（PCB：Printed Circuit Board）布局布线软件：该类软件具有自动布局布线功能，一般只需用户进行少量的人工干预就可完成印刷电路板的设计任务。

（4）电路图绘制和后处理软件：为了运行上述四类软件，需要提供电路的拓扑结构。目前，常用方法是利用绘制电路图的计算机软件将电路图送入计算机，然后再调用相应的后处理程序，生成能完全表征电路拓扑结构的连接网表文件，作为上述几类软件的输入。同时后处理程序还可对绘出的电路图进行电连接规则检验和各种统计报表生成等。

（5）数据库建库软件：绘制电路图和运行印刷电路板布局布线软件时需调用多种型号的器件符号，运行通用逻辑模拟和电路模拟软件时也需要使用不同型号的特性参数值。这些器件符号和元器件参数值都是存放在专用的数据库中提供给用户的。此外电子电路 CAD 软件还提供有数据库建库软件，以便于用户在使

用软件时根据需要扩展数据库内容，添加新的器件符号等。

通常，微机级电子电路 CAD 软件系统如下图 1.2[1]所示。系统中每一部分既可根据需要单独使用，相互之间又有联系，共同组成一个完整的电子电路 CAD 系统。其中电路图绘制软件是该系统的基础，后处理程序是系统中联系各部分软件的纽带。

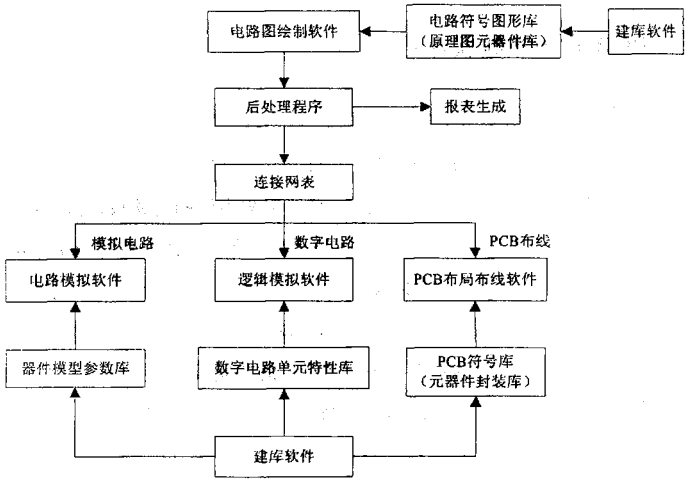


图 1.2 微机级电子电路 CAD 软件系统构成

电路 CAD 软件应具备以下条件[2]：

1. 有开发结构的框架环境：通过开放结构和二次开发工具，允许用户把自己的应用程序集成到软件环境中，和原有的工具一样处理设计数据。
2. 有友好的用户界面：提供的菜单形式应方便实用，便于人机对话，且具有键盘、宏命令、鼠标等多种访问命令的方法。
3. 软件功能强大：能提供图形调入、网表生成和 PCB 布局、布线等多种功能，自动化程度较高，支持从电路图输入到 PCB 生成全过程。
4. 软件的建库方法方便：对可能遇到扩充图形库和封装库的情况时，建库方法一定要方便、简单。
5. 与其他软件接口较好：便于与不同软件之间进行数据交换。
6. 升级发展有保证：软件开发公司应具有使软件更新升级的技术力量。

我国电路 CAD 软件市场，国外产品占据了大部分市场份额。从软件的使用率来看，最知名的供应商是 Protel（现已更名为 Altium），在中国大陆拥有 73%的用户，OrCAD，Synopsys 和 Cadence 各拥有较多的用户，PCAD, PADS 也有一些用户。而在该领域的国产软件，无论从技术上，功能上，还是从用户数量上，都难于和外国软件相争高下。

1.1.3 PCB 布局布线软件

PCB 是英文“Printed Circuit Board”的缩写，直译是印刷电路板，其含义是：以绝缘材料为基板加工成一定尺寸的板，上面至少有一个导电图形及所设计好的孔，以实现电子元器件之间的电气连接，这样的板称为印刷电路板，图 1.3[3]给出了 PCB 的实物图。1936 年，英国 Eisler 博士提出印制电路（Printed Circuit）这个概念，他首创在绝缘基板上全面覆盖金属箔，在其金属箔上涂上耐蚀刻油墨后，再将不需要的金属箔腐蚀掉的 PCB 制造基本技术，1942 年，Eisler 博士制造出世界第一块纸质层压绝缘基板，用于收音机的印制板。50 年代初，这种技术开始广泛应用，并迅速得到发展。印制电路是电子产品的关键电子互连件，无论在市场领域，应用数量和技术水平等各方面都占有极重要的地位并在快速发展。一方面印刷电路板是各种消费类电子产品和投资类电子产品的基本零组件，目前，消费类电子产品和投资类电子产品要求其所对应的电子机器具有“轻，薄，短，小”化，多样化，多功能，高速，高可靠，研制周期短和价廉等特点，这就促使所设计出的印刷电路板朝着小孔径，细线条，轻薄型的方向发展。另一方面在电子工业领域，随着现代科学技术日新月异地发展，大规模、超大规模集成电路的使用使印刷电路板的走线愈加精密和复杂。在这种情况下，传统的手工设计方式已越来越难以适应当前的形势，因此计算机来辅助设计电路板已成为电路板设计制作的必然趋势。

PCB 的制造过程[4, 5]简单来说，首先把集成电路用陶瓷或塑料管壳封装，然后再安装在印刷电路板上，目前的集成电路封装方法主要有三种：双列直插式 DIP、引腿阵列式 PGA 和表面封装（Suface Mounted Device ，SMD），封装好的器件的引腿插入、安放在印制电路板表面，这个过程叫布局，再根据布局测试布线的可能性，然后在各个元器件之间进行连线，实现元件之间的电气连接关系，

我国电路 CAD 软件市场，国外产品占据了大部分市场份额。从软件的使用率来看，最知名的供应商是 Protel（现已更名为 Altium），在中国大陆拥有 73%的用户，OrCAD，Synopsys 和 Cadence 各拥有较多的用户，PCAD, PADS 也有一些用户。而在该领域的国产软件，无论从技术上，功能上，还是从用户数量上，都难于和外国软件相争高下。

1.1.3 PCB 布局布线软件

PCB 是英文“Printed Circuit Board”的缩写，直译是印刷电路板，其含义是：以绝缘材料为基板加工成一定尺寸的板，上面至少有一个导电图形及所设计好的孔，以实现电子元器件之间的电气连接，这样的板称为印刷电路板，图 1.3[3]给出了 PCB 的实物图。1936 年，英国 Eisler 博士提出印制电路（Printed Circuit）这个概念，他首创在绝缘基板上全面覆盖金属箔，在其金属箔上涂上耐蚀刻油墨后，再将不需要的金属箔腐蚀掉的 PCB 制造基本技术，1942 年，Eisler 博士制造出世界第一块纸质层压绝缘基板，用于收音机的印制板。50 年代初，这种技术开始广泛应用，并迅速得到发展。印制电路是电子产品的关键电子互连件，无论在市场领域，应用数量和技术水平等各方面都占有极重要的地位并在快速发展。一方面印刷电路板是各种消费类电子产品和投资类电子产品的基本零组件，目前，消费类电子产品和投资类电子产品要求其所对应的电子机器具有“轻，薄，短，小”化，多样化，多功能，高速，高可靠，研制周期短和价廉等特点，这就促使所设计出的印刷电路板朝着小孔径，细线条，轻薄型的方向发展。另一方面在电子工业领域，随着现代科学技术日新月异地发展，大规模、超大规模集成电路的使用使印刷电路板的走线愈加精密和复杂。在这种情况下，传统的手工设计方式已越来越难以适应当前的形势，因此计算机来辅助设计电路板已成为电路板设计制作的必然趋势。

PCB 的制造过程[4, 5]简单来说，首先把集成电路用陶瓷或塑料管壳封装，然后再安装在印刷电路板上，目前的集成电路封装方法主要有三种：双列直插式 DIP、引腿阵列式 PGA 和表面封装（Suface Mounted Device ，SMD），封装好的器件的引腿插入、安放在印制电路板表面，这个过程叫布局，再根据布局测试布线的可能性，然后在各个元器件之间进行连线，实现元件之间的电气连接关系，

这个过程叫布线。最后再制作导线，将封装件插入并焊接在镀锡的孔中，或焊接在印制电路板表面。图 1.4[3]给出了 PCB 的布线样式。

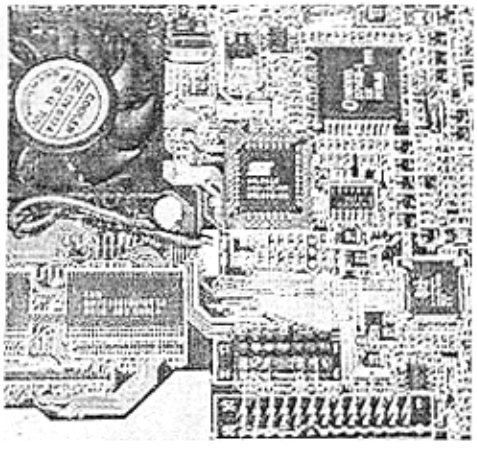


图 1.3 标准 PCB 的实物图

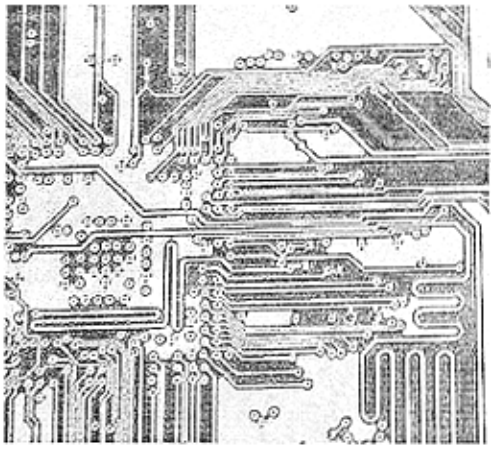


图 1.4 标准 PCB 的导线图

自动布线是 PCB 软件设计的最主要的功能，印刷电路板自动布线就是让程序根据用户设定的有关参数和布线规则，依照一定的程序算法，自动在各个元器件之间进行连线，实现元件之间的电气连接关系，从而完成印刷电路板的布线工作。自动布线算法是衡量 PCB 设计软件好坏的标准，布线算法选择对印刷电路板布线的布通率有很大的影响。

随着计算机工业的飞速发展和大规模，超大规模集成电路的出现，芯片的集成度越来越高，印刷电路越来越复杂，布线的难度也越来越大，已非人工布线所能及，自动布线的研究工作也应运而生。自动布线是电子 CAD 领域中的一个经典课题，在近四十年的研究过程中，其理论、技术和方法均取得很大的进步。然而，自动布线问题的本身是复杂的，尽管人们为提高自动布线的布通率和速度，提出了各种很好的方法，但由于高密度引脚及引脚尺寸日趋物理极限，导致低的布通率，或虽然布通但不能满足设计要求，自动布线仍然需要大量的人工干预时间。就目前的情况来看，在印制电路板自动布线方面，有李氏算法，线探索法，最优通道法，Hitchcook 的细胞结构法，Hadhimoto 及 Stevens 的通道分配法，Mah 及 Stainberg 的拓扑类并法，J.Soukup 的快速迷路法等各种方法。近年来，又提出了朝向目标的线探索法。虽然方法很多，但都有各自的缺点和局限性。而且目前的几大 PCB 布线工具虽然它们的布通率都还差强人意，但在系统自动布完线完成后还得靠手工修改后才能达到预期效果，而且电路愈复杂，所要的手工

修改量就愈大。因此采用什么样的方法能使布线在符合设计要求的前提下，布通率更高，速度更快，让系统自动布线后所做的手工修改最少，仍然是一个没有被完满解决的课题。

1.2 研究内容

本论文研究内容是针对实用要求，研究印刷电路板的自动布线的算法理论和实现以及算法的优化。它是西北工业大学机电学院廖达雄副教授正在开发的电路 CAD 软件的一部分，同时也是该软件的几个核心技术之一。

本文首先介绍了电路 CAD 的概念和电路 CAD 软件的结构以及 PCB 布线软件在电路 CAD 软件中的地位和作用；第二章介绍已有的布线算法，并分析各算法的优点和缺点及适用情况；第三章，简要介绍本论文所涉及到的基础知识，为后续章节的展开做必要的准备，其中的知识涉及到图论、动态规划、斯坦纳树的构造等；第四章，在分析已有算法的基础上，结合图论知识和人工智能解决问题的思路提出了更符合实际要求的速度快，具有高布通率的布线算法——智能识点法的布线理论。这部分是本文的核心也是本文的创新之处，在这部分中，首先给出线网排序的算法；其次，结合人工智能解决问题的思路，提出 MD 布线模型下的广义线段的概念及 MD 线长的计算方法；再次，提出绕障点的概念，给出详细定义绕障点的方法；最后，给出以广义线段为边，以绕障点和待布点为顶点的带边权值的完全图构造方法及在图中搜索最短路径的算法。第五章，首先，给出智能识点法的实现的数据结构；其次，指出了算法实现过程中的几个关键问题，并给出了解决的详细方案；最后，给出算例，并根据实验数据和对数据的分析，总结出算法的特点，指出智能识点法布线的有待改进之处和今后的努力方向。第六章结束语。

修改量就愈大。因此采用什么样的方法能使布线在符合设计要求的前提下，布通率更高，速度更快，让系统自动布线后所做的手工修改最少，仍然是一个没有被完满解决的课题。

1.2 研究内容

本论文研究内容是针对实用要求，研究印刷电路板的自动布线的算法理论和实现以及算法的优化。它是西北工业大学机电学院廖达雄副教授正在开发的电路 CAD 软件的一部分，同时也是该软件的几个核心技术之一。

本文首先介绍了电路 CAD 的概念和电路 CAD 软件的结构以及 PCB 布线软件在电路 CAD 软件中的地位和作用；第二章介绍已有的布线算法，并分析各算法的优点和缺点及适用情况；第三章，简要介绍本论文所涉及到的基础知识，为后续章节的展开做必要的准备，其中的知识涉及到图论、动态规划、斯坦纳树的构造等；第四章，在分析已有算法的基础上，结合图论知识和人工智能解决问题的思路提出了更符合实际要求的速度快，具有高布通率的布线算法——智能识点法的布线理论。这部分是本文的核心也是本文的创新之处，在这部分中，首先给出线网排序的算法；其次，结合人工智能解决问题的思路，提出 MD 布线模型下的广义线段的概念及 MD 线长的计算方法；再次，提出绕障点的概念，给出详细定义绕障点的方法；最后，给出以广义线段为边，以绕障点和待布点为顶点的带边权值的完全图构造方法及在图中搜索最短路径的算法。第五章，首先，给出智能识点法的实现的数据结构；其次，指出了算法实现过程中的几个关键问题，并给出了解决的详细方案；最后，给出算例，并根据实验数据和对数据的分析，总结出算法的特点，指出智能识点法布线的有待改进之处和今后的努力方向。第六章结束语。

第二章 PCB 自动布线理论研究概况

2.1 印刷电路板自动布线在电路 CAD 软件中的作用

在电子电路 CAD 软件设计系统中，印刷电路板的设计系统结构如图 2.1 所示。

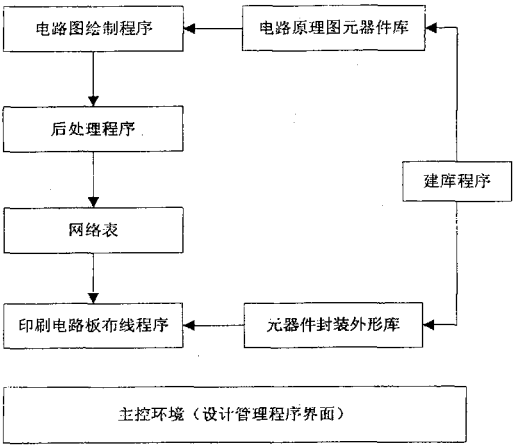


图 2.1 印刷电路板设计系统结构图

其中，电路原理图绘制程序和印刷电路板布线程序可以分别作为两个独立的软件来完成各自对应的功能。而网络表则是联系电路原理图和印刷电路板布线的纽带。印刷电路板的设计工序流程图如图 2.2 所示。对比两图就可以看出其中的对应关系：电路图绘制程序是 PCB 设计流程中的前段处理程序，主要是负责绘制电路原理图、各元件属性与仿真参数的设置。电路原理图经过后处理程序的处理产生网络表，也可以根据需要产生其它文本文件如：元件列表等，接下来就是根据网络表所提供的数据信息和 PCB 布线程序进行印刷电路板的设计，其中包括：电路板的规划、设置参数、装入网络表和元件封装外形、元件布局、自动布线以及存盘打印等步骤。在整个印刷电路板的设计过程中自动布线是核心技术，也就是本论文的主要研究内容。

第二章 PCB 自动布线理论研究概况

2.1 印刷电路板自动布线在电路 CAD 软件中的作用

在电子电路 CAD 软件设计系统中，印刷电路板的设计系统结构如图 2.1 所示。

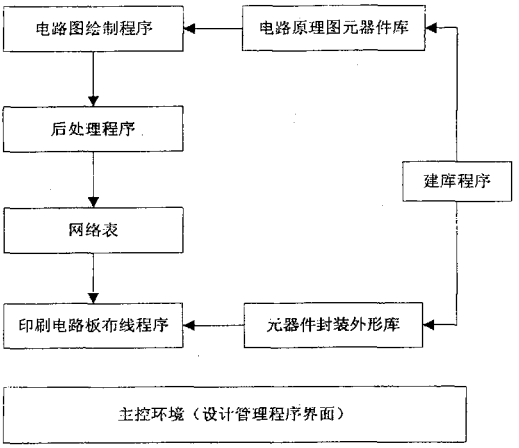


图 2.1 印刷电路板设计系统结构图

其中，电路原理图绘制程序和印刷电路板布线程序可以分别作为两个独立的软件来完成各自对应的功能。而网络表则是联系电路原理图和印刷电路板布线的纽带。印刷电路板的设计工序流程图如图 2.2 所示。对比两图就可以看出其中的对应关系：电路图绘制程序是 PCB 设计流程中的前段处理程序，主要是负责绘制电路原理图、各元件属性与仿真参数的设置。电路原理图经过后处理程序的处理产生网络表，也可以根据需要产生其它文本文件如：元件列表等，接下来就是根据网络表所提供的数据信息和 PCB 布线程序进行印刷电路板的设计，其中包括：电路板的规划、设置参数、装入网络表和元件封装外形、元件布局、自动布线以及存盘打印等步骤。在整个印刷电路板的设计过程中自动布线是核心技术，也就是本论文的主要研究内容。

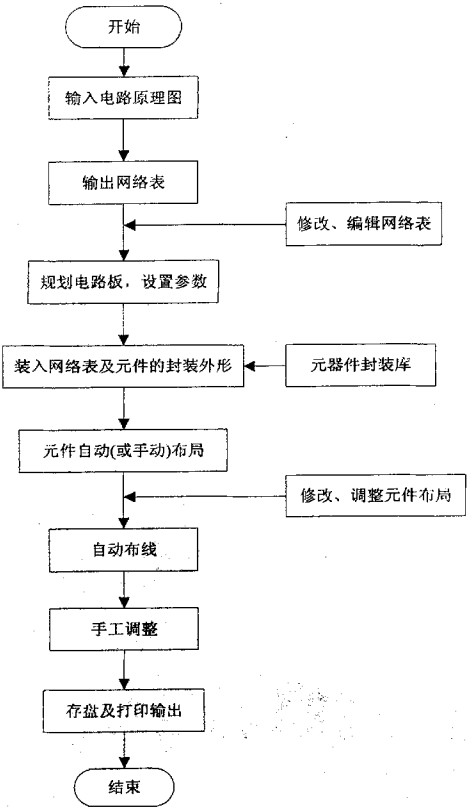


图 2.2 印刷电路板设计工序流程图

网络表所提供的信息主要包括两个方面：一方面是电路原理图中所有元器件的属性信息，另一方面是元器件之间的电气连接关系。这些数据恰恰是 PCB 布线所需要的关键条件。由此也可以看出，网络表是电路原理图和印刷电路板之间的一座桥梁，是实现 PCB 自动布线的基础。

2.2 印刷电路板自动布线的基本概念和分类

2.2.1 印刷电路板的基本概念

印刷电路板是构建电路系统的基础技术，将设计电路中各元件间的电气连接线作成实体铜膜连接线，在一层或数层绝缘板子上作出信号板层，并适当地蚀刻成元件外形的焊点和铜膜走线来安装与连接各个电子元件。早期的绝缘板都是使用电木为材料，现在则大多都改用玻璃纤维材料，厚度更薄，而弹性和韧度都更好。

所谓元件外形主要就是一群根据实际元件包装尺寸而定义好的焊点，另外还附加一些属性和展示元件外观的符号。元件外形符号与属性符号主要是供人看的，不具备特殊的电气或实体铜膜意义。

所谓焊点（或焊盘）就是提供外界用焊接方式来连接元件引脚与电路板走线的铜膜接点。在 PCB 制作流程中蚀刻出各焊点间的连接铜线就是铜膜走线。在铺布铜膜走线时，如果有别的走线或是元件挡住了去向，就得在绝缘板上钻孔形成所谓的导孔（Via，又称贯孔），然后通过导孔的连接使铜膜走线可以切换到另外一个布线板层继续完成连接焊点的工作，如图 2.3 所示。

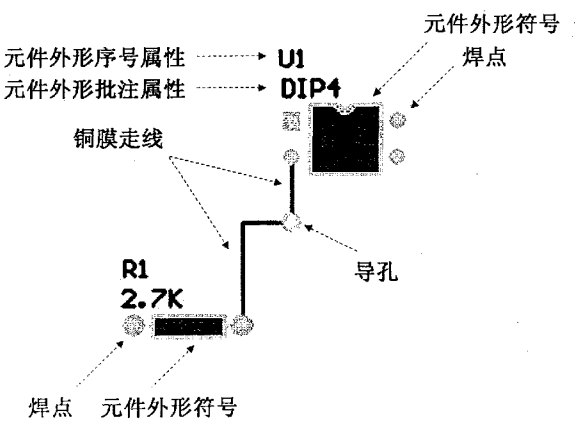


图 2.3 元件外形、焊点、铜膜走线与导孔

2.2 印刷电路板自动布线的基本概念和分类

2.2.1 印刷电路板的基本概念

印刷电路板是构建电路系统的基础技术，将设计电路中各元件间的电气连接线作成实体铜膜连接线，在一层或数层绝缘板子上作出信号板层，并适当地蚀刻成元件外形的焊点和铜膜走线来安装与连接各个电子元件。早期的绝缘板都是使用电木为材料，现在则大多都改用玻璃纤维材料，厚度更薄，而弹性和韧度都更好。

所谓元件外形主要就是一群根据实际元件包装尺寸而定义好的焊点，另外还附加一些属性和展示元件外观的符号。元件外形符号与属性符号主要是供人看的，不具备特殊的电气或实体铜膜意义。

所谓焊点（或焊盘）就是提供外界用焊接方式来连接元件引脚与电路板走线的铜膜接点。在 PCB 制作流程中蚀刻出各焊点间的连接铜线就是铜膜走线。在铺布铜膜走线时，如果有别的走线或是元件挡住了去向，就得在绝缘板上钻孔形成所谓的导孔（Via，又称贯孔），然后通过导孔的连接使铜膜走线可以切换到另外一个布线板层继续完成连接焊点的工作，如图 2.3 所示。

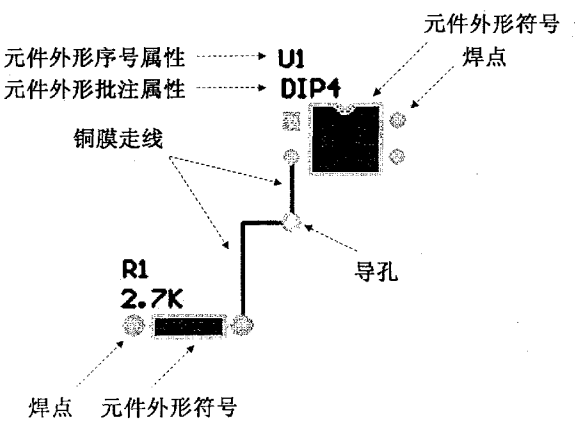


图 2.3 元件外形、焊点、铜膜走线与导孔

由于印刷电路板所使用的实体元件可分为针脚式和表面粘着式（SMD：Surface Mount Device）两种，所以焊点的形式也可以区分为针脚式焊点和 SMD 式焊点。

2.2.2 布线设计的目标

在布图设计中，布线设计的目标可描述为：根据电路的连接关系描述（网络表），在满足设计、工艺规则的要求和满足电学性能的要求的条件下，在限定区域（面积、形状、层次等）内 100%地完成所需的互连。或者是在 100%完成所需互连时，使所需的 PCB（或芯片）面积最小化。同时要求尽可能优化其设计结果（如连线长度最小化，通孔数最小化等）。

2.2.3 布线设计所面临的问题及布线方法的分类

布线设计所面临的问题：

在平面上实现一条线网的布线并不困难，但当需布线网数足够多时，如需求其连线总长最短的布线方案则是一个复杂的难题。而在实际问题中，线网数一般相当多，有些线网还是多端点线网（即一条线网联结着多于两个接点的情况），情况将更加复杂。除了必须考虑线网需满足的电学、工艺要求外，当第 i 条线网布线时，为了不与同平面的已布线交叉，不仅需要考虑线网自身各接点间的关系，而且还必须考虑已布线的影响及对今后其它线网布线的影响。此时，是否存在一条实现互连的路径，如何找到互连的路径，如何找到一条最短的路径，或找到一条对今后布线最有利的路径，都成为相当困难的问题。同时，对于每个线网，又有几百种甚至更多种布线方案，这样就使得布线问题异常复杂。

布线方法的分类：

通常有两种策略实现布线，即直接的区域布线和分两步实现的总体布线和详细布线。就布线的对象来分，布线问题也可以分为面向线网的布线和面向区域的布线。面向线网的布线主要以线网作为考虑对象，如总体布线和区域布线都属于面向线网的布线；而面向区域的布线主要以布线区域作为考虑对象，如两边通道布线和开关盒布线都属于面向区域的布线问题。

根据本课题的实用背景，本文主要讨论面向线网的布线算法。

由于印刷电路板所使用的实体元件可分为针脚式和表面粘着式（SMD：Surface Mount Device）两种，所以焊点的形式也可以区分为针脚式焊点和 SMD 式焊点。

2.2.2 布线设计的目标

在布图设计中，布线设计的目标可描述为：根据电路的连接关系描述（网络表），在满足设计、工艺规则的要求和满足电学性能的要求的条件下，在限定区域（面积、形状、层次等）内 100%地完成所需的互连。或者是在 100%完成所需互连时，使所需的 PCB（或芯片）面积最小化。同时要求尽可能优化其设计结果（如连线长度最小化，通孔数最小化等）。

2.2.3 布线设计所面临的问题及布线方法的分类

布线设计所面临的问题：

在平面上实现一条线网的布线并不困难，但当需布线网数足够多时，如需求其连线总长最短的布线方案则是一个复杂的难题。而在实际问题中，线网数一般相当多，有些线网还是多端点线网（即一条线网联结着多于两个接点的情况），情况将更加复杂。除了必须考虑线网需满足的电学、工艺要求外，当第 i 条线网布线时，为了不与同平面的已布线交叉，不仅需要考虑线网自身各接点间的关系，而且还必须考虑已布线的影响及对今后其它线网布线的影响。此时，是否存在一条实现互连的路径，如何找到互连的路径，如何找到一条最短的路径，或找到一条对今后布线最有利的路径，都成为相当困难的问题。同时，对于每个线网，又有几百种甚至更多种布线方案，这样就使得布线问题异常复杂。

布线方法的分类：

通常有两种策略实现布线，即直接的区域布线和分两步实现的总体布线和详细布线。就布线的对象来分，布线问题也可以分为面向线网的布线和面向区域的布线。面向线网的布线主要以线网作为考虑对象，如总体布线和区域布线都属于面向线网的布线；而面向区域的布线主要以布线区域作为考虑对象，如两边通道布线和开关盒布线都属于面向区域的布线问题。

根据本课题的实用背景，本文主要讨论面向线网的布线算法。

2.3 面向线网自动布线算法的概况及分析

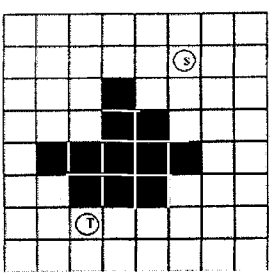
面向线网的自动布线算法主要有迷宫算法和线探索法两类。

2.3.1 迷宫算法自动布线

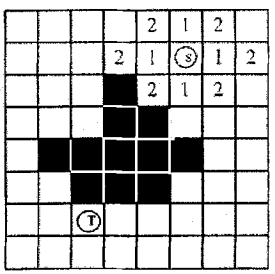
印刷电路板是焊盘、过孔、走线和铜区等物体的组合，每种物体可有任意形状。单纯从电器特性、经济因素和制造关系考虑，好的 PCB 设计应使走线区的总面积和过孔数目尽可能减少，从而可保证最少的面板完成设计，使产量得以提高。

世界上最早的布线器当推 1961 年发布的 Lee 算法[6]，它实际上是图论中最小路径算法在矩形网格上的一种应用。其算法的思想也可以描述为对波传递过程的模拟。在一个存在障碍的湖面上，若需寻找连接点 A 和 B 之间的最短路径，可以在点 A 处投下一个小石子，然后观察所引起的水波的传播情况。假定水波的传播过程中能量没有损失，则当遇到障碍时，波发生绕射，最先到达目标点的波前所经过的路径必是一条最短路径，而且只要二点间存在通路，则从点 A 开始扩展传播的水波一定将波传播到点 B，也即只要通路存在就一定能找到这条通路。这个过程可以形象地在计算机中进行模拟。

李氏算法地布线过程大致可分为（1）数据准备（2）扩展过程（3）回找过程，图 2.4 所示就是李氏算法布线的原理。



（a）



（b）

2.3 面向线网自动布线算法的概况及分析

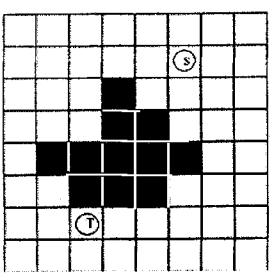
面向线网的自动布线算法主要有迷宫算法和线探索法两类。

2.3.1 迷宫算法自动布线

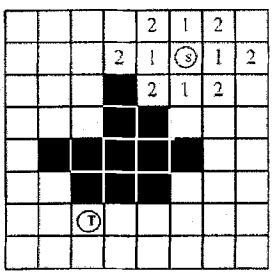
印刷电路板是焊盘、过孔、走线和铜区等物体的组合，每种物体可有任意形状。单纯从电器特性、经济因素和制造关系考虑，好的 PCB 设计应使走线区的总面积和过孔数目尽可能减少，从而可保证最少的面板完成设计，使产量得以提高。

世界上最早的布线器当推 1961 年发布的 Lee 算法[6]，它实际上是图论中最小路径算法在矩形网格上的一种应用。其算法的思想也可以描述为对波传递过程的模拟。在一个存在障碍的湖面上，若需寻找连接点 A 和 B 之间的最短路径，可以在点 A 处投下一个小石子，然后观察所引起的水波的传播情况。假定水波的传播过程中能量没有损失，则当遇到障碍时，波发生绕射，最先到达目标点的波前所经过的路径必是一条最短路径，而且只要二点间存在通路，则从点 A 开始扩展传播的水波一定将波传播到点 B，也即只要通路存在就一定能找到这条通路。这个过程可以形象地在计算机中进行模拟。

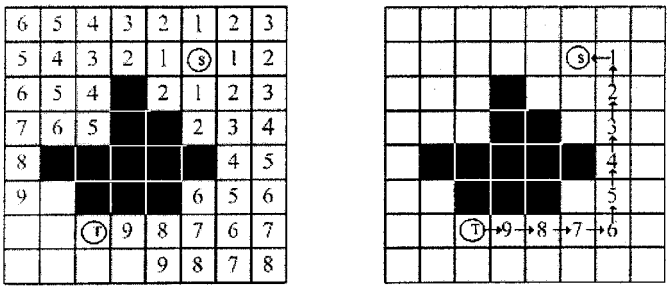
李氏算法地布线过程大致可分为（1）数据准备（2）扩展过程（3）回找过程，图 2.4 所示就是李氏算法布线的原理。



（a）



（b）



（c）　　（d）

图 2.4 李氏布线算法的原理图

图中黑格为配线禁区，空格为可配线区，S、T 为待连接的两个端点。S－T 两点间的连线过程是：

（1）确定起始单元为 S（T 为终止单元），图（a）所示；

（2）将 S 点的相邻单元设为“1”单元，再设“1”单元的相邻单元为“2”，依此类推，直至某单元（图中为“9”单元）的相邻单元为配线终止单元为止，图（b）、（c）所示；

（3）由终止单元开始进行逆探索，确定配线路径（T－9－8－…－2－1－S）。图 2.4（d）所示为 S－T 点的最终配线结果，并且为最短的配线路径。

李氏算法特点的讨论：

1、极强的绕障能力。李氏算法在解决两点的互连问题时，只要接点间存在着通路，则无论其间障碍何等复杂也一定能找到其中最短的路径，换言之，李氏算法具有极强的绕障能力且保证当前连线的最短化。

2、由于布线区域的网格化，使算法可较容易地描述布线区域各处的特性，并容易模拟各种具体的工艺要求。因此，李氏算法的另一重要特点是具有很好的适应性,它可以适用不同电路的电性能要求采用各种不同的布线模型。图 2.5 和图 2.6 分别表示了 4－λ 几何结构和 3－λ 几何结构的李氏算法。

3、可用于多层布线[7]。在实际问题中，往往允许布线可在几层平面上进行，不同层上的连线可通过通孔（过孔）实现必要的互连。以两层布线为例，人工布线的实践表明：为获得较高的布线完成率，一个有效的方法是把

连线的水平线放在一层，而把垂直线放在另一层。考虑到尽可能减少通孔，上述原则可修改为把大部分垂直线或近似为垂直线的连线放在另一层。而李氏算法应用于双层布线时，可很方便地模拟人们这种布线原则。从布线层的特性来看，上述原则可等价地描述为：在一层上，水平方向的布线价格较小而垂直方向的布线价格较大，因而在该层上宜于布水平线，另一层上恰相反。因此可规定在一层上水平方向扩展的网格权重为1，而在垂直方向的扩展网格权重为2，而且另一层水平扩展网格权重为2，垂直扩展网格权重为1，设计者为尽量减少通孔数而规定在上层上连接垂直线段长度小于和等于L时（在下层连接水平线段长度小于、等于L时）不穿孔，这样就可以用李氏算法实现双层布线。

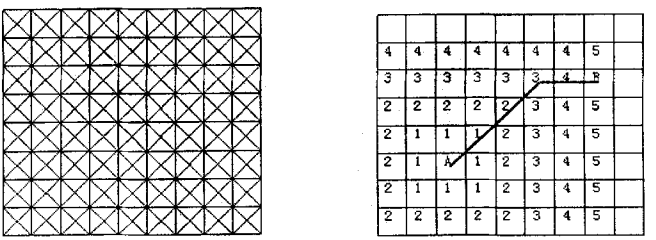


图 2.5　4－λ几何结构的李氏布线算法

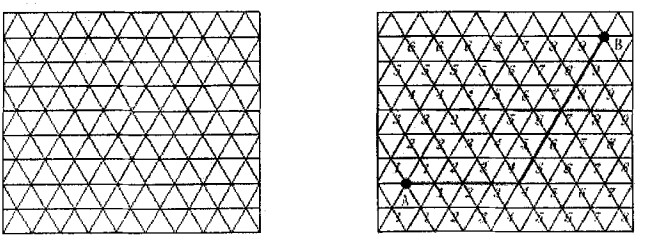


图 2.6　3－λ几何结构的李氏布线算法

由于李氏算法存在多种优点，因此在 PCB 和 IC（集成电路）的布线设计中，该算法得到了广泛应用。

由于这种布线算法要求存储所有网点的信息，因此所需存储量相当可观。网

格的缩小导致了 PCB 上网格点成平方倍地增长，因而存储量和计算时间增长很快，对硬件的能力要求较高。

自李氏算法提出后，有许多对该算法的改进，包括提高其速度和减小其计算空间。李氏算法与其改进算法形成了一组布线算法，统称为迷宫算法（maze routing algorithms）。如 Akers[8] 提出了一种减少存储空间的方法；J.Soukup[9] 提出了一个带有固定意义的非对称搜索方法；Hadlock[10] 提出最小迂回算法；F.Rubin[11] 提出定向搜索算法；Korn[12] 提出了一种新的波前元素扩展优先度的描述方法等。尽管如此，需要存储量大仍是该类算法的一个令人遗憾的问题。

2.3.2 线探索法自动布线

迷宫算法的最大缺点是它要搜索较大的布图空间，所以要化费较大的时间和存储空间。人们开始考虑能否在搜索中避免大量网格空间的搜索而代之以较简单的搜索方法。当然，人们想到的简单图形搜索方法就是直线探索的方法。1969 年 D.W.Hightower 提出了一种基于线扩展的布线方法，也就是线探索法（linear expansion）[13]。这种算法的基本思想就是用线探索的方法减少存储空间和计算时间，因为探索用到的空间要比探索用的网格所存储的空间少得多，它的计算时间与空间复杂度均为 O（L），L 为算法所产生的线数。

线探索法的基本算法描述如下：

从源点及目标点出发，交替地作逃避线，用两个表 slist 和 tlist 存放两点各自产生的线（逃避线），该线不能穿越障碍物。如果 slist 中的直线与 tlist 中的相交，那么探索就结束；否则，继续产生新的直线。这些直线要起源于 slist 与 tlist 线上的“逸出点”（escape points），新的 slist(tlist)线要与原 slist(tlist)线相交。若探索结束，从目标点回溯找到源点，从而形成一通路。线的产生是在水平和垂直方向交替进行的。在 Hightower 算法中，每根线只有一个逸出点。最简单的情况是搜索的线与障碍物平行，那么逸出点就在障碍物尽头那边线的尾端。图 2.7 所示就是 Hightower 算法的原理图。

为了提高搜索效率，可对逸出点的搜索顺序给定一些简单的原则，如离源点近的逸出点先搜索，离目标近的逸出点先搜索，朝向目标的逸出点先搜索等。

格的缩小导致了 PCB 上网格点成平方倍地增长，因而存储量和计算时间增长很快，对硬件的能力要求较高。

自李氏算法提出后，有许多对该算法的改进，包括提高其速度和减小其计算空间。李氏算法与其改进算法形成了一组布线算法，统称为迷宫算法（maze routing algorithms）。如 Akers[8] 提出了一种减少存储空间的方法；J.Soukup[9] 提出了一个带有固定意义的非对称搜索方法；Hadlock[10] 提出最小迂回算法；F.Rubin[11] 提出定向搜索算法；Korn[12] 提出了一种新的波前元素扩展优先度的描述方法等。尽管如此，需要存储量大仍是该类算法的一个令人遗憾的问题。

2.3.2 线探索法自动布线

迷宫算法的最大缺点是它要搜索较大的布图空间，所以要化费较大的时间和存储空间。人们开始考虑能否在搜索中避免大量网格空间的搜索而代之以较简单的搜索方法。当然，人们想到的简单图形搜索方法就是直线探索的方法。1969 年 D.W.Hightower 提出了一种基于线扩展的布线方法，也就是线探索法（linear expansion）[13]。这种算法的基本思想就是用线探索的方法减少存储空间和计算时间，因为探索用到的空间要比探索用的网格所存储的空间少得多，它的计算时间与空间复杂度均为 O（L），L 为算法所产生的线数。

线探索法的基本算法描述如下：

从源点及目标点出发，交替地作逃避线，用两个表 slist 和 tlist 存放两点各自产生的线（逃避线），该线不能穿越障碍物。如果 slist 中的直线与 tlist 中的相交，那么探索就结束；否则，继续产生新的直线。这些直线要起源于 slist 与 tlist 线上的“逸出点”（escape points），新的 slist(tlist)线要与原 slist(tlist)线相交。若探索结束，从目标点回溯找到源点，从而形成一通路。线的产生是在水平和垂直方向交替进行的。在 Hightower 算法中，每根线只有一个逸出点。最简单的情况是搜索的线与障碍物平行，那么逸出点就在障碍物尽头那边线的尾端。图 2.7 所示就是 Hightower 算法的原理图。

为了提高搜索效率，可对逸出点的搜索顺序给定一些简单的原则，如离源点近的逸出点先搜索，离目标近的逸出点先搜索，朝向目标的逸出点先搜索等。

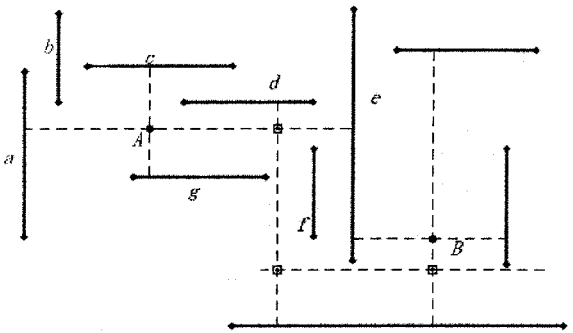


图 2.7　Hightower 布线算法

线搜索法的特点：

由于采用线扩展，使算法可望具有较高的布线效率，尤其在开始布线时，由于已布线数量较少，障碍较少，该算法有极高的布线效率。这一点和李氏算法的情况恰好相反，李氏算法一般来讲，随着已布线数量的增加，由于所需扩展的网格数减少，而使布线效率得到一定的提高。

由于线探索法采用线扩展的方法，只需存贮插孔及布出线段的端点坐标，因而存储量也不象李氏算法那样成平方关系地增加，从而使运算速度大大加快。使其较为适合用于规模大的布线问题。

线探索法的问题是：当逃避线进入一个入口较窄的区域时，无法出来，从而出现有通路而探索不出的情况，其绕障能力低于李氏算法。所求得的路径并不总是最短路径。在实际应用中，其布线结果常常是非常接近李氏算法的布线结果，该算法的布线效率随着已布线数量的增加而下降。此外，还存在一些其它的不足之处，如：做逃避线时，没有利用对方端点的坐标信息，探索带有盲目性，探索范围较大，可能多花时间；从两端交替作逃避线，处理起来较困难。

Mikami 和 Tabuchi 在 1986 年[14]提出了改进的线探索方法，在 Mikami 算法中，线上的每一个点都是逸出点，可以产生与之垂直的新线。这相当于一个广度优先搜索，它保证能找到一条存在的通路，但该通路不一定是最短的。J.Soukup 提出的快速迷路法[15]是线探索法与李氏算法的结合，就是先朝目标进行线探索，遇到障碍后，再由迷路法来绕过障碍。他的方法比李氏算法快，内存也比较节省，但本质上仍是逐点查找，同样需要存贮所有网点信息。Heyns 在文献【16】中也

提出了类似的算法。非李氏型算法都是启发性的，都没有李氏算法那样的普遍性，不能保证路径最短，也不能保证两点之间有通路一定能布出。这方面国内外学者做了大量的工作，如： Hashimoto及Stevens的通道分配法[17]，杨瑞元[18,19,20]提出朝向目标的线探索法等。

文献【21】描述了另一种方法，它利用波传播中的绕射规律，在简单的线探索的基础上，把波扩展到整个区域，当遇到布线障碍时，波发生绕射，并记录这些绕射边界，然后绕过障碍继续扩展直至找到存在的互连路径为止。这种方法布线时，可使布线路径“逼近障碍”，所需存储量较少而布线效率较高。

2.4 几种其它自动布线算法简介

1、整体布线

J.Soukup在文献【22】中提出了一种“同时”布线的想法。其设想为每条连线在最后确定前并不以线条的方式出现，而象一条变形虫一样可以伸展、收缩，似乎是“活的”，而相互竞争实现接点互连所需的空间。整体布线的扩展过程非常象李氏算法，不同点是在这种布线思想中，不存在已布线所形成的那种“僵硬的”障碍，它们是连续地变化而且不断地相互作用，直到最后才以线条的形式实现所有线网实际布线。

2、线性规划法

Vanneli[23]设计出一种纯并行算法——线性规划法。这种算法找到各个线网所有可能的连接树及其代价（布线长度或拥挤度的代价），再根据总体布线单元边与单元的容量限制列出线性方程组，最终找到最优解。这种方法的最大难处在于其极高的布线复杂度。要使它投入实用，还需要有可行的简化模型。

3、层次布线算法

层次布线算法[24,25]是一种总体布线算法，其基本思想是任务分治，即把大的任务逐步细分成多个小任务，而后一一解决。这与总体布线和详细布线分开的方案有异曲同工之妙。层次算法是一种并行性较强的高效率算法，但它的不足之处在于，在做高层块级总体布线时，由于对底层布线时的局部拥挤状况不了解，会导致一些上层决策不适合实际布线的问题。

提出了类似的算法。非李氏型算法都是启发性的，都没有李氏算法那样的普遍性，不能保证路径最短，也不能保证两点之间有通路一定能布出。这方面国内外学者做了大量的工作，如： Hashimoto及Stevens的通道分配法[17]，杨瑞元[18,19,20]提出朝向目标的线探索法等。

文献【21】描述了另一种方法，它利用波传播中的绕射规律，在简单的线探索的基础上，把波扩展到整个区域，当遇到布线障碍时，波发生绕射，并记录这些绕射边界，然后绕过障碍继续扩展直至找到存在的互连路径为止。这种方法布线时，可使布线路径“逼近障碍”，所需存储量较少而布线效率较高。

2.4 几种其它自动布线算法简介

1、整体布线

J.Soukup在文献【22】中提出了一种“同时”布线的想法。其设想为每条连线在最后确定前并不以线条的方式出现，而象一条变形虫一样可以伸展、收缩，似乎是“活的”，而相互竞争实现接点互连所需的空间。整体布线的扩展过程非常象李氏算法，不同点是在这种布线思想中，不存在已布线所形成的那种“僵硬的”障碍，它们是连续地变化而且不断地相互作用，直到最后才以线条的形式实现所有线网实际布线。

2、线性规划法

Vanneli[23]设计出一种纯并行算法——线性规划法。这种算法找到各个线网所有可能的连接树及其代价（布线长度或拥挤度的代价），再根据总体布线单元边与单元的容量限制列出线性方程组，最终找到最优解。这种方法的最大难处在于其极高的布线复杂度。要使它投入实用，还需要有可行的简化模型。

3、层次布线算法

层次布线算法[24,25]是一种总体布线算法，其基本思想是任务分治，即把大的任务逐步细分成多个小任务，而后一一解决。这与总体布线和详细布线分开的方案有异曲同工之妙。层次算法是一种并行性较强的高效率算法，但它的不足之处在于，在做高层块级总体布线时，由于对底层布线时的局部拥挤状况不了解，会导致一些上层决策不适合实际布线的问题。

2.5 智能识点法自动布线算法

根据上面的分析和对迷宫算法和线探索布线算法优缺点的比较，迷宫算法因为运用了网格而使算法的具有极强的绕障能力，但由于这种布线算法要求存储所有网点的信息，所需存储量很大。网格的缩小导致了 PCB 上网格点成平方倍地增长，因而存储量和计算时间增长很快，对硬件的能力要求较高。而线探索法采用线扩展的方法，只需存贮插孔及布出线段的端点坐标，从而使运算速度大大加快。但由于未能搜索足够的空间，而常使本应存在的路径寻找不到，并且所得路径常常并非最佳路径。如何能找到一种即具有较强的绕障能力，又用较少搜索时间的自动布线算法是多年来该方面的专家学者一直追求的目标。

为了绕过障碍，迷宫算法采用了广度优先搜索的思想，逐网格点搜索绕过障碍的路径。事实上，在搜索过程中，许多网格布线的路径并未用到，但却耗费了许多宝贵的运行时间。本文模仿人工智能绕过障碍物的思维，在布线中出现的障碍物周围定义一些能绕过它的点，称为绕障点，以绕障点和待布点组成布线点集合，将布线问题转化为在点集合中求布线点间的最短路径的问题，从而得到最优布线路径。我们称这种方法为智能识点法。该算法优点：1、极强的绕障能力；2、布线速度与布线绕障点的数量有关，与布线间距无关。当线网数量不多，但布线的间距极小时，与 Lee 氏算法相比，速度的优越性很明显；3、算法在确定绕障点时借用了网格，但其空间复杂性与网格间距无关，因此可用于高密度、高精度布线，这点优于迷宫算法。

第三章　智能识点法布线的基本知识

前面章节介绍了各种布线算法及其优缺点，这一章我们将介绍本论文采用的算法——智能识点法所用到的基本知识。

3.1 智能识点法自动布线思路

李氏自动布线算法是通过搜索可布线区域内由近及远的所有可能的网格点，直到获得最佳布线路径，但事实上搜索的网格点中大多数是对布线无用的点，从而耗费了大量的运行时间和存储空间。本文算法思路是：从大量的网格点中选出对布线有用的网格点——绕障点，在以绕障点构成完全图，应用相关图论的知识和求解最短路径的算法求得最佳布线路径。

3.2 智能识点法布线所用到的相关知识

3．2．1　图的基本术语与相关知识

本文算法是以图论中的完全图为基础完成布线的路径搜索的，所以，为了方便叙述，这一节将简要介绍一下相关的图论的基本术语与知识[26,27]。

图论是以图为研究对象的数学分支。图论中的图指的是一些点以及连接这些点的线的总体。通常用点代表事物，用连接两点的线代表事物间的关系、图论则是研究事物对象在上述表示法中具有的特征与性质的学科。

图：图 G（如图 3.1）是一个二元组 G(V,E)，其中 V 是节点集合，E 是边的集合。如果（u,v）是一个边，即（u,v）∈E，则节点 u 是 v 的一个邻接节点。所有 v 的邻接节点记为 Adj(v)，一条边 e=(u,v)连接 u 和 v，则 u 和 v 是 e 的端点。节点 u 的度是与它连接的边的数目。

完全图：（如图 3.2）一个有 n 个节点的完全图是这样一个图，它的每一个节点都与其它所有节点邻接。

第三章　智能识点法布线的基本知识

前面章节介绍了各种布线算法及其优缺点，这一章我们将介绍本论文采用的算法——智能识点法所用到的基本知识。

3.1 智能识点法自动布线思路

李氏自动布线算法是通过搜索可布线区域内由近及远的所有可能的网格点，直到获得最佳布线路径，但事实上搜索的网格点中大多数是对布线无用的点，从而耗费了大量的运行时间和存储空间。本文算法思路是：从大量的网格点中选出对布线有用的网格点——绕障点，在以绕障点构成完全图，应用相关图论的知识和求解最短路径的算法求得最佳布线路径。

3.2 智能识点法布线所用到的相关知识

3．2．1　图的基本术语与相关知识

本文算法是以图论中的完全图为基础完成布线的路径搜索的，所以，为了方便叙述，这一节将简要介绍一下相关的图论的基本术语与知识[26,27]。

图论是以图为研究对象的数学分支。图论中的图指的是一些点以及连接这些点的线的总体。通常用点代表事物，用连接两点的线代表事物间的关系、图论则是研究事物对象在上述表示法中具有的特征与性质的学科。

图：图 G（如图 3.1）是一个二元组 G(V,E)，其中 V 是节点集合，E 是边的集合。如果（u,v）是一个边，即（u,v）∈E，则节点 u 是 v 的一个邻接节点。所有 v 的邻接节点记为 Adj(v)，一条边 e=(u,v)连接 u 和 v，则 u 和 v 是 e 的端点。节点 u 的度是与它连接的边的数目。

完全图：（如图 3.2）一个有 n 个节点的完全图是这样一个图，它的每一个节点都与其它所有节点邻接。

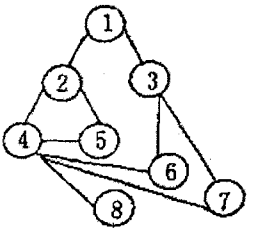


图 3.1 图

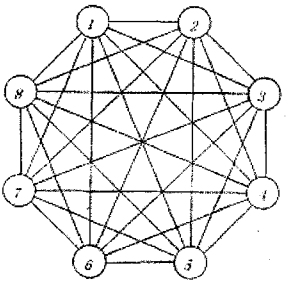


图 3.2 完全图

连通性：如果对于图中的每一对顶点都存在一条链把它们联结起来，则称该图是连通的。

环路：图 G 的边 e 是 V 的两个元素 v_i，v_j 的无序对（v_i，v_j），称 v_i，v_j 是 e 的端点。当 $v_i=v_j$ 时，称 e 为环。

无向图和有向图：一般来说，图是由回路和边组成，在图中有些边是带方向的，而有些边是不带方向的。不带方向的边叫无向边(undirected edge),带方向的边叫有向边（directed edge）。如果用（i,j）来表示一条边，对于无向边来说，（i,j）与（j,i）是一样的，而对于有向边来说，（i,j）与（j,i）是不同的。

如果图中所有的边都是无向边，那么该图叫作无向图（undirected graph），如果所有的边都是有向的，那么该图叫作有向图（directed graph）。

加权图：在一些无向图和有向图的应用中，为每条边赋予一个权或耗费，这时的图分别称为加权无向图（weighted digraph）和加权有向图（weighted graph），一般统称为网路（network）。

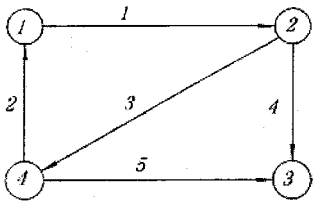
图的邻接矩阵表示：一个 n 顶点的图 G=(V,E)的邻接矩阵（adjacency matrix）是一个 nxn 的矩阵 A，A 中的每一个元素是 0 或 1，假设 V={1,2,...,n}，如果 G 是一个无向图，那么 A 中的元素定义如下：

$$A(i,j)=\begin{cases}1 & \text{如果 } (i,j)\in E \text{或 } (j,i)\in E\\ 0 & \text{其它}\end{cases}$$

如果 G 是有向图，A 中的元素定义如下：

$$A(i,j)=\begin{cases}1 & \text{如果 } (i,j)\in E\\ 0 & \text{其它}\end{cases}$$

例如图 3.3 所示的图结构，A1 表示其对应的无向图，A2 表示其有向图。



$$A_1=\begin{bmatrix}0&1&1&0\\1&0&1&1\\1&1&0&1\\0&1&1&0\end{bmatrix}\qquad A_2=\begin{bmatrix}0&1&0&0\\0&0&1&1\\1&0&0&1\\0&0&0&0\end{bmatrix};$$

图 3.3 图的邻接矩阵表示

图的搜索算法[28]：从一个给定的顶点开始，访问能够到达的所有顶点。（当且仅当存在一条从 v 到 u 的路径时，顶点 v 可到达顶点 u。）搜索这些顶点的两种标准方法是广度优先搜索和深度优先搜索。

1、　广度优先搜索

从一个顶点开始，识别所有可到达顶点的方法叫作广度优先搜索（Breadth-First Search, BFS）。下面给出实现广度优先搜索的伪代码：

```
//从顶点 v 开始的广度优先搜索
把顶点 v 标记为已到达顶点;
初始化队列 Q，其中仅包含一个元素;
while (Q 不空) {
从列队中删除顶点 w;
    令 u 为为邻接于 w 的顶点;
    while(u){
    if(u 尚未被标记){
     把 u 加入队列;
     把 u 标记为已到达顶点; }
     u=邻接于 w 的下一个顶点;
    }
}
```

2、　深度优先搜索

深度优先搜索（Depth－First Search, DFS）是另一种搜索方法。从顶点 v 出发，DFS 按如下过程进行：首先将 v 标记为已到达顶点，然后选择一个与 v 邻接的尚未到达的顶点 u，如果这样的 u 不存在，搜索终止。假设这样的 u 存在，那么从 u 又开始一个新的 DFS。当从 u 开始的搜索结束时，再选择另外一个与 v 邻接的尚未到达的顶点，如果这样的顶点不存在，那么搜索终止。而如果存在这样的顶点，又从这个顶点开始 DFS，如此循环下去。

广度优先搜索和深度优先搜索在图的搜索算法中都很流行，但比较而言，深度优先搜索使用的频率更高一些，相应的效率也高一些。

最短路径[29]：设 G=(V,E)是具有边权的有向图，w(e)为边 e 的长度，则路径 $p=v_0,v_1,\ldots v_k$，从 v0 到 vk 的长度 w(p)为：

$$w(p)=\sum_{i=1}^{k} w(v_{i-1},v_i)$$

有三类最短路径问题：单点对最短路径（Single-Pair Shortest-Path, SPSP）问题、单源点最短路径（Single-Source Shortest-Path, SSSP）问题和全点对最短路径（All-Pairs Shortest-Path, APSP）问题。本文中要用到单点对最短路径的概念，所以，这里只给出单点对最短路径的定义：

单点对最短路径：给定 s, t∈V, 求任一从 s 到 t 的具有最小长度 d(s, t)的路径，若不能到达 t，则 d(s, t)=∞, d(s, t)称为从 s 到 t 的距离（distance），s 为源点(source)，t 为目标点（target）。

3．2．2　动态规划算法

动态规划是数学规划中的一个分支，主要研究和解决多阶段决策过程的最优化问题。1951 年美国数学家 R. Bellman 等人创立了“动态规划”。

动态规划[30,31,32]是解决多阶段决策过程最优化的一种方法。动态规划的主要依据是最优化原理：一个最优策略具有这样的性质：不论初始状态和初始决策如何，相对于第一个决策所形成的状态来说，余下的决策必定构成一个最优策略。从概念上讲，可把这个原理看作是：如果已经给出从点 A 到点 C 的最优轨迹，那

2、　深度优先搜索

深度优先搜索（Depth－First Search, DFS）是另一种搜索方法。从顶点 v 出发，DFS 按如下过程进行：首先将 v 标记为已到达顶点，然后选择一个与 v 邻接的尚未到达的顶点 u，如果这样的 u 不存在，搜索终止。假设这样的 u 存在，那么从 u 又开始一个新的 DFS。当从 u 开始的搜索结束时，再选择另外一个与 v 邻接的尚未到达的顶点，如果这样的顶点不存在，那么搜索终止。而如果存在这样的顶点，又从这个顶点开始 DFS，如此循环下去。

广度优先搜索和深度优先搜索在图的搜索算法中都很流行，但比较而言，深度优先搜索使用的频率更高一些，相应的效率也高一些。

最短路径[29]：设 G=(V,E)是具有边权的有向图，w(e)为边 e 的长度，则路径 $p=v_0,v_1,\ldots v_k$，从 v0 到 vk 的长度 w(p)为：

$$w(p)=\sum_{i=1}^{k} w(v_{i-1},v_i)$$

有三类最短路径问题：单点对最短路径（Single-Pair Shortest-Path, SPSP）问题、单源点最短路径（Single-Source Shortest-Path, SSSP）问题和全点对最短路径（All-Pairs Shortest-Path, APSP）问题。本文中要用到单点对最短路径的概念，所以，这里只给出单点对最短路径的定义：

单点对最短路径：给定 s, t∈V, 求任一从 s 到 t 的具有最小长度 d(s, t)的路径，若不能到达 t，则 d(s, t)=∞, d(s, t)称为从 s 到 t 的距离（distance），s 为源点(source)，t 为目标点（target）。

3．2．2　动态规划算法

动态规划是数学规划中的一个分支，主要研究和解决多阶段决策过程的最优化问题。1951 年美国数学家 R. Bellman 等人创立了“动态规划”。

动态规划[30,31,32]是解决多阶段决策过程最优化的一种方法。动态规划的主要依据是最优化原理：一个最优策略具有这样的性质：不论初始状态和初始决策如何，相对于第一个决策所形成的状态来说，余下的决策必定构成一个最优策略。从概念上讲，可把这个原理看作是：如果已经给出从点 A 到点 C 的最优轨迹，那

么，从任一中间点 B 到点 C 的那部分轨迹也必定是 B 到 C 的最优轨迹。

动态规划是一个自下而上的技术，应用于可以分解的优化问题，若每个形式化的问题可以分为几个小的问题，且这些子问题的解可以很快得到，并可生成原始问题的解，则称该优化问题是可以分解的。动态规划实际上是一种枚举方法，只是在列举所有可行解时从最后的决策倒退到较早的决策。

动态规划的实例：

图 3.4 所示是分层网络中最短路径的动态规划求解方法，从节点 s 到达节点 t 的最短路径。

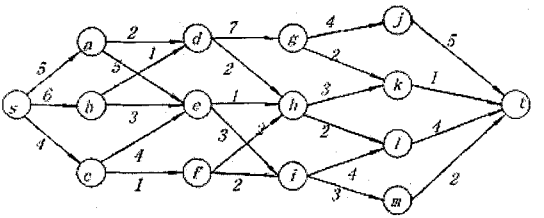


图 3.4 分层网络中最短路径问题

假设 x 表示节点号，$f_i(x)$表示从第 i 阶段 x 节点到达终点的最短距离，$d_i(x)$表示从第 i 阶段 x 节点到达终点取最短路径时应到达的下一个节点号。

（1）、假设已到达第 5 阶段的 j,k,l,m 节点，若已到达节点 j，则只有一条路可到达 t，故 $f_5(j)=5$，$d_5(j)=t$ 同理可得 $f_5(k)=5$，$d_5(k)=t$；$f_5(l)=4$，$d_5(l)=t$；$f_5(m)=2$，$d_5(m)=t$。

（2）、假设已到达第四阶段的 g,h,i 节点。若已到达节点 g，则有两条路可到达 t,此时应检查两条路的距离的长短，显然可得 $f_4(g)=3$，$d_4(g)=k$；同理可得 $f_4(h)=4$，$d_4(h)=k$；$f_4(i)=5$，$d_4(i)=m$。

同样的方法可以得到其它阶段的结果如下：

$f_3(d)=6$，$d_3(d)=h$；$f_3(e)=5$，$d_3(e)=h$；$f_3(f)=6$，$d_3(f)=h$；

$f_2(a)=8$，$d_2(a)=d$；$f_2(b)=7$，$d_2(b)=d$；$f_2(c)=7$，$d_2(c)=f$；

$f_1(s)=11$，$d_1(s)=c$；

根据计算结果，可以找出从节点 s 到节点 t 的最短路径为 s-c-f-h-k-t。

3．2．3　斯坦纳树和最小生成树

当线网布线以加权线长最短为优化目标时，对于多端线网的互联实现应以构造最小斯坦纳（Steiner）树为宜。Steiner 树问题就是利用可选点集 V 中点之间的互相连线，使点集 P⊆V 中的点互相连通。将 P 中的点称为顶点，将从 V 中选出进行互连但不属于 P 的点称为 Steiner 点。这两种点和它们之间的连线构成的树称为 Steiner 树。

最小 Steiner 树（Minimum Steiner Tree, SMT）[32,33,34] 问题是一个 NP－困难问题(一类不能在多项式时间内求解的问题)，通常采用启发式算法求解。在欧几里德空间中 Steiner 树各点之间连线的角度是没有限制的。在电路布线的应用中，由于工艺的原因，连线的角度受到一定的限制。构造 90 度的 Steiner 树方法比较成熟。

第四章　智能识点法布线理论

这一章介绍本论文的核心部分——智能识点法的布线理论。

4.1 智能识点法概述

智能识点法布图理论是本文作者通过总结前人主要布线算法，借鉴了迷宫算法和线探索布线算法的优点，并努力克服这两类算法的缺点，运用相关的数学知识，设计的布线算法。该算法既适用于 PCB 的布线，也可以运用在集成电路的布线中。

迷宫算法以其极强的绕障能力被广泛应用。但迷宫算法最大的缺点是数据存储空间和路径搜索时间随布线间距的减少以平方关系增加，不能适应工程上遇到的各种高密度、高精度电路自动布线的要求。Hightower[13]和 Mikam[14]分别提出了线探索的思想和算法，由于 Hightower 试图探索尽量小的搜索空间，虽然探索的速度加快，但往往因为没有探索足够大的空间而找不到本该存在的路径。Mikami 算法应用了广度优先搜索的思想，它保证能找到一条存在的通路，但需要较多的空间和时间，且找到的路径不一定最短。

当人行走遇到障碍物时，他会直接搜寻障碍物的端点，通过障碍物的端点而绕过障碍物。根据这种人工智能的绕障思维，在布线中出现的障碍物的周围定义一些能绕过它的点，称为绕障点，以绕障点和待布点组成布线点集合，将布线问题转化为在点集合中求布线点间的最短路径，从而得到最优布线路径。我们称这种方法为智能识点法。

根据实际的需要，我们主要以单层 PCB 的布线为研究对象构造算法。相对来讲，单层 PCB 布线要比多层布线难，这主要体现在：由于布线层只有一层，在这一层上既要布水平线，还要布垂直线和±45 度方向的斜线。各种线型之间相互干扰大，线网布通的困难加大。为了适应微电子行业的发展趋势，本文算法的布线模型为 MD 布线模型（Manhattan-Diagonal model），但对于其它布线模型也同样适用。

4．2线网布线的设计目标

4．2．1布图综合考虑的设计目标

在电路板与集成电路的布图设计中需要综合考虑以下设计目标：

1、满足电路的各项电性能参数的要求及工艺要求。

2、使设计结果具有足够高的集成度。

3、使设计过程具有足够高的设计成功率，其中主要的是指布线成功率。

4、使设计结果具有较高的设计质量，如尽可能减少通孔数量、缩短连线长度、提高电性能等。

5、考虑设计周期、设计成本以及结果验证等。

6、考虑具体电路的设计过程。根据特定的工艺条件，往往还会对布图设计提出一些专门的要求。如某些线网的树型要求、长度限制等。

4．2．2．线网布线的一般要求

电路板在布局阶段已经将电路模块的位置以及模块的引脚位置确定下来，接下来就是要实现各模块之间的连接。互联信息由网络表提供，这是由电路本身的要求决定的，模块之间的连接由导线实现。

我们把电路板上可以布线的区域称为布线区域。线网必须布在布线区域内，且不能超过布线区域的容量。布线阶段的首要目标是百分之百地完成引脚间的互联，其次是完成布线的前提下进一步优化布线结果，如缩短线网长度、提高电性能、减少通孔数量、PCB的可靠性和电磁兼容性等。

4．2．3．本文布线算法的设计目标

本论文研究内容是针对实用要求，研究印刷电路板的自动布线的算法理论和实现。它是正在开发的电路CAD软件的一部分，也是该软件的几个核心技术之一。由于本文旨在布线算法的讨论，因此将布线目标定为：在尽量满足百分之百地完成引脚间的互联的前提下，使所有布线的连线总长最短。

4．3 布线的线网排序

4．3．1 布线顺序对布通率的影响

布线大体上可以分为串行布线和并行布线。正如串行布线的字面意思所表达的那样，它是将待布的线网按某个顺序一条一条地进行布线。一条已布完的线网可能对下一条待布线网产生堵塞等影响。因此串行布线对线网的排序相当敏感，因为先布的线网自由度大，而后布的线网自由度小。为了获得尽可能高的布通率，在布线时，必须考虑当前布线对今后布线的影响，如何估计当前布线对未布线的影响，使已布线不影响未布接点互连的实现，这是一个人所共知的难题，也是影响布线成功率的一个主要原因。如图 4.1、4.2 为先布的线网将使未布线网无法实现互连的两种情况。

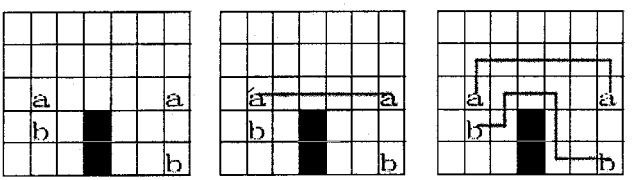


图 4.1 当前布线对未来布线的影响例一

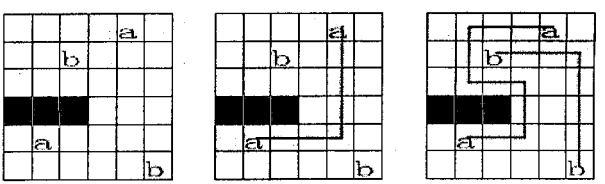


图 4.2 当前布线对未来布线的影响例二

图 4.1 和图 4.2 中，如果先布线网 a，则线网 b 就无法实现互连，若先布 b 则两个线网都可布通。

并行布线算法力求克服串行算法中的线网必须排序这一缺点，同时考虑线网集的所有线网布线。显然，并行算法的计算复杂性很高，甚至于两端线网的布线

4．3 布线的线网排序

4．3．1 布线顺序对布通率的影响

布线大体上可以分为串行布线和并行布线。正如串行布线的字面意思所表达的那样，它是将待布的线网按某个顺序一条一条地进行布线。一条已布完的线网可能对下一条待布线网产生堵塞等影响。因此串行布线对线网的排序相当敏感，因为先布的线网自由度大，而后布的线网自由度小。为了获得尽可能高的布通率，在布线时，必须考虑当前布线对今后布线的影响，如何估计当前布线对未布线的影响，使已布线不影响未布接点互连的实现，这是一个人所共知的难题，也是影响布线成功率的一个主要原因。如图 4.1、4.2 为先布的线网将使未布线网无法实现互连的两种情况。

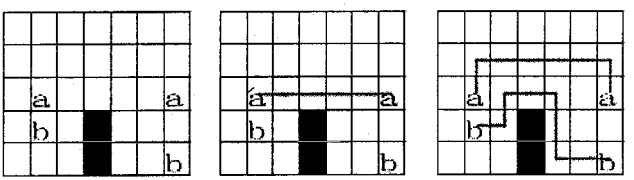


图 4.1 当前布线对未来布线的影响例一

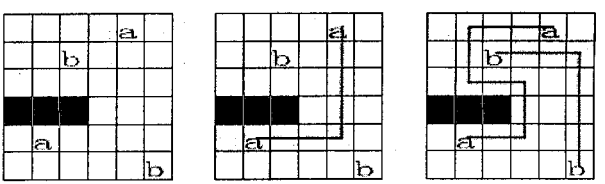


图 4.2 当前布线对未来布线的影响例二

图 4.1 和图 4.2 中，如果先布线网 a，则线网 b 就无法实现互连，若先布 b 则两个线网都可布通。

并行布线算法力求克服串行算法中的线网必须排序这一缺点，同时考虑线网集的所有线网布线。显然，并行算法的计算复杂性很高，甚至于两端线网的布线

问题也没有一种有效的多项式算法。在并行算法中通常使用的是整数规划方法，为了解决整数规划在总体布线中因问题规模太大而效率不高的缺点，人们常采用自顶向下的层次式方法，即将布线问题层次式地划分成规模较小的布线问题，以便于能用整数规划方法有效地解决总体布线问题。

4．3．2 布线顺序常用处理方法

在要求不太高的布线问题中，大量地采用一些简捷的顺序处理方法。

1、短线序：这种方法采用了启发式原则，即接点间曼哈顿距离较小的连线先布，对今后的布线影响较小，应用这种方法决定布线顺序时，首先定义各线网的尺寸 S_i，i 线网的尺寸 S_i 为覆盖该线网所有接点的最小矩形的半周长。布线时，每次选择当前未布线集中 S_i 最小者先布。

2、线长序：一般来讲，长的线网布线的难度大，留到最后布，这些长线布线时就更困难了，因此有人主张按“布线难度大的线网先布”，即每次布线时选择布线集中 S_i 最大者优先（S_i 定义同上）。

3、点的干扰度序方法[35]：线网的干扰度定义为覆盖线网的最小矩形内其它线网的接点数目。干扰度小的线网优先布线（即不影响其它布线）

4、线干扰度序方法：这种方法考虑的处理原则与点干扰度序很相似，但在度量每条线对今后布线的影响程度时，采用了线干扰度 L_i。如图 4.3 所示，设线网 B 在以线网 A 的接点为顶点的矩形边上的投影分别为 l_{vB} 和 l_{hB}。则 L_A 为 $\sum_{V_i}(l_{vi}+l_{hi})$，其中所有 i 线网都至少有一个接点处于覆盖 A 线网所有接点的最小矩形域内，布线时，每次选择 L_i 值最小者先布线。

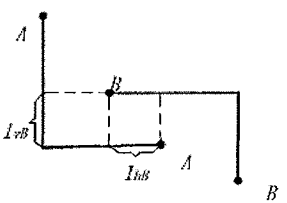


图 4.3 线干扰图示意图

5、整体布线的思想：针对上述情况，J. Soukup[16] 提出了“同时”布线的想

问题也没有一种有效的多项式算法。在并行算法中通常使用的是整数规划方法，为了解决整数规划在总体布线中因问题规模太大而效率不高的缺点，人们常采用自顶向下的层次式方法，即将布线问题层次式地划分成规模较小的布线问题，以便于能用整数规划方法有效地解决总体布线问题。

4．3．2 布线顺序常用处理方法

在要求不太高的布线问题中，大量地采用一些简捷的顺序处理方法。

1、短线序：这种方法采用了启发式原则，即接点间曼哈顿距离较小的连线先布，对今后的布线影响较小，应用这种方法决定布线顺序时，首先定义各线网的尺寸 S_i，i 线网的尺寸 S_i 为覆盖该线网所有接点的最小矩形的半周长。布线时，每次选择当前未布线集中 S_i 最小者先布。

2、线长序：一般来讲，长的线网布线的难度大，留到最后布，这些长线布线时就更困难了，因此有人主张按“布线难度大的线网先布”，即每次布线时选择布线集中 S_i 最大者优先（S_i 定义同上）。

3、点的干扰度序方法[35]：线网的干扰度定义为覆盖线网的最小矩形内其它线网的接点数目。干扰度小的线网优先布线（即不影响其它布线）

4、线干扰度序方法：这种方法考虑的处理原则与点干扰度序很相似，但在度量每条线对今后布线的影响程度时，采用了线干扰度 L_i。如图 4.3 所示，设线网 B 在以线网 A 的接点为顶点的矩形边上的投影分别为 l_{vB} 和 l_{hB}。则 L_A 为 $\sum_{V_i}(l_{vi}+l_{hi})$，其中所有 i 线网都至少有一个接点处于覆盖 A 线网所有接点的最小矩形域内，布线时，每次选择 L_i 值最小者先布线。

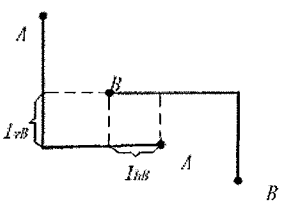


图 4.3 线干扰图示意图

5、整体布线的思想：针对上述情况，J. Soukup[16] 提出了“同时”布线的想

法，也就是整体布线的思想。

上述各种启发式的顺序决定方法都具有处理简捷的优点，而且对不少实际问题是有效的，共同的问题是对不同的问题处理效果不稳定。对于复杂的问题，处理的结果不理想。

4．3．3 干扰图排序方法

由于本文主要是针对单层 PCB 讨论布线算法，布线模型为 MD 模型如图 4.4

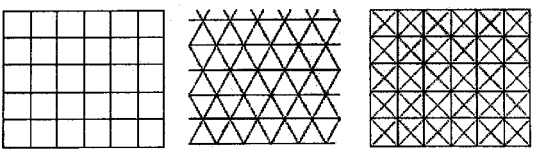


图 4.4　布线几何结构　(a) 2-几何结构(b) 3-几何结构(c) 4-几何结构（MD 模型）

由于在 PCB 一层上既要布水平线，还要布垂直线和 45 度方向的斜线。各种线型之间相互干扰大，线网的布通的困难加大。为了减少线网之间的相互堵塞，线网的布线顺序变得十分重要。本文以文献【36】中所述干扰图（Interference graph）为基础，结合本文算法特点，对线网进行排序处理。事实上，排序和布线是同时进行的，为了叙述方便，我们先来解释干扰图排序方法。

一、干扰图的定义

L 表示一组未连接的布线点对集合，G 表示 PCB 上可布线区域，即 PCB 板上去掉已经存在的布线障碍——先布线、焊盘、引脚、过孔等的区域。利用智能识点法产生最短布线路径的处理方法（详细请看后面章节，以后简称“最短路径”），首先为 L 中的各布线点对产生临时最短路径，产生临时最短路径时允许各临时路径之间相交（即先布线不作为后布线的障碍），我们把 L 中点对按此方法产生的路径称为临时路径，记为 R(L)，例如给出一组五对的待连接点：L={(a,a),(b,b),(c,c),(d,d),(e,e)}，在 MD 模型中如图 4.5，应用产生临时最短路径的方法，获得临时路径 R（L）={A,B,C,D,E}，如图 4.6 所示，产生临时最短路径时，允许各路径之间相交，但不能与固有障碍相交。

法，也就是整体布线的思想。

上述各种启发式的顺序决定方法都具有处理简捷的优点，而且对不少实际问题是有效的，共同的问题是对不同的问题处理效果不稳定。对于复杂的问题，处理的结果不理想。

4．3．3 干扰图排序方法

由于本文主要是针对单层 PCB 讨论布线算法，布线模型为 MD 模型如图 4.4

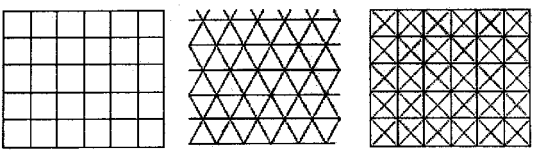


图 4.4　布线几何结构　(a) 2-几何结构(b) 3-几何结构(c) 4-几何结构（MD 模型）

由于在 PCB 一层上既要布水平线，还要布垂直线和 45 度方向的斜线。各种线型之间相互干扰大，线网的布通的困难加大。为了减少线网之间的相互堵塞，线网的布线顺序变得十分重要。本文以文献【36】中所述干扰图（Interference graph）为基础，结合本文算法特点，对线网进行排序处理。事实上，排序和布线是同时进行的，为了叙述方便，我们先来解释干扰图排序方法。

一、干扰图的定义

L 表示一组未连接的布线点对集合，G 表示 PCB 上可布线区域，即 PCB 板上去掉已经存在的布线障碍——先布线、焊盘、引脚、过孔等的区域。利用智能识点法产生最短布线路径的处理方法（详细请看后面章节，以后简称“最短路径”），首先为 L 中的各布线点对产生临时最短路径，产生临时最短路径时允许各临时路径之间相交（即先布线不作为后布线的障碍），我们把 L 中点对按此方法产生的路径称为临时路径，记为 R(L)，例如给出一组五对的待连接点：L={(a,a),(b,b),(c,c),(d,d),(e,e)}，在 MD 模型中如图 4.5，应用产生临时最短路径的方法，获得临时路径 R（L）={A,B,C,D,E}，如图 4.6 所示，产生临时最短路径时，允许各路径之间相交，但不能与固有障碍相交。

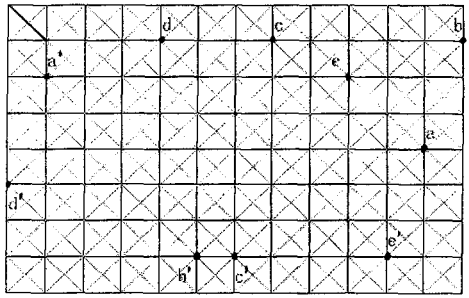


图 4.5 待布的五对点

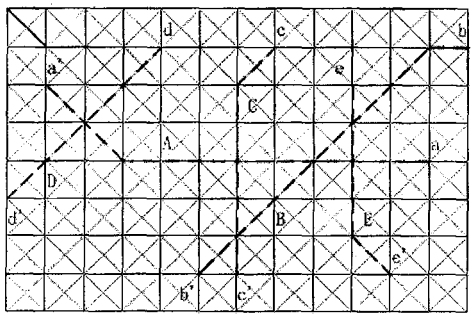


图 4.6　R(L)={A,B,C,D,E}的临时布线

一旦获得 L 的临时路径 R(L)，我们就可以定义一个 R(L)的权重图 I[R(L)]，沿用文献【36】的提法，称之为干扰图（Interference graph）。干扰图定义方法如下：

1）I[R（L）]图中每个顶点 P_i，代表 R（L）的一个路径。

2）在 I[R(L)]图中，当且仅当 P_i，和 P_j 路径相交时有直接边(P_i, P_j)和(P_j, P_i)，以此来表示 P_i 和 P_j 相互之间有干扰存在。

3）每条边（P_i, P_j）的权重 w（P_i, P_j）定义如下：

对于路径 $P_j(s_j, t_j)$，在不与 P_i 路径相交并允许 P_i 以外的 P_k 相交的约束下，重新产生一条路径 P_j'，如果该 P_j'存在，则 w（P_i, P_j）$\triangleq$ | P_j'| − | P_j|，即变化后的 P_j'的长度| P_j'|减去变化前的长度| P_j|，如果|P_j'不存在，则定义 w（P_i, P_j）$\triangleq\infty$；如果 w（P_i, P_j）<∞，表示为了绕过 P_i，P_j必须增加长度。这样 w（P_i, P_j）就可以

看成 P_i 对发现路径 $P_j(s_j, t_j)$ 的一种干扰。

4）每个顶点 P_i 的权重 w（P_i）如下确定：

a、如果 P_i 是孤立的点 w（P_i）$\stackrel{\Delta}{=}0$；

b、如果有一边权重为∞，则 w（P_i）$\stackrel{\Delta}{=}\infty$；

c、如果 0 <w（P_i）<∞，则表示有干扰产生。用 w_{out}（P_i）表示 P_i 对其它线网的影响，w_{in}（P_i）表示 P_i 受其它线网的影响。以 deg(P_i)表示 P_i 相交的图线的数目，则 w（P_i）定义如下：

$$w(P_i) \stackrel{\Delta}{=} w_{out}(P_i) - w_{in}(P_i) + k\ deg(P_i);$$

其中 k 是一个经验值，一般情况下可取 5。这样 w（P_i）就表示了 P_i 对其它未连接线网的综合干扰程度，因此在权重图 I[R（L）]中，对任意 P_i 和 P_j，如果有 w（P_i）< w（P_j）<∞，我们可以认为 P_i 对其它线网寻找路径的干扰小于 P_j。这里，我们把 w（P_j）称为 P_i 的干扰度。

例如对于图 4.6 中的布线情况，我们可以构造出干扰图 I[R（L）]如下：

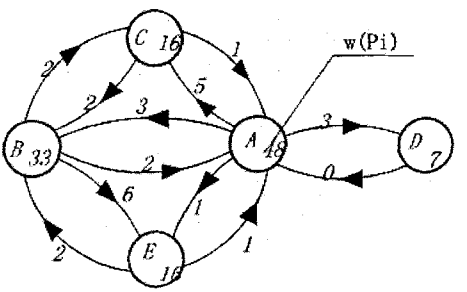


图 4.7　与图 4.6 对应的干扰图 I[R(L)]

为了便于读者理解，我们在干扰图中所标注的权重作了简化如下：

以图中每小格为单位，且认为每个单位格的对角线的长度等于每格的宽度，事实上它们之间有 $\sqrt{2}$ 倍的关系，即：给出一对点 v(x_v, y_v)和 w(x_w, y_w)，它们之间的长度为：

$$d_{min}(v,w) \stackrel{\Delta}{=} \max\left[|x_v - x_w|, |y_v - y_w|\right];$$

在实际布线中，应根据实用精度的需要来计算线的长度。

二、干扰图排序方法

由于干扰图中权重 w（P_i）的大小代表了 P_i 对其它线网寻找路径的干扰程度，因此根据干扰度（w（P_i）值）的升序来排列干扰图中的各顶点(即布线顺序的排列)，具有很强的合理性。能较好地解决串行布线时，线网间的相互干扰问题。具体处理如下：

1、将待布点和 PCB 中的固有障碍（即焊盘、引脚等）放在 OB 中。

2、将待布点对组成点对集合 L，暂时的路径放于 R（L）中，按干扰图的定义构造干扰图，并计算干扰图中各顶点的干扰度。

3、将各顶点按干扰度的升序排列，并将结果放在 U 中。

4、取出 U 中的第一个顶点 P_i（干扰度最小的点），并从 L 中取出该顶点所代表的布线点对，将 OB 中的元素设为布线障碍，使用智能识点法产生最短布线路径（见后面章节的叙述）的方法产生最短布线路径，并将该布线结果存入 OB 中作为以后布线的障碍。

5、从干扰图中删除 P_i，从 L 中删除对应的布线点对。

6、判断，如果 L＝ϕ，则表明所有的待布点对都已经完成布线，结束。若 L≠ϕ，则返回到 1，继续处理。下面给出图 4.5 所示的五对点算例：

1、OB＝（a, a′, b, b′, c, c′, d, d′, e, e′）;

2、L={（a, a′）,（b, b′）,（c, c′）,（d, d′）,（e, e′）}，R（L）={A, B, C, , D, E}，构造的干扰图 I[R（L）]如图 4.7 所示。

3、将 I[R（L）]中各顶点按升序排序，U＝{D, E, C, B, A}；

4、取出 D 所代表的点对{(d, d′)，使用智能识点法产生最短路径，并将该布线结果存入 OB 中作为以后布线的障碍，OB＝（a, a′, b, b′, c, c′, d, d′, e, e′，{d, d}）;

5、从干扰图中删除 D，从 L 中删除对应的布线点对(d, d)，从 R（L）删除 D;

6、L={（a, a′）,（b, b′）,（c, c′）,（e, e′）}，R（L）={A, B, C, E}；

7、重新产生临时布线路径（图 4.8），重新构造干扰图（图 4.9）；

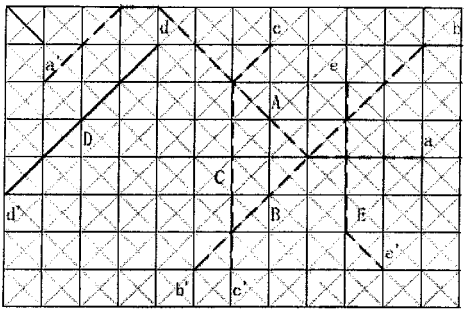


图 4.8　R(L)={A,B,C,E}的临时布线

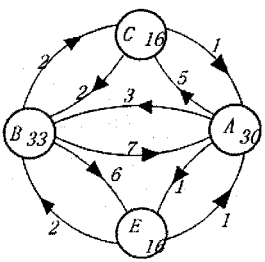


图 4.9　与图 4.8 对应的干扰图 I[R(L)]

8、从干扰图中可以看出{c, c'}, {e, e'}为干扰度最小的线网，使用智能识点法产生最短布线路径，并将该布线结果存入 OB 中作为以后布线的障碍，OB＝(a, a', b, b', c, c', d, d', e, e', { d, d }, {c, c'}, {e, e'});

9、从干扰图中删除 C，E,从 L 中删除对应的布线点对(c, c')，（e, e'），从 R（L）删除 C，E;

10、L={ (a, a') }, (b, b')}，R（L）={A, B};

11、重新产生临时布线路径（图 4.10），重新构造干扰图（图 4.11）；

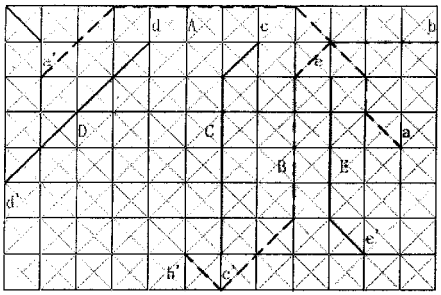


图 4.10　R(L)={A,B}的临时布线

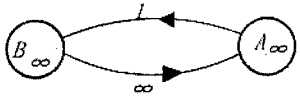


图 4.11　干扰图 I[R(L)]

12、从干扰图中可以看出{a, a′}为干扰度最小的线网，使用智能识点法产生最短布线路径，并将该布线结果存入 OB 中作为以后布线的障碍，OB＝（a, a′, b, b′, c, c′, d, d′, e, e′，｛d, d｝，｛c, c′｝, {e, e′}, {a, a′}）；

13、L=｛(b, b′)｝，R（L）=｛B｝；

14、使用智能识点法产生最短布线路径，并将该布线结果存入 OB 中作为以后布线的障碍，OB＝（a, a′, b, b′, c, c′, d, d′, e, e′，｛d, d｝，｛c, c′｝, {e, e′}, {a, a′}，｛b, b′｝），布线结果如图 4.12；

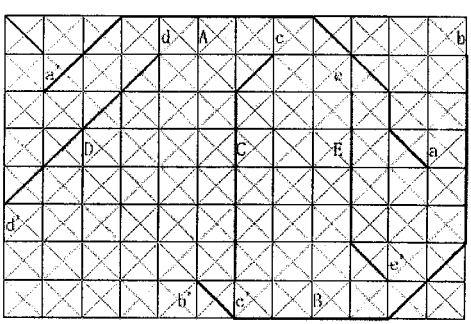


图 4.12　最终布线结果

15、L=ϕ，终止布线，所有线网布线完成。

4．4 智能识点法布线的绕障基本线型

电路布线要求，所布的导线不能与已有障碍物相交。如何能使所布导线实现自动搜索，绕过障碍物，并能使布线路径最短，一直是布线算法所面临的关键问题，也是难题之一。这里我们先讨论在 MD 模型上实现具有两个端点 s（起始点）、t（目标点）的线网互连的问题。为了使所布导线能绕过障碍，本文规定两种线型：

1．简单线段；

2．复合线段。

4．4．1 绕障基本线型 1——简单线段

简单线段分为三种：水平线段；垂直线段；±45° 斜线。这种线段为两个待连接点水平坐标或垂直坐标相同，可以直接用水平线或垂直线直接连线；或两个待连接点水平坐标差与垂直坐标差的绝对值相同，可以用±45 度线段直接连线的情况。该线型连接简单，是两点连线中距离最短的（欧氏距离）的一种线型（与复合线段相比），但要求满足一定的条件（见表 4.1）。两点的 MD 线长（连线在 MD 模型上的线长，含义见图 4.13）计算见表 4.1。

4．4．2 绕障基本线型 2——复合线段

复合线段是由一条水平（垂直）线和一条＋45° （－45° ）斜线组合而成，这种线型是为适应 MD 布线模型的需要而定义。当 MD 布线模型上不能用简单线段实现两点的互连时，就需要采用复合线段。复合线段分为Ⅰ型和Ⅱ型两种，其中Ⅰ型为复合线段中的水平、垂直线段在布线中优先搜索，当无法用水平、垂直线段搜索路径时再转折为±45°斜线搜索。Ⅱ型则相反，优先用±45°斜线搜索，再转折为水平、垂直线段进行路径搜索。本文规定的基本线型如表 4.1 。

4．4．3 广义线段

为叙述方便，我们将简单线段和复合线段统称为广义线段。表 4.1 表示了

4．4 智能识点法布线的绕障基本线型

电路布线要求，所布的导线不能与已有障碍物相交。如何能使所布导线实现自动搜索，绕过障碍物，并能使布线路径最短，一直是布线算法所面临的关键问题，也是难题之一。这里我们先讨论在 MD 模型上实现具有两个端点 s（起始点）、t（目标点）的线网互连的问题。为了使所布导线能绕过障碍，本文规定两种线型：

1．简单线段；

2．复合线段。

4．4．1 绕障基本线型 1——简单线段

简单线段分为三种：水平线段；垂直线段；±45° 斜线。这种线段为两个待连接点水平坐标或垂直坐标相同，可以直接用水平线或垂直线直接连线；或两个待连接点水平坐标差与垂直坐标差的绝对值相同，可以用±45 度线段直接连线的情况。该线型连接简单，是两点连线中距离最短的（欧氏距离）的一种线型（与复合线段相比），但要求满足一定的条件（见表 4.1）。两点的 MD 线长（连线在 MD 模型上的线长，含义见图 4.13）计算见表 4.1。

4．4．2 绕障基本线型 2——复合线段

复合线段是由一条水平（垂直）线和一条＋45° （－45° ）斜线组合而成，这种线型是为适应 MD 布线模型的需要而定义。当 MD 布线模型上不能用简单线段实现两点的互连时，就需要采用复合线段。复合线段分为Ⅰ型和Ⅱ型两种，其中Ⅰ型为复合线段中的水平、垂直线段在布线中优先搜索，当无法用水平、垂直线段搜索路径时再转折为±45°斜线搜索。Ⅱ型则相反，优先用±45°斜线搜索，再转折为水平、垂直线段进行路径搜索。本文规定的基本线型如表 4.1 。

4．4．3 广义线段

为叙述方便，我们将简单线段和复合线段统称为广义线段。表 4.1 表示了

绕障基本线型的详细的定义方法和 MD 线长计算的参考公式。

表 4.1 MD 布线模型上的基本线型

线型名称		坐标条件	图示	MD 线长
简单线段		$y_s=y_t$ $x_s\neq x_t$		$\lvert x_t-x_s\rvert$
		$x_s=x_t$ $y_s\neq y_t$		$\lvert y_t-y_s\rvert$
		$\lvert x_t-x_s\rvert=$ $\lvert y_t-y_s\rvert$		$1.4*$ $\lvert x_t-x_s\rvert$
复合线段	I 型（水平、垂直线优先）	$\lvert x_t-x_s\rvert$ $>=$ $\lvert y_t-y_s\rvert$		$\lvert x_t-x_s\rvert$ $+0.4*$ $\lvert x_t-x_m\rvert$
		$\lvert x_t-x_s\rvert<$ $\lvert y_t-y_s\rvert$		$\lvert y_t-y_s\rvert$ $+0.4*$ $\lvert y_t-y_m\rvert$
	II 型（45°线优先）	$\lvert x_t-x_s\rvert$ $>=$ $\lvert y_t-y_s\rvert$		$\lvert x_t-x_s\rvert$ $+0.4*$ $\lvert x_s-x_m\rvert$
		$\lvert x_t-x_s\rvert<$ $\lvert y_t-y_s\rvert$		$\lvert y_t-y_s\rvert$ $+0.4*$ $\lvert y_s-y_m\rvert$

表中图形上标记的 m 点为水平、垂直线与±45°线相连接的点。MD 线长是指 MD 布线模型上计算布线长度的计算方法，复合线段的 MD 线长计算为：水平、垂直线段线长与±45°斜线线长（欧氏线长）之和，计算时可参照表 4.1 中的计算公式，用表中的计算方法可加快计算速度。在实际应用中应考虑布线的精度要求。关于各种线长的概念见图 4.13，连接 A、B 两点的各种线长为：

欧氏线长：　L　；

曼哈顿线长：H2＋V　；

MD 线长：H1＋R　。

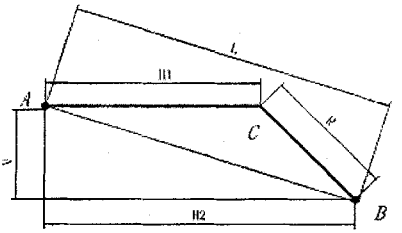


图 4.13 各种线长的含义

4．5 智能识点法布线的绕障点的规定

4．5．1 布线中的障碍物

在印刷电路板的自动布线过程中往往会遇到障碍物。布线中的障碍物可以分为两种：一种为印刷电路板上固有的障碍物，如焊盘、过孔、引脚等，另一种是已布出的走线，对于串行布线，已布线是未布线的障碍。障碍物的外形是指已存在的布放元素在布线平面上所占的外形区域，像焊盘类障碍物的外形就可以通过元件的封装外形表现出来，而元件的封装形式等参数就包含在网络表中该元件的声明部分。走线则是在自动布线过程中逐渐生成的线型障碍物。所有障碍物的外形集合构成了布线区域内的障碍区，正是由于障碍物的存在缩小了可供布线的区域面积。在智能识点法自动布线中，我们将布线中固有的障碍物简化成几种常见的几何图形：圆形、矩形、三角形，并将圆形进一步简化为正多边形（通常为正八边形）。

4．5．2 绕障点的规定

为了使所布导线能自动绕过障碍，我们模仿人的智能化行为。当人行走遇到障碍物时，他会直接搜寻障碍物的端点，通过障碍物的端点而绕过障碍物。我们模仿这种人工智能的绕障思维，在布线中出现的障碍物的周围定义一些能绕过它的点，称为绕障点

绕障点的规定原则：

在障碍物周围，连接若干条广义线段将该图形以线网布线允许的最小间隙

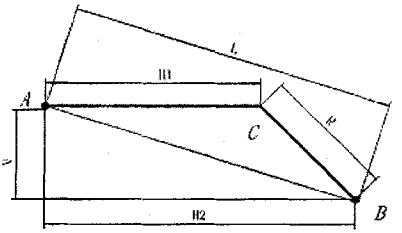


图 4.13 各种线长的含义

4．5 智能识点法布线的绕障点的规定

4．5．1 布线中的障碍物

在印刷电路板的自动布线过程中往往会遇到障碍物。布线中的障碍物可以分为两种：一种为印刷电路板上固有的障碍物，如焊盘、过孔、引脚等，另一种是已布出的走线，对于串行布线，已布线是未布线的障碍。障碍物的外形是指已存在的布放元素在布线平面上所占的外形区域，像焊盘类障碍物的外形就可以通过元件的封装外形表现出来，而元件的封装形式等参数就包含在网络表中该元件的声明部分。走线则是在自动布线过程中逐渐生成的线型障碍物。所有障碍物的外形集合构成了布线区域内的障碍区，正是由于障碍物的存在缩小了可供布线的区域面积。在智能识点法自动布线中，我们将布线中固有的障碍物简化成几种常见的几何图形：圆形、矩形、三角形，并将圆形进一步简化为正多边形（通常为正八边形）。

4．5．2 绕障点的规定

为了使所布导线能自动绕过障碍，我们模仿人的智能化行为。当人行走遇到障碍物时，他会直接搜寻障碍物的端点，通过障碍物的端点而绕过障碍物。我们模仿这种人工智能的绕障思维，在布线中出现的障碍物的周围定义一些能绕过它的点，称为绕障点

绕障点的规定原则：

在障碍物周围，连接若干条广义线段将该图形以线网布线允许的最小间隙

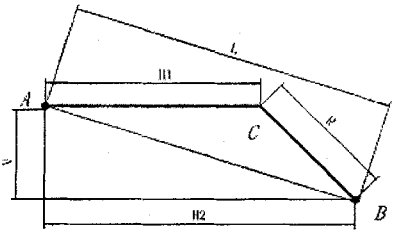


图 4.13 各种线长的含义

4．5 智能识点法布线的绕障点的规定

4．5．1 布线中的障碍物

在印刷电路板的自动布线过程中往往会遇到障碍物。布线中的障碍物可以分为两种：一种为印刷电路板上固有的障碍物，如焊盘、过孔、引脚等，另一种是已布出的走线，对于串行布线，已布线是未布线的障碍。障碍物的外形是指已存在的布放元素在布线平面上所占的外形区域，像焊盘类障碍物的外形就可以通过元件的封装外形表现出来，而元件的封装形式等参数就包含在网络表中该元件的声明部分。走线则是在自动布线过程中逐渐生成的线型障碍物。所有障碍物的外形集合构成了布线区域内的障碍区，正是由于障碍物的存在缩小了可供布线的区域面积。在智能识点法自动布线中，我们将布线中固有的障碍物简化成几种常见的几何图形：圆形、矩形、三角形，并将圆形进一步简化为正多边形（通常为正八边形）。

4．5．2 绕障点的规定

为了使所布导线能自动绕过障碍，我们模仿人的智能化行为。当人行走遇到障碍物时，他会直接搜寻障碍物的端点，通过障碍物的端点而绕过障碍物。我们模仿这种人工智能的绕障思维，在布线中出现的障碍物的周围定义一些能绕过它的点，称为绕障点

绕障点的规定原则：

在障碍物周围，连接若干条广义线段将该图形以线网布线允许的最小间隙

（该间隙为线网间距）包围的最少点即为该几何图形的绕障点。

在绕障点规定的原则中，要注意强调两点：

1、　用广义线段包围障碍物时应使广义线段距离障碍物的轮廓线为最小间隙，否则规定的绕障点布线时会浪费布线区域，或使本身及其它本可以布通的线网无法实现互连。

2、　在满足第一条要求的前提下，尽量使绕障点的数量少。绕障点的数量会直接影响布线所用的时间，绕障点数量越多，布线所需时间越长。

图 4.14 表示几种基本线型绕障点的定义方法。

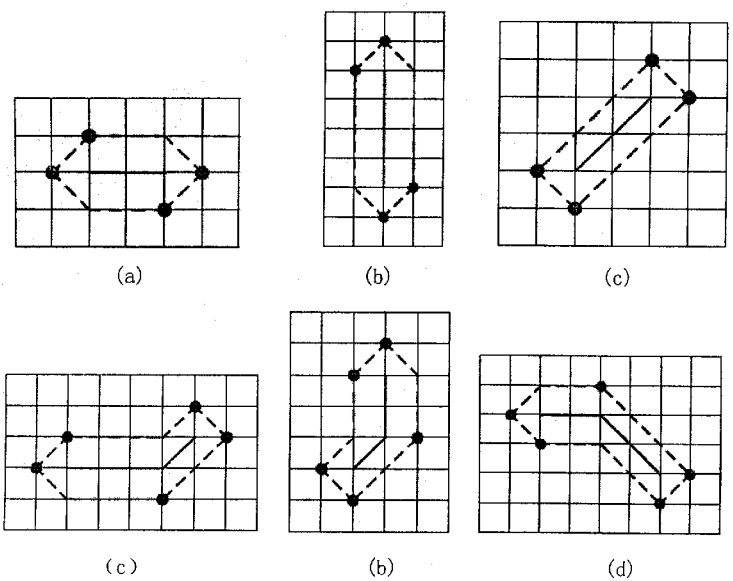


图 4.14　几种基本线性绕障点的定义方法

基本线型的绕障点是定义在广义线段的端点和转折点处，通过这些点可以用最短的路径绕过障碍。表 4.2 中归纳了广义线段的绕障点的规定方法。对于障碍物为其它几何形状时，可根据绕障点的规定原则来确定。图 4.15 中列举了几种常见的封闭的平面几何图形绕障点的确定，其中对于圆形障碍，先将其封装成正外切八边形，然后再规定正外切八边形的绕障点。图中虚线为包围该几何图形的广义线段。绕障点与布线的间距可以根据几何形状要求及各种电器规则来确定。

实验证明这样规定的绕障点完全可以实现以最短路径绕障的功能。

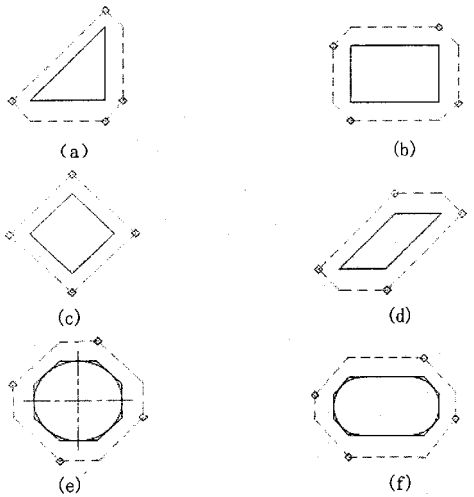


图 4.15 几种常见几何图形的绕障点规定

表 4.2 广义线段绕障点的定义

简单线段（4 个绕障点）	复合线段（5 个绕障点）	
	$\lvert x_t - x_s \rvert >= \lvert y_t - y_s \rvert$	$\lvert x_t - x_s \rvert < \lvert y_t - y_s \rvert$
备注：表中的棱形点为定义的绕障点。		

4．6 智能识点法最短路径搜索

4．6．1 智能识点法最短路径搜索的步骤

智能识点法最短路径搜索算法的执行步骤如下：

Step0：建立绕障点容器 Q（可用 VC 中的 Coblist 类的对象），将所有已存在的障碍物的绕障点取出，经过处理，去掉不可用的绕障点后，存入绕障点容器 Q 中。

Step1：取出欲布线网，将待布点放入绕障点容器 Q 中，以 Q 中所有点为顶点，以连接任两顶点的广义线段为边构造一个带边权值的完全图，以邻接矩阵表示，并用∞（程序中设定一个足够大的数）为每条边初始化。

Step2：判断完全图中的每条边是否与障碍相交，如与任何障碍相交，则该边权值为∞，如不与任何障碍相交，则计算该边的 MD 线长，以计算所得线长作为该边的边权值。如果该边为复合线段，先用Ⅰ型复合线段判断，如不与障碍相交则结束判断，计算 MD 线长；如与障碍相交，则改用Ⅱ型复合线段判断。

Step3：在构造好的带边权值的完全图中，以图论中相当成熟的求源点－汇点最短路径算法求解待布点间的最短路径。本文用动态规划算法，效果很好。如果没有路径存在，则应考虑修改布局。

Step4：将求得的最短路径所经过的各顶点两两以广义线段为单位形成布线障碍结构存入障碍集，同时将本次布线所占用的绕障点和待布点从绕障点容器 Q 中删除。

Step5：将本次布线所形成的各条广义线段的绕障点经过处理，去掉不可用点后，存入绕障点容器 Q 中。

Step6：判断是否还有未布线网，如果有，转到 Step1，否则结束。

4．6．2 带权重的完全图的建立及最短路径的搜索举例

下面以图 4.16 来说明智能识点法最短路径的搜索过程：

图 4.16 表示了带权重的完全图的建立及最短路径的搜索的简单算例，图（a）为初始条件：已知存在一条已布线障碍（图中粗实线），搜索连接点 A 与 B 的最短路径。算法过程如下：

1、建立绕障点容器 Q，并将绕障点处理后放入 Q 中，Q＝{1、2、3、4、5}。如图 4.16

4．6 智能识点法最短路径搜索

4．6．1 智能识点法最短路径搜索的步骤

智能识点法最短路径搜索算法的执行步骤如下：

Step0：建立绕障点容器 Q（可用 VC 中的 Coblist 类的对象），将所有已存在的障碍物的绕障点取出，经过处理，去掉不可用的绕障点后，存入绕障点容器 Q 中。

Step1：取出欲布线网，将待布点放入绕障点容器 Q 中，以 Q 中所有点为顶点，以连接任两顶点的广义线段为边构造一个带边权值的完全图，以邻接矩阵表示，并用∞（程序中设定一个足够大的数）为每条边初始化。

Step2：判断完全图中的每条边是否与障碍相交，如与任何障碍相交，则该边权值为∞，如不与任何障碍相交，则计算该边的 MD 线长，以计算所得线长作为该边的边权值。如果该边为复合线段，先用Ⅰ型复合线段判断，如不与障碍相交则结束判断，计算 MD 线长；如与障碍相交，则改用Ⅱ型复合线段判断。

Step3：在构造好的带边权值的完全图中，以图论中相当成熟的求源点－汇点最短路径算法求解待布点间的最短路径。本文用动态规划算法，效果很好。如果没有路径存在，则应考虑修改布局。

Step4：将求得的最短路径所经过的各顶点两两以广义线段为单位形成布线障碍结构存入障碍集，同时将本次布线所占用的绕障点和待布点从绕障点容器 Q 中删除。

Step5：将本次布线所形成的各条广义线段的绕障点经过处理，去掉不可用点后，存入绕障点容器 Q 中。

Step6：判断是否还有未布线网，如果有，转到 Step1，否则结束。

4．6．2 带权重的完全图的建立及最短路径的搜索举例

下面以图 4.16 来说明智能识点法最短路径的搜索过程：

图 4.16 表示了带权重的完全图的建立及最短路径的搜索的简单算例，图（a）为初始条件：已知存在一条已布线障碍（图中粗实线），搜索连接点 A 与 B 的最短路径。算法过程如下：

1、建立绕障点容器 Q，并将绕障点处理后放入 Q 中，Q＝{1、2、3、4、5}。如图 4.16

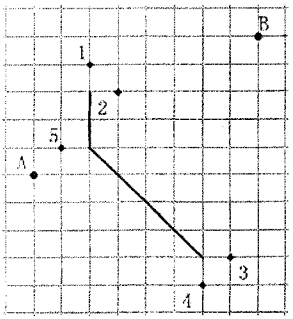
（a）所示。

2、将待布点 A，B 放入绕障点容器 Q 中，Q={A，B，1、2、3、4、5}。以 Q 中的点为顶点，以广义线段为边构造带边权值的完全图,如图 4.16（b），图中细虚线边权重为∞（无法连通），细实线边为可连通边，其权重为 MD 线长。

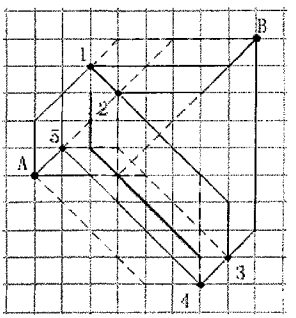
3、在所形成的完全图中用动态规划方法（见 3.2.2 节）搜索最短路径，得到图 4.16（c）中粗实线 A－1－B 的布线最短路径。

4、将本次布线所得路径以广义线段 A1 和 B1 存储到障碍集，从 Q 中删除 A, B, 1 点。

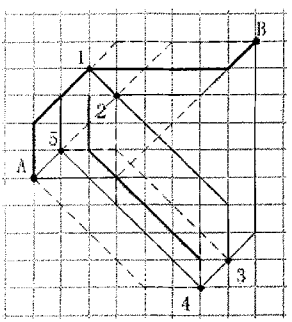
5、本次布线得到广义线段 A1 和 B1 的绕障点如图 4.16(d)中的三角形点，去掉不可用点 11，此时 Q={2，3，4，5，6，7，8，9，10，12，13，14}。



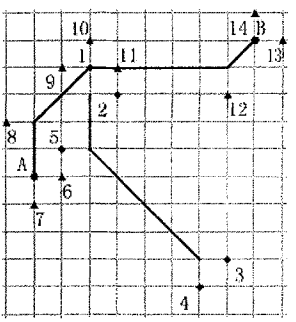
(a) 初始条件



(b)广义线段为边的完全图



(c)搜索路径



（d）绕障点的后处理

图 4.16　带权重的完全图的建立及最短路径的搜索

第五章 智能识点法自动布线算法的实现及算法性能检验

前面章节叙述了智能识点法的布线理论。在这一章中，我们将以 VC6.0＋＋为编程语言，以 WinXP 为操作系统，根据布线理论用程序实现智能识点法自动布线算法，从而检验算法性能。

5.1 智能识点法布线算法的数据结构

在程序开发过程中通常需要做到如下两点：一是高效地描述数据；二是设计一个好的算法，使算法最终可用程序来实现。因此高效的数据结构是程序开发中十分重要的一个方面。算法的数据结构是实现算法并使算法获得优良性能的关键，良好的数据结构使算法易于实现并使程序获得较快的执行速度。这一节我们将介绍智能识点法自动布线算法中各步骤实现的数据结构和实现过程。

5.1.1 智能识点法布线算法中的障碍物的数据结构

自动布线中的障碍物可以分为两种：一种为印刷电路板上固有的障碍物，如焊盘、过孔、引脚等；另一种是已布出的导线，对于串行布线，已布线是未布线的障碍。

一．已布线的数据结构

1．首先 C＋＋定义已布线障碍物的基类 CRshape：

```
class CRshape:public CObject//定义布线的线的基类结构
{
protected :
    SIZE dx;
    SIZE dy;
    int VerNo_x;//形成的连线的版本号
    CRshape();
```

第五章 智能识点法自动布线算法的实现及算法性能检验

前面章节叙述了智能识点法的布线理论。在这一章中，我们将以 VC6.0＋＋为编程语言，以 WinXP 为操作系统，根据布线理论用程序实现智能识点法自动布线算法，从而检验算法性能。

5.1 智能识点法布线算法的数据结构

在程序开发过程中通常需要做到如下两点：一是高效地描述数据；二是设计一个好的算法，使算法最终可用程序来实现。因此高效的数据结构是程序开发中十分重要的一个方面。算法的数据结构是实现算法并使算法获得优良性能的关键，良好的数据结构使算法易于实现并使程序获得较快的执行速度。这一节我们将介绍智能识点法自动布线算法中各步骤实现的数据结构和实现过程。

5.1.1 智能识点法布线算法中的障碍物的数据结构

自动布线中的障碍物可以分为两种：一种为印刷电路板上固有的障碍物，如焊盘、过孔、引脚等；另一种是已布出的导线，对于串行布线，已布线是未布线的障碍。

一．已布线的数据结构

1．首先 C＋＋定义已布线障碍物的基类 CRshape：

```
class CRshape:public CObject//定义布线的线的基类结构
{
protected :
    SIZE dx;
    SIZE dy;
    int VerNo_x;//形成的连线的版本号
    CRshape();
```

```
    DECLARE_SERIAL(CRshape)
public:
    virtual~CRshape();
    CRshape(CPoint pointSR,CPoint pointTR){}
    virtual void Get_OB_type(int &k){}
    virtual void Get_OB_Points(CPoint &pointInS,CPoint &pointInM,CPoint &pointInT){}
    virtual void Drawing_CR(CDC*pDC){}
    virtual CRshape * Get_Ponit_pp1(){return NULL;}
    virtual CRshape *   Get_Ponit_pp2(){return NULL;}
    virtual void Get_OB_circumPoints(CPoint *pp){}
    virtual void Get_VerNo(int &No_xx){}
};
```

2．定义简单线段（表 4.1）的类 CRLine：

```
class CRLine:public CRshape//定义简单线段的结构
{
protected :
    CPoint start_point_R;
    CPoint target_point_R;
    CPoint jp[5];//用以存放绕障点的数组。
    int n_type;//为 1 则为单线，为 2 则为双线
    CRLine(){}
    DECLARE_SERIAL(CRLine)
public:
    CRLine(CPoint pointSR,CPoint pointTR,int No_xx);
    virtual ~CRLine();
    void Get_OB_Points(CPoint &pointInS,CPoint &pointInM,CPoint &pointInT);
    void Get_OB_type(int &key);
    void Drawing_CR(CDC*pDC);
    virtual CRshape *   Get_Ponit_pp1(){return NULL;}
    virtual CRshape *   Get_Ponit_pp2(){return NULL;}
```

```
    virtual void Get_OB_circumPoints(CPoint *pp);
    void Get_VerNo(int &No_xx);
};
```

3．定义复合线段（表 4.1）的类 CRRLine_1（表 4.1 中Ⅰ型）和 CRRLine_2（表 4.1 中Ⅱ型）类：

```
class CRRLine_1:public CRshape//定义复合线段 1 型的结构
{
protected :
    CPoint start_point_R;
    CPoint target_point_R;
    CPoint mid; //存储复合线段的中间连接点
    CPoint jp[5]; //用以存放绕障点的数组。
    CRRLine_1(){}
    DECLARE_SERIAL(CRRLine_1)
public:
    CRRLine_1(CPoint pointSR,CPoint pointTR, int No_xx);
    virtual ~CRRLine_1();
    void Get_OB_Points(CPoint &pointInS,CPoint &pointInM,CPoint &pointInT);
    //void Get_OB_type(int &key);
    void Get_OB_circumPoints(CPoint *pp);//取绕障点
    void Drawing_CR(CDC*pDC); //显示布线结果
    void Get_VerNo(int &No_xx);
};

class CRRLine_2:public CRshape//定义复合线段 2 型的结构
{
protected :
    CPoint start_point_R;
    CPoint target_point_R;
    CPoint mid; //存储复合线段的中间连接点
```

```
    CPoint jp[5]; //用以存放绕障点的数组。
    CRRLine_2(){}
    DECLARE_SERIAL(CRRLine_2)
public:
    CRRLine_2(CPoint pointSR,CPoint pointTR, int No_xx);
    virtual ~CRRLine_2();
    void Get_OB_Points(CPoint &pointInS,CPoint &pointInM,CPoint &pointInT);
    void Get_OB_circumPoints(CPoint *pp);//取绕障点
    void Drawing_CR(CDC*pDC);//显示布线结果
    void Get_VerNo(int &No_xx);
};
```

二. 印刷电路板上固有的障碍物的数据结构

印刷电路板上固有的障碍物，如焊盘、过孔、引脚等，在电路自动布线中，我们将布线障碍物简化成几种常见的几何图形：圆形、矩形、三角形。分别定义了 Ccircle(圆形)、Crectangle(矩形)、Ctriangle(三角形)，这里只给出矩形的数据结构，其它形状的数据结构与之相类似。

```
class Crectangle:public CRshape//定义固有障碍为矩形的数据结构
{
protected :
    CPoint left_point;
    CPoint right_point;
    CPoint jp[4]; //用以存放绕障点的数组。
    int n_type;//为形状的标识
    Crectangle (){}
    DECLARE_SERIAL(Crectangle)
public:
    Crectangle (CPoint pointSR,CPoint pointTR,int No_xx);
    virtual ~ Crectangle ();
    void Get_OB_Points(CPoint &pointInS,CPoint &pointInM,CPoint &pointInT);
    void Get_OB_type(int &key);//去取标识类型成员函数
```

```
        void Drawing_CR(CDC*pDC);//显示结果
        void Get_OB_circumPoints(CPoint *pp); //取绕障点函数
        void Get_VerNo(int &No_xx);
    };
```

5.1.2 智能识点法中加权有向图的耗费邻接矩阵数据结构

由于在求解最佳布线路径时是用图论中完全图里的最短路径算法，因此需要构造带边权值的完全图的数据结构如下：

```
template<class T>
class AdjacencyWDigraph:public CObject
{
//    DECLARE_SERIAL(AdjacencyWDigraph<T>)
    private:
        T NoEdge;
        int n;      //顶点数目
        int e;      //边数
        T**a;       //二维数组(权值数组)，i 代表第 i 个点，j 代表第 j 个点
                    //a[i][j]代表权值
    public:
        AdjacencyWDigraph(int Vertices,T noEdge);
        ~AdjacencyWDigraph(){Delete2DArray(a,n+1);}
        bool Exist(int i,int j)const;
        int Edges()const{return e;}
        int Vertices()const {return n;}
        AdjacencyWDigraph<T>&Add(int i,int j,const T&w);
        AdjacencyWDigraph<T>&Delete(int i,int j);
        int OutDegree(int i)const;
        int InDegree(int i)const;
        void Make2DArray(T**&x,int rows,int cols);//二维数组创建函数
```

```
        void Drawing_CR(CDC*pDC);//显示结果
        void Get_OB_circumPoints(CPoint *pp); //取绕障点函数
        void Get_VerNo(int &No_xx);
    };
```

5.1.2 智能识点法中加权有向图的耗费邻接矩阵数据结构

由于在求解最佳布线路径时是用图论中完全图里的最短路径算法，因此需要构造带边权值的完全图的数据结构如下：

```
template<class T>
class AdjacencyWDigraph:public CObject
{
//    DECLARE_SERIAL(AdjacencyWDigraph<T>)
    private:
        T NoEdge;
        int n;      //顶点数目
        int e;      //边数
        T**a;       //二维数组(权值数组)，i 代表第 i 个点，j 代表第 j 个点
                    //a[i][j]代表权值
    public:
        AdjacencyWDigraph(int Vertices,T noEdge);
        ~AdjacencyWDigraph(){Delete2DArray(a,n+1);}
        bool Exist(int i,int j)const;
        int Edges()const{return e;}
        int Vertices()const {return n;}
        AdjacencyWDigraph<T>&Add(int i,int j,const T&w);
        AdjacencyWDigraph<T>&Delete(int i,int j);
        int OutDegree(int i)const;
        int InDegree(int i)const;
        void Make2DArray(T**&x,int rows,int cols);//二维数组创建函数
```

```
    void Delete2DArray(T**&x,int rows);//二维数组释放函数
    void Allpairs(T**c,int **kay);
};
```

5.2 智能识点法布线算法实现中的两个问题

在实现该算法中有两个关键的值得注意的问题需要解决：

1、绕障点存储前的处理；

2、完全图构造中两条广义线段相交与否的判断。

5.2.1 绕障点存储前的处理

在寻求最佳布线路径中，先要将待布点放入绕障点容器 Q（本程序用 VC 的 Coblist 类的对象 m_NG_ListRouting_Point），然后构成完全图，从而搜索到最短布线路径。在搜索到最短布线路径后，需要将本次布线和与之相对应的绕障点进行存储。但存储之前要做一些必要的处理，为下一次布线作准备。下面以图 5.1 中的情形为例说明这些处理步骤，图 5.1 中连接 AB 的折线为本次布线搜索到的布线路径。搜索到布线路径后，所需做的处理步骤包括：

从绕障点容器 Q 中：

1. 删除本次布线的待布点和本次布线占用的绕障点。图中为 A、B、6、7。
2. 删除本次布线所得路径的绕障点与 Q 中已存在的绕障点重合的点。图中为 3 和 8 重合，删除 8（或 3）。
3. 删除本次布线所得路径的绕障点与障碍重合的点。图中为 10。
4. 删除本次布线所得新绕障点之间重合的点。图中为 12 和 13，删除 13（或 12）。

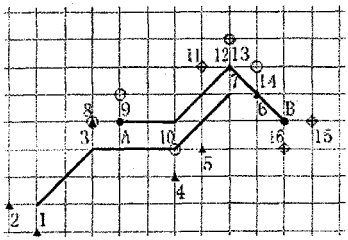


图 5.1　绕障点存储前的处理

```
    void Delete2DArray(T**&x,int rows);//二维数组释放函数
    void Allpairs(T**c,int **kay);
};
```

5.2 智能识点法布线算法实现中的两个问题

在实现该算法中有两个关键的值得注意的问题需要解决：

1、绕障点存储前的处理；

2、完全图构造中两条广义线段相交与否的判断。

5.2.1 绕障点存储前的处理

在寻求最佳布线路径中，先要将待布点放入绕障点容器 Q（本程序用 VC 的 Coblist 类的对象 m_NG_ListRouting_Point），然后构成完全图，从而搜索到最短布线路径。在搜索到最短布线路径后，需要将本次布线和与之相对应的绕障点进行存储。但存储之前要做一些必要的处理，为下一次布线作准备。下面以图 5.1 中的情形为例说明这些处理步骤，图 5.1 中连接 AB 的折线为本次布线搜索到的布线路径。搜索到布线路径后，所需做的处理步骤包括：

从绕障点容器 Q 中：

1. 删除本次布线的待布点和本次布线占用的绕障点。图中为 A、B、6、7。
2. 删除本次布线所得路径的绕障点与 Q 中已存在的绕障点重合的点。图中为 3 和 8 重合，删除 8（或 3）。
3. 删除本次布线所得路径的绕障点与障碍重合的点。图中为 10。
4. 删除本次布线所得新绕障点之间重合的点。图中为 12 和 13，删除 13（或 12）。

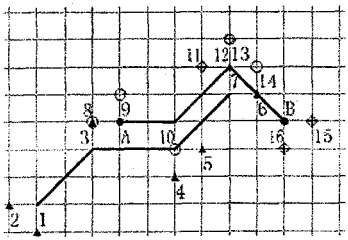


图 5.1　绕障点存储前的处理

5.2.2 两条广义线段相交的判断

在构造带边权值的完全图时，需判断完全图中的每条边（由广义线段构成）是否与障碍相交，如与任何障碍相交，表明对应的路径不通，因此所构造的完全图中对应边权值为∞，如不与任何障碍相交，则表明对应的路径可以走通，因此计算该边的 MD 线长，以计算所得线长为该边的边权值。根据 4.4.3 节的定义，广义线段分为简单线段和复合线段两种，判断两广义线段相交的子程序为：

```
bool Line_Intercourse_YorN(Circum_Point *pInCir_1,Circum_Point *pInCir_2);
```

下面是该子程序的伪码表示：

```
bool Line_Intercourse_YorN(Circum_Point *pInCir_1,Circum_Point *pInCir_2)
//判断绕障点指针 pInCir_1 和 pInCir_2 所指的绕障点是否可可以连通
begin
    if((InCir_1.x==InCir_2.x)||(InCir_1.y==InCir_2.y)||(abs(InCir_1.x-InCir_2.x)==abs(InC
    ir_1.y-InCir_2.y)))      //判断两点连线为简单线段
        for each ob∈OB do  //取出已存在的障碍集合中的每一个障碍
            if(k==1)  //判断障碍为简单线段
                Tow_PointLine_YorN_OB_Any  //调用判断两简单线段相交否子程序
                if(kk==1)      return true;  //相交，返回
            if(k==2)  ////判断障碍为复合线段
                Tow_PointLine_YorN_OB_Any  //调用判断两简单线段相交否子程序
                if(kk==1) return true;  //相交，返回
        return false;  //不相交，返回
    else  ////判断两点连线须形成复合线段
        bool YorN1=0，YorN2=0;  //连接两点的复合线段可以用两种路径，设两种情
        况是否走通的标志
        Discompose_Points(InCir_1,InCir_2,Mid_1,Mid_2);  //拆分成两简单线段
            //连接两点的复合线段可以有两条（如图 5.2），以下分别判断两复合线段
            与障碍集合中的每一个障碍相交的情况：
        //第一条复合线段的判断：
        for each ob∈OB do  //取出已存在的障碍集合中的每一个障碍
```

```
            if(k==1)  //判断障碍为简单线段
                Tow_PointLine_YorN_OB_Any  //分别用连线两点形成的每一段
                调用判断两简单线段相交否子程序
                if(kk==1) ;YorN1=1; break;    //相交则设标志 YorN1=1;
            if(k==2)  ////判断障碍为复合线段
                Tow_PointLine_YorN_OB_Any  //分别用连线两点形成的每一段
                与障碍中复合线段的两段简单线段调用判断两简单线段相交否
                子程
                if(kk==1) ;YorN1=1; break;    //相交则设标志 YorN1=1;
        //第二条复合线段的判断（与第一中情况的判断过程相同）：略
            if(kk==1) ;YorN2=1; break;    //相交则设标志 YorN2=1;
    if((YorN1==1)&&(YorN2==1)) return true;
    else return false;    //如果两条路径都无法走通，则返回 true，否则返回 false.
end
```

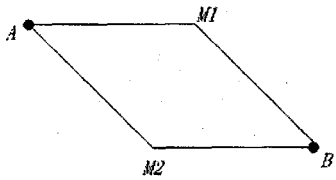


图 5.2 连接两点的两条复合线段

在计算机图形学中判断两简单线段相交的通常方法为：

1．求出两条线段所在直线的方程；

2．通过求解方程组，求出两直线的交点；

3．判断该交点是否同时在两条直线上，满足条件则判断相交，否则不相交。

用 VC6.0 编程实现，发现运用上述方法所编程序在运行时耗费大量机时，因为在上述算法中运用了较多的乘法，所以在本程序中，用上述算法判断两简单线段相交与否是不适用的。

分析发现，广义线段中的线段都是由水平线、垂直线或±45 度斜线组合而成，将广义线段进一步拆分后得到的都是水平线、垂直线或±45 度斜线，通过拆分，将判断两广义线段相交的问题转化为判断两简单线段相交的问题。充分利用各线

段位置特殊的几何特性，本文设计了判断两简单线段相交与否的算法，实现该算法的子程序为：

```
Tow_PointLine_YorN_OB_Any(int x0, int y0, int x1, int y1,
                          int x00, int y00, int x11, int y11, int &k)
```

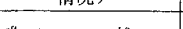
该子程序中，（x0,y0）和（x1,y1）为一条简单线段，(x00,y00)和（x11,y11）为另一条简单线段，k 是两条简单线段是否相交的标志。该子程序结构可用表格 5.1 表示：

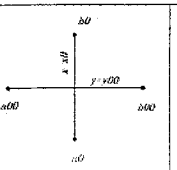
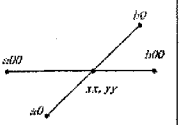
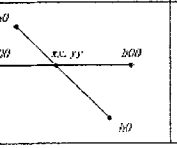
表 5.1 “判断两简单线段相交性”子程序结构表

序号	满足条件	图例	执行任务（调用相应子程序）
1	(y00==y11)&&(x00!=x11)		Tow_PointLine_YorN_OB0(x0,y0,x1,y1,x00,y00,x11,y11,k0)
2	(x00==x11)&&(y00!=y11)		Tow_PointLine_YorN_OB90(x0,y0,x1,y1,x00,y00,x11,y11,k90);
3	abs(x00-x11)==abs(y00-y11)	 或	Tow_PointLine_YorN_OB45(x0,y0,x1,y1,x00,y00,x11,y11,k45)

Tow_PointLine_YorN_OB0(x0,y0,x1,y1,x00,y00,x11,y11,k0)子程序结构可用表格 5.2 表示：

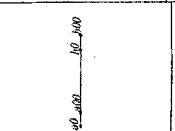
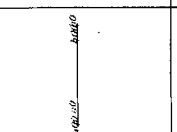
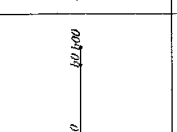
表 5.2 当一条线段为水平时相交性判断

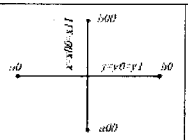
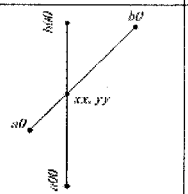
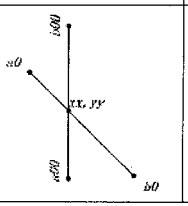
序号	前提条件（另条线的位置）	相应变量含义	满足相交的条件	图例（只列一种满足条件的通常情况）	相交性
1	(y0==y1) && (y0==y00)	a0=min(x0,x1); b0=max(x0,x1); a00=min(x00,x11); b00=max(x00,x11);	(a0<=a00)&&(a00<=b0)		相交
			(a0<=b00)&&(b00<=b0)		相交
			(a00<=a0)&&(b0<=b00)		相交

2	(x0==x1) && (y0!=y1)	a0=min(y0,y1); b0=max(y0,y1); a00=min(x00,x11); b00=max(x00,x11);	((x0>=a00)&&(x0<=b00))&&((a0<=y00)&&(b0>=y00))		相交
3	abs(y1-y0) == abs(x1-x0)	a00=min(x00,x11); b00=max(x00,x11); a0=min(x0,x1); b0=max(x0,x1);	当(y1-y0)==(x1-x0)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))，其中：xx=x0+y00-y0		相交
			当(y1-y0)==(x0-x1)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))，其中：xx=x0-y00+y0		相交

Tow_PointLine_YorN_OB90(x0,y0,x1,y1,x00,y00,x11,y11,k0)子程序结构可用表格 5.3 表示：

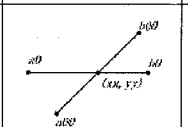
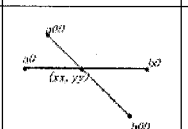
表 5.3　当一条线段为垂直时相交性判断

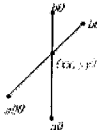
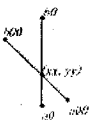
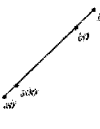
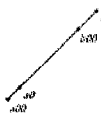
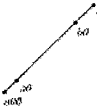
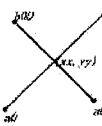
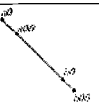
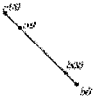
序号	前提条件	相应变量含义	满足相交的条件	图例（只列一种满足条件的通常情况）	相交性
1	(x0==x1) && (x0==x00)	a0=min(y0,y1); b0=max(y0,y1); a00=min(y00,y11); b00=max(y00,y11);	(a0<=a00)&&(a00<=b0)		相交
			(a0<=b00)&&(b00<=b0)		相交
			(a00<=a0)&&(b0<=b00)		相交

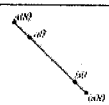
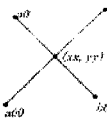
2	(y0==y1) && (x0!=x1)	a0=min(x0,x1); b0=max(x0,x1); a00=min(y00,y11); b00=max(y00,y11);	((y0>=a00)&&(y0<=b00))&&((a0<=x00)&&(b0>=x00))		相交
3	abs(y1-y0) == abs(x1-x0)	a0=min(y0,y1); b0=max(y0,y1); a00=min(y00,y11); b00=max(y00,y11);	当(y1-y0)==(x1-x0)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))，其中：xx=y0+x00-x0		相交
			当(y1-y0)==(x0-x1)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))，其中：xx=y0-x00+x0		相交

Tow_PointLine_YorN_OB45(x0,y0,x1,y1,x00,y00,x11,y11,k0) 子程序结构可用表格 5.4 表示：

表 5.4 当一条线段为±45 度时相交性判断

序号	前提条件	相应变量含义	满足前提条件下再满足相交的条件	图例（只列一种满足条件的通常情况）	相交性
1	(y0==y1) && (x0!=x1)	a0=min(x0,x1); b0=max(x0,x1); a00=min(x00,x11); b00=max(x00,x11);	当(x11-x00)==(y11-y00)时：((a0<=xx)&&(xx<=b0))&&((a00<=xx)&&(xx<=b00))，其中：xx=y0-y00+x00		相交
			当(x11-x00)==(y00-y11)时：((a0<=xx)&&(xx<=b0))&&((a00<=xx)&&(xx<=b00)) 其中：xx=x00+y00-y0		相交

2	x0==x1 && (y0!=y1)	a0=min(y0,y1); b0=max(y0,y1); a00=min(y00,y11); b00=max(y00,y11);	当(x11-x00)==(y11-y00)时：((a0<=yy)&&(yy<=b0))&&((a00<=yy)&&(yy<=b00))其中：yy=y00+x0-x00				相交
			当(x11-x00)==(y00-y11)时：((a0<=yy)&&(yy<=b0))&&((a00<=yy)&&(yy<=b00)) 其中：yy=y00-x0+x00				相交
3	abs(y1-y0)==abs(x1-x0)	a0=min(x0,x1); b0=max(x0,x1); a00=min(x00,x11); b00=max(x00,x11);	(y1-y0)==(x1-x0)	((x11-x00)==(y11-y00))&&((y0-y0)==(x0-x00))	(a0<=a00)&&(a00<=b0)		相交
					(a0<=b00)&&(b00<=b0)		相交
					(a00<=a0)&&(b0<=b00)		相交
			当(x11-x00)==(y00-y11)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))其中：xx=((x00+y00+x0-y0)/2)				相交
			(y1-y0)==(x0-x1)	((x11-x00)==(y00-y11))&&((y0-y0)==(x00-x0))	(a0<=a00)&&(a00<=b0)		相交
					(a0<=b00)&&(b00<=b0)		相交

				(a00<=a0)&&(b0<=b00)		相交
				当(x11-x00)==(y11-y00)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))其中：xx=((x00+y0+x0-y00)/2)		相交

5.3 智能识点法布线算法实验结果及算法分析

在 WinXP 操作系统下，用 VC＋＋6.0 编写运行程序，对算法的有效性和程序的执行性能进行检验。

5.3.1 算法绕障能力的检验

为检验算法的绕障能力，用鼠标任点取若干组待布点，观察布线结果并与李氏算法的绕障能力相比较，经过多次检验观察发现，本文算法和李氏算法在绕障能力方面基本相同，李氏算法能布通的线网，用本文算法也能布通。这一绕障能力强的特性从智能识点法的布线原理方面也可以得到：智能识点法自动布线算法提取了当前已布图中所有的对于绕过障碍物有用的信息——绕障点，并在绕障点和待布点组成的集合中采用广度优先搜索的思想，对最佳布线路径进行搜寻，从而得到最好的布线结果。图 5.3 是对比实验中的一组图，待连接点的数据如下：

1　700　850　　1000　1000

2　750　950　　750　750

3　750　900　　750　800

4　1000　1050　　1000　750

5　900　950　　950　850

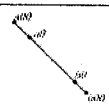
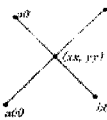
6　800　900　　900　800

7　850　1000　　900　750

8　850　1050　　900　700

9　850　1100　　900　650

对该组数据，分别用李氏算法和本文算法依数据序号的顺序布线，得到的布

				(a00<=a0)&&(b0<=b00)		相交
				当(x11-x00)==(y11-y00)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))其中：xx=((x00+y0+x0-y00)/2)		相交

5.3 智能识点法布线算法实验结果及算法分析

在 WinXP 操作系统下，用 VC＋＋6.0 编写运行程序，对算法的有效性和程序的执行性能进行检验。

5.3.1 算法绕障能力的检验

为检验算法的绕障能力，用鼠标任点取若干组待布点，观察布线结果并与李氏算法的绕障能力相比较，经过多次检验观察发现，本文算法和李氏算法在绕障能力方面基本相同，李氏算法能布通的线网，用本文算法也能布通。这一绕障能力强的特性从智能识点法的布线原理方面也可以得到：智能识点法自动布线算法提取了当前已布图中所有的对于绕过障碍物有用的信息——绕障点，并在绕障点和待布点组成的集合中采用广度优先搜索的思想，对最佳布线路径进行搜寻，从而得到最好的布线结果。图 5.3 是对比实验中的一组图，待连接点的数据如下：

1　700　850　　1000　1000

2　750　950　　750　750

3　750　900　　750　800

4　1000　1050　　1000　750

5　900　950　　950　850

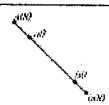
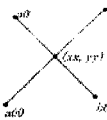
6　800　900　　900　800

7　850　1000　　900　750

8　850　1050　　900　700

9　850　1100　　900　650

对该组数据，分别用李氏算法和本文算法依数据序号的顺序布线，得到的布

				(a00<=a0)&&(b0<=b00)		相交
				当(x11-x00)==(y11-y00)时：((a00<=xx)&&(xx<=b00))&&((a0<=xx)&&(xx<=b0))其中：xx=((x00+y0+x0-y00)/2)		相交

5.3 智能识点法布线算法实验结果及算法分析

在 WinXP 操作系统下，用 VC＋＋6.0 编写运行程序，对算法的有效性和程序的执行性能进行检验。

5.3.1 算法绕障能力的检验

为检验算法的绕障能力，用鼠标任点取若干组待布点，观察布线结果并与李氏算法的绕障能力相比较，经过多次检验观察发现，本文算法和李氏算法在绕障能力方面基本相同，李氏算法能布通的线网，用本文算法也能布通。这一绕障能力强的特性从智能识点法的布线原理方面也可以得到：智能识点法自动布线算法提取了当前已布图中所有的对于绕过障碍物有用的信息——绕障点，并在绕障点和待布点组成的集合中采用广度优先搜索的思想，对最佳布线路径进行搜寻，从而得到最好的布线结果。图 5.3 是对比实验中的一组图，待连接点的数据如下：

1　700　850　　1000　1000

2　750　950　　750　750

3　750　900　　750　800

4　1000　1050　　1000　750

5　900　950　　950　850

6　800　900　　900　800

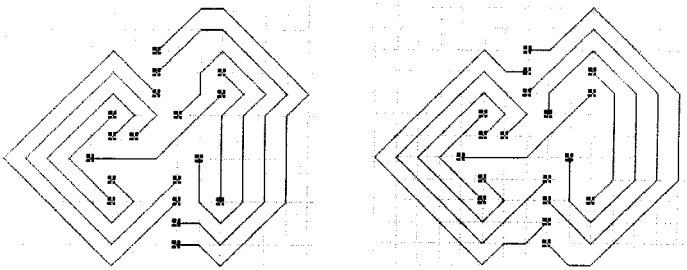
7　850　1000　　900　750

8　850　1050　　900　700

9　850　1100　　900　650

对该组数据，分别用李氏算法和本文算法依数据序号的顺序布线，得到的布

线结果如图 5.3 所示，其中 a 图为用李氏算法得到的结果，b 图是用智能识点法得到的布线结果。从图中可以看出，两种算法的布线结果基本相同，只是在不增加布线长度的前提下，有细微差别。



（a）李氏算法的布线结果　　(b)智能识点法的布线结果

图 5.3 李氏算法与智能识点法绕障能力的比较

5.3.2 布线算法综合性能测试与分析

为了测试算法的综合性能，用 50 对、70 对和 100 对待连接点分别用 MD 模式下的李氏算法和智能识点法对程序进行测试，所得程序的各项性能见表 5.5(为使数据具有更一般的代表性，各组数据是用鼠标随机点取获得)。

表 5.5 中的布通率的计算公式为：

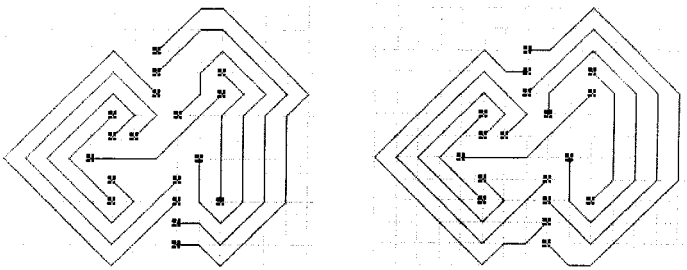
$$\frac{\text{已布通的线网数}}{\text{予布通的线网总数}} \text{X100\%}$$

表 5.5 中的数据是在布线层为单层的情况下获得的布线性能，如果分层，智能识点法的算法性能会进一步提高。表中 Lee 氏算法的网格为 3000x1800。从表中可以看出，智能识点法的绕障能力与 Lee 氏算法基本相同，用 Lee 氏算法能布通的线网，用本文算法亦能布通。当线网数较少时，智能识点法的运行速度要比 Lee 氏算法快，这是本文算法性能优越的方面。图 5.4 是表 5.5 中 70 对待布点的最后布线结果。

表 5.5 智能识点法与 MD_Lee 氏算法布线比较

对比项目 / 实验号	线网数	引脚数	布线层数	布线面积 $inch^2$	智能识点法		MD_Lee 氏算法	
					布通率%	用时 sec	布通率%	用时 sec

线结果如图 5.3 所示，其中 a 图为用李氏算法得到的结果，b 图是用智能识点法得到的布线结果。从图中可以看出，两种算法的布线结果基本相同，只是在不增加布线长度的前提下，有细微差别。



（a）李氏算法的布线结果　　(b)智能识点法的布线结果

图 5.3 李氏算法与智能识点法绕障能力的比较

5.3.2 布线算法综合性能测试与分析

为了测试算法的综合性能，用 50 对、70 对和 100 对待连接点分别用 MD 模式下的李氏算法和智能识点法对程序进行测试，所得程序的各项性能见表 5.5(为使数据具有更一般的代表性，各组数据是用鼠标随机点取获得)。

表 5.5 中的布通率的计算公式为：

$$\frac{\text{已布通的线网数}}{\text{予布通的线网总数}} \text{X100\%}$$

表 5.5 中的数据是在布线层为单层的情况下获得的布线性能，如果分层，智能识点法的算法性能会进一步提高。表中 Lee 氏算法的网格为 3000x1800。从表中可以看出，智能识点法的绕障能力与 Lee 氏算法基本相同，用 Lee 氏算法能布通的线网，用本文算法亦能布通。当线网数较少时，智能识点法的运行速度要比 Lee 氏算法快，这是本文算法性能优越的方面。图 5.4 是表 5.5 中 70 对待布点的最后布线结果。

表 5.5 智能识点法与 MD_Lee 氏算法布线比较

对比项目 / 实验号	线网数	引脚数	布线层数	布线面积 $inch^2$	智能识点法		MD_Lee 氏算法	
					布通率%	用时 sec	布通率%	用时 sec

实验 1	50	112	1	5×4	100	9	100	53
实验 2	70	140	1	5×4	100	33	100	82
实验 3	100	236	1	6×5	97	138	97	147

注：表中布线点为鼠标随机输入。

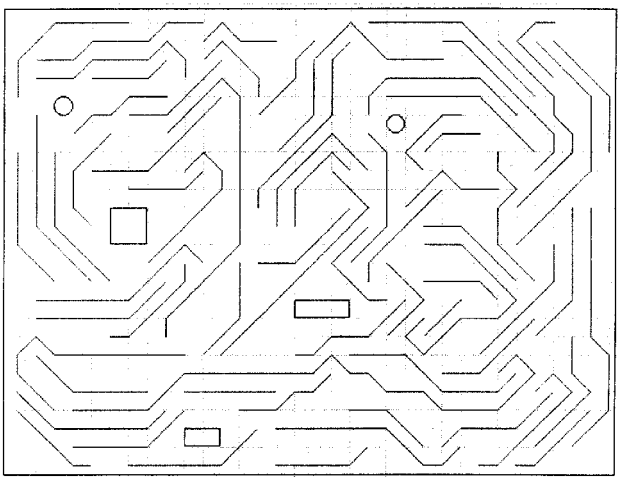


图 5.4　70 对待布点时智能识点法的自动布线结果

综合一下，我们可以得到以下智能识点法的特点：

1. 极强的绕障能力；
2. 布线速度与布线绕障点的数量有关，与布线间距无关。当线网数量不多，但布线的间距极小时，与 Lee 氏算法相比，速度的优越性很明显；
3. 算法在确定绕障点时借用了网格，但其空间复杂性与网格间距无关，因此可用于高密度、高精度布线，这点优于迷宫算法。

5.3.3 算法存在的问题以及进一步改进的方法

智能识点法是在完全图中搜索待连接点之间的最短布线路径。如果一个完全图有 m 条边，有 n 条已布线（包括固有障碍），最坏情况需要作 m×n 次两广义线

实验 1	50	112	1	5×4	100	9	100	53
实验 2	70	140	1	5×4	100	33	100	82
实验 3	100	236	1	6×5	97	138	97	147

注：表中布线点为鼠标随机输入。

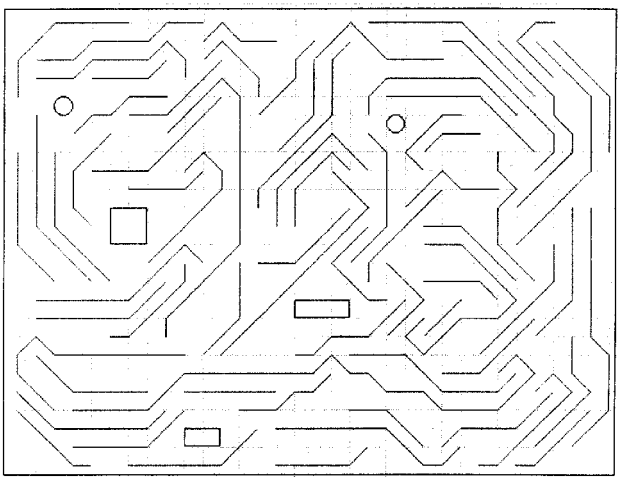


图 5.4　70 对待布点时智能识点法的自动布线结果

综合一下，我们可以得到以下智能识点法的特点：

1. 极强的绕障能力；
2. 布线速度与布线绕障点的数量有关，与布线间距无关。当线网数量不多，但布线的间距极小时，与 Lee 氏算法相比，速度的优越性很明显；
3. 算法在确定绕障点时借用了网格，但其空间复杂性与网格间距无关，因此可用于高密度、高精度布线，这点优于迷宫算法。

5.3.3 算法存在的问题以及进一步改进的方法

智能识点法是在完全图中搜索待连接点之间的最短布线路径。如果一个完全图有 m 条边，有 n 条已布线（包括固有障碍），最坏情况需要作 m×n 次两广义线

段相交与否的判断.因此随着绕障点和已布广义线段数量的增加，算法所需时间增长较快。通过实验检测得到的表 5.5 中的数据也说明了这个特点，解决的办法可以从以下几方面考虑：

1．引入深度优先搜索的思想，缩小搜索区域，减少所形成图的规模或边数；

将绕障点分为两类：一类为在待连接点对的最大包容矩形中的绕障点，这类绕障点为实现最佳路径的最有利点，在搜索路径时优先考虑。第二类为所有绕障点中，第一类以外的绕障点，当在第一类绕障点中无法搜索到最佳路径时，程序在第二类绕障点中搜索。

2．增加广义线段的类型，减少绕障点的数量；

如果增加图 5.5 中的几种线段作为广义线段，则表 4.1 中所列的简单线段只需规定两个绕障点，复合线段只需规定三个绕障点（如图 5.6 所示）则可以满足算法要求，从而大大缩短算法所需的时间。

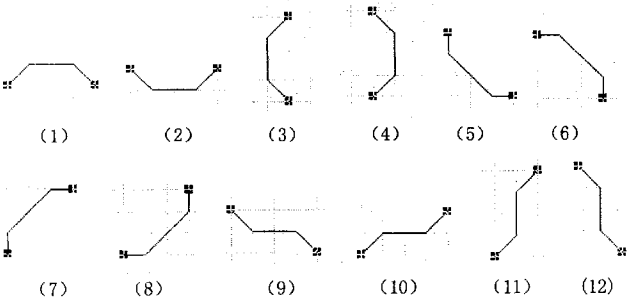
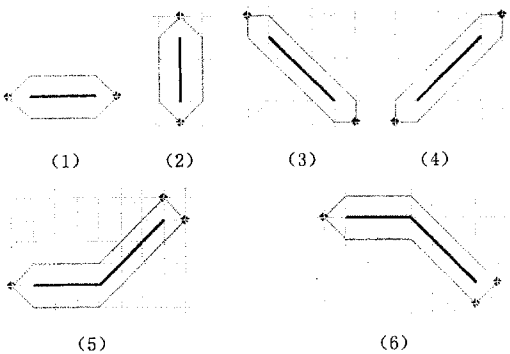
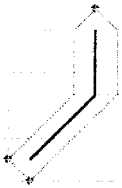


图 5.5 新增几种基本线型



(7)



(8)

图 5.6 几种基本线型的绕障点

3. 充分利用 MD 布线模型的几何特点，改进判断广义线段相交与否的方法。

第六章 结束语

CAD 技术在印刷电路板行业的应用越来越受到重视，尤其是在上个世纪八十年代以后，在 PCB 的设计、制造等方面普遍使用了 CAD 技术。

自动布线是电子 CAD 领域中的一个经典的课题，在几十年的研究发展过程中，其理论、技术和方法均取得了很大的进步。目前，各种优秀的电子 CAD 系统都包含自动布线工具，它们在 PCB 和 IC 设计中起着不可替代的作用。

然而，由于自动布线问题本身的复杂性，尽管人们为提高自动布线的速度和布通率提出了各种各样的算法，但在实际应用中，自动布线的速度、线网布通率、线网的电性能、布线的合理性等一系列问题多不能取得令人满意的效果，需要大量的人工干预。高效的自动布线算法仍然是值得人们深入研究和探讨的课题。鉴于此，本课题在较系统地分析了自动布线算法的基础上，提出了一种基于图论的布线算法——智能识点法。主要完成了以下工作：

（1）、提出了适合于 MD 布线模型上布线的广义线段的概念。在 MD 布线模型上布线，除了能布水平、垂直线外，还要求能布±45°的斜线。为了满足布线算法要求，本文提出了布线的两种基本线型——简单线段和复合线段，并将它们统称为广义线段。

（2）、针对广义线段提出了绕障点的概念及定义方法。布线算法的绕障能力是衡量算法性能的重要标志，也是实现线网自动连接的必备功能。为了使智能识点法具有较好的绕障能力，本文模仿人的绕障行为，在障碍物周围定义了绕障点，并给出了几种常见障碍物的绕障点的详细定义方法。

（3）、提出了广义线段的 MD 线长的概念。

（4）、以广义线段为边，以待布点和绕障点为顶点，并以广义线段的 MD 线长作为边的权，构造出带边权值的完全图。并用最短路径的算法在构造的完全图中搜索最佳布线路径。

（5）、为了适应自动布线速度的要求，根据 MD 布线模型的特点，设计了适用于 MD 模型上的判断两条简单线段相交性的快速算法。

由于专业知识和时间的限制，所设计的算法性能还存在有待提高之处，在

5.3.3 节，本文指出了算法具体存在的不足并提出了改进的方法，这些也是今后进一步深入研究的方向。

在整个课题的研究过程中，作者检索和查阅了大量的国内外文献资料，借鉴了前人的工作成果，同时在此基础上也提出了自己的一些思路并加以实现。总的来说，比较圆满地完成了预期的内容。本论文的所有程序均是在 Visual C++ 6.0 上调试通过，运行后达到预期效果。

在学期间发表学术论文

[1] 王洪申，廖达雄，张琦．智能识点法实现 MD 模型上的自动布线[J]．微电子学与计算机，2005 年，6 期．

[2]强会英，王洪申．椭圆拉伸变形长短轴的计算及应用[J]．电脑开发与应用，2005 年，4 期．

[3] 张琦，廖达雄，王洪申．TTF 字型在 CAD 软件设计中的应用[J]．计算机工程与应用，2005 年，11 月．

致 谢

本论文的研究工作是在导师廖达雄副教授的严格要求和热情帮助下完成的。整个课题的研究工作历时一年多，在此期间，无论是文献检索、选题论证，还是论文的进度安排和撰写，都得到了导师的指导和帮助。借此机会向廖老师的辛勤指导表示真挚的感谢！

在做论文期间，本人得到了教研室高满屯教授、李西琴教授、臧宏琦副教授的关心和帮助。此外，还得到了张琦、孙科峰、王淑侠等同学的帮助和支持，在此向他们表示深深的谢意！

在论文的研究和撰写期间，参阅了大量的文献资料，在此，向这些文献的撰写者表示最诚挚的谢意！他们的辛勤劳动给了我莫大的帮助和启迪。

尤其要感谢的是我的爱人强会英，她一个人承担起全部的家庭重担，对我是莫大的支持！

参考文献

[1] 贾新章，郝跃，武岳山编著． 电子电路 CAD 技术[M]， 西安电子科技大学出版社，1999，p3

[2] 周雅冰．电路 CAD 软件的选择及标准化[J]，电子标准与质量．1997，4：p18~21

[3] http://www.gb.tomshardware.com/howto/01q3/010815/index.html#pcb.

[4] 荒井英辅[日]编著，集成电路 A[M]，科学出版社[北京]，2000.7，p73-86。

[5] 荒井英辅[日]编著，集成电路 B[M]，科学出版社[北京]，2000.7，p81-85。

[6] C. Y. Lee. An algorithm for path connections and its applications[J]. IRE Transations on Electronic Computers，1961（9）：p346～365

[7] 庄文君，李玉兴编著，集成电路布图设计自动化，[M] 上海交通大学出版社.1986

[8] S.B.Akers，A Modification of Lee's Path Connection Algorithm[J],IEEE Trans.on Electronic Computers，1967,16(4), p97-98。

[9] J. Soukup . Fast Maze Router[J], Proc. of 15th Design Automation Conference,1978,p100-102。

[10] Hadlock ， A Shortest Path Algorithm for Grid Graphs[J] ，Networks,1977.7,p323-334。

[11] F.Rubin，The Lee Connection Algorithm[J],IEEE Trans. On Computers，1974，23，p907－914。

[12] R.K.Korn，An Efficient Variable Cost Maze Router[J]，Proc.of 19th Design Automation Conference，1982，p425-431。

[13] D.W.Hightower,A Solution to line－Routing Problems on the Continuous Plane[J],1969,D.A.workshop,p1-24.

[14] Mikami K.and Tabuchi K.,A Computer Program for Optimal Routing of Printed Circuit Connectors[J],IFIPS Proc.,1986,H47，p1475-1478.

[15] J. Soukup. Fast maze router[J]. Design Automation Conference Pro.，1978，

15：p100-102

[16] W.Heyns et.al,A Line Expansion Algorithm for the General Routing Problem with a Guran-teed Solution[J]，1980 17th D.A.conf，p243-249.

[17] Hashimoto A，Stevens J. Wire routing by optimizing channel assighnment with in large apertures[J]. Proc Desigh Automation workshop，1971，8：p115～169

[18] 杨瑞元．朝向目标的线探索法—用小型计算机实现自动布线[J]．计算机学报．1981，4：p273～282

[19] 杨瑞元．多层印制电路板的线探索布线[J]．计算机辅助设计与图形学学报．1992，4（4）：p62～67

[20] 杨瑞元．无网格线探索布线算法．计算机辅助设计与图形学学报[J]．1998，10（3）：p200～207。

[21] 熊继光，以波的传播和绕射规律为理论基础的自动布线算法系列[C]，第二届半导体集成电路计算机辅助设计会议论文，1983。

[22] J.Soukup,Global Router[J],1979,16th D.A.conf. p481-484。

[23] A. Vanneli，An Adaptation of the Interior Point Method for Solving the Global Routing Problem[J]，IEEE Trans.on CAD，1991，10（2），p193-203。

[24] M.S. Burstein and P. Pelavin，Hierarchical Wire Routing[J]， IEEE Trans.on CAD，1984，3(3)，p250-255。

[25] T.M. Parng and R.S. Tsay，A New Approach to Sea-of-Gate Global Routing[J]，Proc.of ICCAD，1989，p52-55。

[26] 谢政，戴丽，组合图论[M],国防科技大学出版社，湖南长沙，2003，p1-13

[27] 曹立明，魏兵，图论及其在计算机科学中的应用，[M],中国矿业大学出版社，1995，p3-30.

[28] Sartaj Sahni:Data Structures,Algorithms,and Applications in C++[M].Original edition copyright 1998by McGraw-Hill.Chinese edition copyright 2000 by China Machine Press. Translated by Wang Shilin,Sun Xiaodong .p202-204,p480-483[(美) Sartaj Sahni 著 数据结构算法与应用——C++ 语言描述 [M]. 汪诗林，孙晓东译，机械工业出版社 2000.1 .p202-204,p335-336]

[29] Robert Sedgewick:Algorithms in C++(Third Edition),Part 5:Graph

Algorithms[M].Simplified Chinese edition copyright 2003 by Pearson Education Asia Limited and Tsinghua University Press. Translated by Lin Qi[(美) Robert Sedgewick 著.C++算法（第 3 版）——图算法[M].林琪译. 清华大学出版社 2003.10 p266－270]

[30] 范玉妹，徐尔等，数学规划及其应用[M]，冶金工业出版社，(第二版)，2003，p297-299。

[31] 黄桐城，鲍祥霖，数学规划与对策论[M]，上海交通大学出版社 2002.7 p182-190。

[32] 洪先龙，严晓浪，乔长阁，超大规模集成电路布图理论与算法[M].北京：科学出版社，1998，p81-84，p66-p69

[33] 洪先龙，朱祺，经彤，王垠，杨 旸，蔡懿慈.非直角互连---布线技术发展的新趋势[J].半导体学报，2003，24（3）：p225-233

[34] Sarrafzadeh M，Wong C K，Hierarchical Steiner tree construction in uniform orientations[J]，IEEE Trans CAD，1992，11(9)，p1095-1106。

[35] M.A.Breuer,Design Automation of Digital Systems[J],Vol.1,Theory and Techniques,Chapter 4,5,6Printice-Hall,1972.

[36] S.Futagami,I.Shirakawa,and H.Ozaki,An automatic routing system for single-layer printed wiring board,[J],IEEE trans,Circuit Syst,vol.CAS 1982，29,p46-51.

西北工业大学

学位论文知识产权声明书

本人完全了解学校有关保护知识产权的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属于西北工业大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版。本人允许论文被查阅和借阅。学校可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律注明作者单位为西北工业大学。

保密论文待解密后适用本声明。

学位论文作者签名：王洪申　　指导教师签名：李达雄

2005 年 3 月 10 日　　2005年 3 月 10 日

西北工业大学

学位论文原创性声明

秉承学校严谨的学风和优良的科学道德，本人郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容和致谢的地方外，本论文不包含任何其他个人或集体已经公开发表或撰写过的研究成果，不包含本人或他人已申请学位或其它用途使用过的成果。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。

本人学位论文与资料若有不实，愿意承担一切相关的法律责任。

学位论文作者签名：王洪申

2005 年 3 月 10 日