



## 摘 要

电网建设中输电线路的选择过程比较复杂，需要综合考虑地理，地形，交通，施工和维护等因素对线路的影响。随着电力行业信息化的不断发展，传统的电力选线方法已不再适应信息化时代电网建设的需要，近年来利用数字化手段规划电力线路的研究已经取得了显著的成绩。

电力线路规划本质上属于航迹规划问题，在航迹规划算法中由于 A\*算法搜索过程简单，算法效率高而得到广泛的应用。本文主要讨论了基于 A\*算法的电力线路规划演示软件的设计思路；首先简要介绍了课题背景和国内外研究现状；详细论述了系统的设计原则；并粗略探讨了 A\*算法的工作原理；然后基于电力线路规划约束条件，利用稀疏 A\*算法原理设计并实现了一种适用于电力选线的路径规划算法；最后通过测试验证了算法的正确性。

本系统借鉴一般管理软件和航迹规划算法的设计原则，人机交互界面采用 QT 图形界面实现，搜索算法采用 C/C++语言实现，开发环境采用 VS2010，从而使得系统可以稳定的运行在 Windows 平台。本课题来源于华中电网横向科研项目，系统已经通过验收投入了实际使用。

**关键字：**电力选线，航迹规划，A\*算法，路径代价



## Abstract

Selection of transmission lines in power grid construction process is complicated, need comprehensive consideration of geography, topography, traffic, construction and maintenance and so on the influence of factors on the line. With the continuous development of electric industry information the traditional electric power line selection method is no longer meet the needs of the power grid construction of information age, using digital means planning electric lines of research in recent years has made significant achievements.

Electric line planning in essence belongs to the route planning problem, in the route planning algorithm due to the A \* algorithm search process is simple, high efficiency and widely used. This article mainly discussed the design idea of electric power line planning presentation software based on A\* algorithm.; first briefly introduced the topic background and research status at home and abroad; the design principle of the system are discussed in detail; and discuss the working principle of A \* algorithm; then based on power line planning constraints, a route planning algorithm for power line selection is designed and implemented by using the principle of sparse A\* algorithm; finally the correctness of the algorithm is verified by tests.

This system from general management software and the design principle of route planning algorithm, using QT graphical interface to realize the human-computer interaction interface, search algorithm is implemented using C/C++ language, development environment using VS2010, thus making the operation of the system can be stable in the Windows platform. This topic comes from the horizontal research projects of central China power grid, and has passed the actual use of the system.

**Keywords:** power line selection, route planning, A\*algorithm, route cost



## 目 录

摘 要.....	I
Abstract .....	II
1 绪 论	
1.1 课题背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 课题主要研究内容和目的.....	2
1.4 论文组织结构.....	3
2 系统需求分析	
2.1 系统概述.....	4
2.2 路径规划限制.....	4
2.3 功能性需求.....	5
2.4 数据需求.....	7
2.5 非功能性需求.....	8
3 A*算法概述	
3.1 A*算法基础.....	9
3.2 A*算法详解.....	12
4 系统概要设计	
4.1 概述.....	16
4.2 系统总体设计.....	16
4.3 系统模块设计.....	18
5 系统详细设计	
5.1 工程管理设计.....	25
5.2 约束条件设计.....	28
5.3 预处理设计.....	29
5.4 线路规划设计.....	31



---

5.5 效果演示设计.....	37
<b>6 实验结果演示</b>	
6.1 系统主界面.....	38
6.2 工程管理功能.....	38
6.3 约束条件功能.....	39
6.4 预处理功能.....	40
6.5 线路规划功能.....	40
6.6 规划结果演示.....	41
<b>7 总结与展望</b>	
7.1 论文主要内容总结.....	43
7.2 不足与展望.....	43
<b>参考文献.....</b>	<b>444</b>



## 1 绪 论

### 1.1 课题背景及意义

#### 1.1.1 课题背景

1882 年中国第一台发电机组落户上海，中国电力工业从零开始起步。虽然此后国内战乱不断，但电力工业却逐渐成长起来，到 1949 年新中国成立时全国发电装机总容量为 185 万千瓦，年发电量为 43 亿千瓦时，在当时排世界第 25 位<sup>[1]</sup>。到 1996 年底时全国发电装机总容量为 2.5 亿千瓦，发电量为 1.13 万亿千瓦时，居世界第二位。到 2014 年底，全国发电装机容量为 13.6 亿千瓦，全国总发电量 5.55 万亿千瓦时。经过近 130 多年的发展，我国电力工业从无到有，从若到强，终于迎来了持续健康发展的时期。

由于电力工业属于国民经济的基础产业，随着我国经济的不断发展，工业化和现代化进程的加快将对电力工业发展提出更高的要求<sup>[2]</sup>。电力系统是由电力生产，电力变换，电力传输，电力分配，电力消耗等子系统构成，当我国经济不断向前发展时，势必增加对电量的需求，这就使得电力系统越来越复杂，管理越来越困难。伴随着飞速发展的计算机信息技术，电力行业信息化为电力系统的现代化生产和管理带来了解决方案。电力行业信息化起源于上世纪 60 年代，即将 IT 技术运用在电力规划设计，基建，发电，输变电，供电，用电，以及企业管理经营和对外服务的各个领域<sup>[3]</sup>。近年来，随着信息技术的突飞猛进，我国电力行业信息化建设已经取得了显著的成绩，国内各大电力企业已全部实现了生产和管理的信息化。

输电线路是电力系统的重要组成部分，它将发电厂，变电站，配电设备及电力用户连接成一个有机的整体<sup>[4]</sup>。在电网建设过程中输电线路的选择极为复杂，传统线路选择是将收集到的相关地理信息制作成一定比例的地图，事先在地图上选择多条线路，然后根据实地考察对这些线路进行修改并找到最合理的一条路径。传统线路选择效率低下，误差较大，所以计算机辅助选线便应运而生。

本课题来源于华中电网横向科研项目，其中线路规划模块已经通过武汉大学验收并投入了实际使用。

#### 1.1.2 课题意义

在电网建设中合理的线路选择能够降低成本，缩短工期，降低施工难度和维护难度，并且能够提高电力传输效率和安全性。输电线路的选择通常还需要综合考虑地形，地貌，地质，交通，环境以及各地部门政策法规等多方面的因素<sup>[5]</sup>。

上节讨论到传统线路选择分为图上选线和野外选线两步，线路选择不能只考虑地形



地貌。线路相邻塔杆间的间距要合理，不能过长也不能过短；相邻塔杆间的转弯角度必须在施工和维护可以接受的范围内；线路应该尽量绕过湖泊，房屋，建筑，居民区；线路不能经过军事禁区，易燃易爆区；线路不能和铁路，公路，高速重叠，但可以交叉等等。这些因素都给线路选择人员增加了难度，不但效率低而且还极易出错，一条线路往往要实地徒步查看多次才能确定，这就需要线路选择人员具有扎实的专业功底和从业经验。

本系统针对图上选线的低效性，复杂性和易错性，把人工智能搜索算法应用到电力线路规划中，将启发式搜索和稀疏搜索结合，改进了传统搜索算法的效率，为图上选线提高了工作效率，降低了选线的复杂性，提高了线路的准确性。

## 1.2 国内外研究现状

设计一条合理的电力线路需要对电力线通过区域内的地形，环境非常了解，传统的地理勘察方法要获取比较精确的地理信息比较困难，随着现代卫星遥感技术的飞速发展，使得地理勘测人员可以获得非常精确的地理信息。遥感图像是地表的真实拷贝，具有覆盖面大，数据现势和信息直观的特点其光谱信息可以反映地质特征，在电力线路的规划中可以起到很好的辅助作用<sup>[6]</sup>。根据获取的 DEM(数字高程模型)数据和相应的地理约束条件就可以利用人工智能算法规划一条或者多条合理的路径，从而可以让电力线路选择人员轻松的获取多个线路方案。

计算机电力线路规划的基础是卫星地理信息的获取而核心则是路径规划算法，一个优秀的路径规划算法规划出的路径可以为企业节省大量的人力和资金。文献[5]讨论了基于分层模型的电力线路规划方法；文献[7]使用卫星数字地图规划出一条最优路径；文献[8]讨论了城市街区电力线路的选择方法；文献[9-11]讨论了利用栅格数据规划电力线路；文献[12]讨论了如何消除公共参与对电力线路规划的影响；文献[13]研究了人工引导选线的方法。

电力线路规划算法属于航迹规划算法中的一种，常用的航迹规划算法有 A\*算法<sup>[14]</sup>，Dijkstra 算法<sup>[15]</sup>，遗传算法<sup>[16]</sup>，蚁群算法<sup>[17]</sup>，粒子群算法<sup>[18]</sup>等。由于 A\*算法可以找到最优路径并且易于实现，故而应用范围非常广泛，比如在无人机航迹规划，机器人航迹规划，以及游戏中都有大量的应用。A\*算法属于启发搜索算法，算法依据搜索过程中结点的代价选择最优的结点从而得到最优的路径。本文针对传统 A\*算法搜索过程中搜索结点过得多的缺点，采用稀疏 A\*算法减少了搜索中间结点，优化了算法规划效率。

## 1.3 课题主要研究内容和目的

### 1.3.1 主要研究内容

课题主要通过深入研究国内外路径规划算法，根据 DEM 数据和规划约束条件，对



电力线路规划算法进行建模，用 C/C++ 语言在 Windows 平台下实现算法并测试其正确性。具体研究内容有：

- 1) DEM 数据的转换。由于原始 DEM 地形数据是文本文件，不方便规划算法处理也不方便显示路径规划结果，所以需将 DEM 数据读取到程序的地图数据结构中保存，并将 DEM 地图数据转换为一张可视化的灰度图。
- 2) 规划算法的设计。由于地图可能比较大，所以如果使用传统的 A\* 规划算法就会非常耗时，所以需要对地图数据进行采样即稀疏地图，然后在稀疏地图上规划一条最优的路径。
- 3) A\* 搜索算法代价函数的设计。上节提到 A\* 规划算法的代价函数很重要，这直接决定了算法的优劣，所以需根据电力线路规划的各个约束条件对代价函数进行设计。

### 1.3.2 研究目的

由于传统电力线路规划的种种复杂性，课题主要的研究目的是实现电力线路规划的数字化，信息化，选线的简单化和准确化，并根据国内外研究现状重点研究 A\* 算法在电力线路规划中的优化策略和代价函数设计方法。

具体来说有以下几个研究目的：

- 1) 研究原始 DEM 数据的转换为系统内部数据的方法，完成地图的可视化工作。
- 2) 研究基本 A\* 算法的工作原理和搜索过程中路径节点的数据结构设计。
- 3) 研究大地图下稀疏搜索的方法，并将其与 A\* 算法有机高效的结合在一起。
- 4) 研究大地图下在有限 RAM 中地图和约束区域的数据结构设计和存储方法。

## 1.4 论文组织结构

本文共有六章其结构如下：

- 1) 第一章，绪论。本章对课题的主要研究内容进行了概括性的描述。
- 2) 第二章，系统需求分析。本章讨论了电力线路规划演示软件的各种功能性和非功能性需求。
- 3) 第三章，A\* 算法概述。本章详细讨论的 A\* 算法的基础原理，并用伪代码描述了算法流程，最后通过一个例子说明的算法的执行过程。
- 4) 第四章，系统概要设计。本章详细论述了电力线路规划演示软件的总体架构以及各个子模块的设计原则。
- 5) 第五章，系统详细设计。本章详细介绍了系统各个模块的具体实现方法，并深入讨论了线路规划算法的设计。
- 6) 第六章，实验结果演示。本章对电力规划演示软件的各个功能模块进行了演示，并测试了在不同约束区域中算法的正确性。
- 7) 第七章，总结与展望。本章回顾了论文探讨的主要问题，并总结了系统的不足之处。





## 2 系统需求分析

### 2.1 系统概述

本系统在用户输入的 DEM 地形数据上按照地理环境和各种选线约束规划出一条最优路径并验证算法的正确性，为方便规划将各种规划约束条件整合到一张组合地图中，系统需要完成的主要任务有：

- 1) 规划环境地形数据的预处理。将用户输入的数据格式转换为程序的内部各式并可视化为一幅地图，建立规划威胁区和规划禁止区，并将这些信息写到规划用的组合地图中。电力线路可以通过但是代价较大的区域为威胁区，电力线路绝对不能通过的区域为禁止区。
- 2) 规划约束区域的指定。为方便验证算法的正确性，规划约束区域人为指定。约定威胁区为圆形，禁止区域为矩形，实际情况中约束区域大多为不规则的形状，但是可以将这些形状简化为不规则的多变形。
- 3) 实现规划算法。根据电力线路规划的特殊性和特殊要求在现有国内外路径规划研究的基础上确定一种适合电力线路规划的算法并改进算法，使算法在满足指定线路起点，终点和规划约束的条件下，以尽量少的时间规划出一条最优路径。
- 4) 坐标转换。由于 DEM 数据给出的坐标是地球经纬度，不适合直接在系统中使用，所以必须将其转换系统方便处理的计算机屏幕坐标。
- 5) 演示规划结果。当规划算法规划结束后将规划结果输出到文件保存，系统再从文件中将规划结果读取并显示在地图上以便选线人员分析。

### 2.2 路径规划限制

电力线路规划需要考虑很多因素，这些因素都对最终的规划结果有影响，在设计系统时必须充分考虑。主要因素有：

- 1) 电力线路禁止进入沼泽地，水草地，盐碱地，炸药厂，鞭炮厂，采石场，采矿区，飞机场，军事禁区以及江河湖海等地区。
- 2) 电力线路应该尽量避免进入村庄，规划区，零星房屋，成片森林等地区。
- 3) 电力线路不能和道路，规划道路，高速公路，铁路，油气管线，输电线路重叠，但是可以交叉。
- 4) 线路在高度上不能超过指定的爬升角度和下滑角度。
- 5) 线路的转弯角度不能超过指定的转弯角度。
- 6) 线路各个塔杆之间有最小距离与最大距离的约束。塔杆之间的距离不得小于最小距离也不得大于最大距离。





综上所述我们可以将规划限制总结为两类：地理限制和线路选择限制。地理限制是指某些地区不能通过，视为禁止区，某些地区应该避免通过但是可以通过，视为威胁区。线路选择限制是指对线路施工的要求，比如某个点的转弯角度不能过大，线路的爬升角度和下滑角度不能过大否则施工和维护难度将会增加或者根本不能施工。

## 2.3 功能性需求

本系统根据电力线路规划的特点和实际情况，以及规划环境，路径规划条件的限制，在数字高程地图的基础上对地图数据进行预处理，生成系统可以直接使用的地形海拔数据和组合地图数据，然后自动进行规划并对规划结果进行演示和观察。整个系统的结构流程如图 2.1 所示：

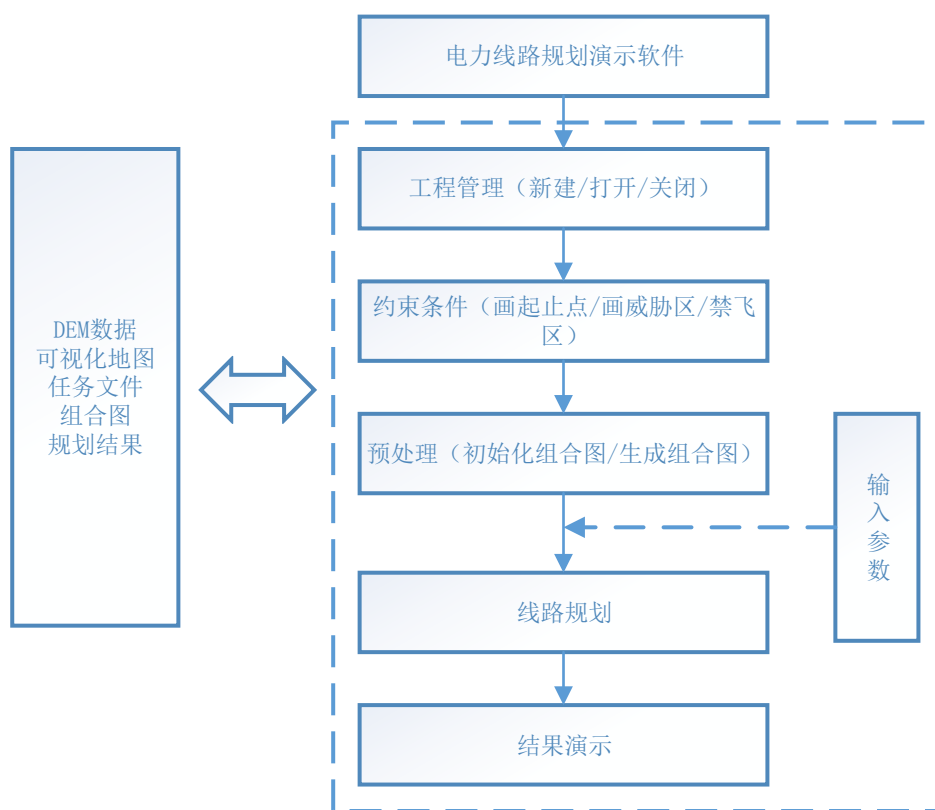


图 2.1 系统整体结构流程图

本系统各个模块按照如图 2.1 所示的流程分工合作最终规划出一条合理的路径。工程管理模块负责打开原始数字高程图并将其可视化；约束条件模块负责指定规划的起止点，禁止区和威胁区；预处理模块负责将禁止区和威胁区的信息写到一张组合地图中并生成规划任务文件；线路规划模块负责根据任务文件和组合地图规划出一条合理路径并输出规划结果；结果演示模块将规划结果显示在可视化地图上。

各个模块的具体功能如下所述：



### 2.3.1 工程管理模块

当系统完成启动后必须新建或者打开一个原有的工程文件才能开始一次新的规划，本模块是后续所有模块的基础模块，负责将用户输入的 DEM 数据转换为系统可以统一处理的格式化文件，然后将 DEM 数据可视化。本模块由打开工程，新建工程，关闭工程等三个子模块构成。

- 1) 新建工程文件。读取原始 DEM 地图数据并将其转换为系统方便处理的格式化文件（后续简称为高程图）。
- 2) 打开工程文件。读取新建工程文件时生成的高程图文件，将其转换为可视化的灰度图（BMP 图）显示在系统主界面上。
- 3) 关闭工程文件。关闭整个工程，将系统恢复到刚刚启动时的状态。

### 2.3.2 约束条件模块

约束条件模块主要是为后续规划算法指定规划起止点，规划区域中的禁止区域，威胁区域。为了验证线路规划算法的正确性，稳定性，在算法的设计、实现、测试过程中必须考虑到各种可能出现的问题。如果使用真实的地理约束区域有以下两点劣势：

- 1) 测试用的数据收集比较繁琐，收集到的数据量有限不能代表大部分的实际情况。
- 2) 收集到的数据是固定的，不能测试数据极限的情况，更不能产生复杂多变的约束区域。

综上两点本系统决定采用人工指定约束区域的方式产生测试数据测试算法的正确性。人工指定约束区域有以下几点优势：

- 1) 产生测试数据简单，而且可以随机产生大量的测试数据。
- 2) 可以模拟很多复杂的约束区域以验证线路规划算法是否能在这些复杂的环境中规划一条正确的路径。

### 2.3.3 预处理模块

预处理模块负责生成最终规划用的组合地图，分为初始化组合图和生成组合图两个部分。初始化组合图模块从高程图中读取地图的有关信息，然后将地图中的所有点初始化为未限制区域。生成组合图模块将系统记录的各个规划约束区域的信息写入到组合图中，并生成任务文件。任务文件只包含规划的起止点坐标和海拔信息。

### 2.3.4 线路规划模块

线路规划模块是整个系统的核心模块，由于规划区域可能会特别大，所以本模块采取稀疏 A\*算法行规划。算法根据任务文件，高程图，组合地图规划一条最优路径，规划完成后将规划路径坐标信息输出到文件保存。

### 2.3.5 结果演示模块

规划结果是一系列的点的集合，各个点表示线路的塔杆位置。结果演示模块负责将



线路规划模块的规划结果显示在可视化地图上，可方便观察规划结果是否合理。

### 2.3.6 参数输入模块

由于在实际电力线路的选择过程中存在诸多约束，比如两个相邻塔杆之间的距离选择，两个相邻塔杆之间的坡度选择，相邻塔杆之间转弯角度的选择都会影响线路走向，所以系统在规划电力线路时必须将这些因素考虑在内。系统应在规划之前设置相邻塔杆之间的这些约束参数，这些参数将会影响算法的搜索行为和搜索效率并影响最终的规划线路。

## 2.4 数据需求

本系统数据采取自定义文件的方式统一管理。原始 DEM 数据，高程图，组合图，任务文件，规划结果文件都放在指定的文件夹中。DEM 数据放在 dataorg 目录下，高程图放在 datamap 目录下，组合图放在 datazht 目录下，任务文件放在 Task 目录下，规划结果放在 LocalPlan 目录下。高程图和组合图是二进制文件，任务文件和规划结果是文本文件。系统数据流程图如图 2.2 所示：

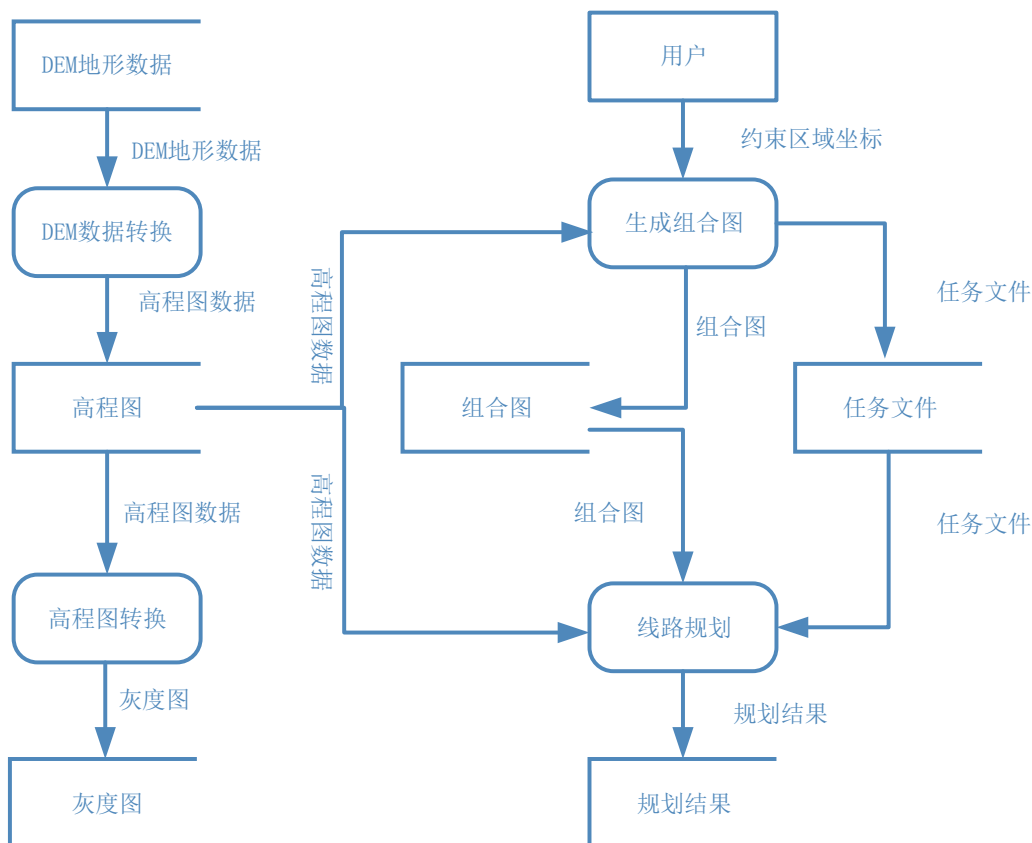


图 2.2 系统数据流程图

系统的新建工程模块以 DEM 地形数据为输入以高程图数据为输出；打开工程模块以高程图为输入以可视化的灰度图为输出；当用户指定了起止点与约束区域后，预处理



模块以高程图和约束区域坐标信息为输入以组合图和任务文件为输出；线路规划模块以高程图、组合图和任务文件为输入以规划结果为输出。

## 2.5 非功能性需求

系统各种非功能性需求如表 2.1 所示：

表 2.1 系统软硬件需求

设备项	最低要求	推荐值
CPU	2.0GHz	2.5GHz 以上
内存	2.0GB	4.0GB 以上
硬盘	50.0GB	100.0GB 以上
显卡	显存 512.0MB	显存 1.0GB 以上
操作系统	WinXP	Win7
开发环境	VS2010	VS2010+QT4



## 3 A\*算法概述

### 3.1 A\*算法基础

#### 3.1.1 A\*算法简介

A\*算法属于人工智能算法,人工智能算法主要是以问题特定的知识为基础求解其最优解的算法,其搜索策略有盲目式搜索和启发式搜索两种。盲目式搜索并不利用实际问题提供的信息计算搜索结点的状态,而是根据当前结点不断的产生后继结点,直到后继结点是目标点为止。盲目式搜索并没有“实际问题实际分析”的能力,算法只能根据事先确定的步骤寻找目标点,这样算法的效率就只能单纯的依赖问题的规模。启发式搜索可以克服盲目搜索的不足之处,利用问题特定的知识制定搜索策略,待求解的问题不同相应的搜索策略不同。启发式搜索在每次搜索过程中都会计算搜索结点的状态,利用状态信息求解问题的最优解,并能有效减小问题的求解规模,提高算法的效率。

A\*算法通过综合源点经过搜索路径到当前点的实际代价,以及从该点到目标点的预估代价来计算点的状态信息。A\*算法在搜索过程中会将所有搜到的中间结点保存起来,所以算法效率仍然和源点到目标点的深度密切相关。本章将会分析 A\*算法的各个特性,并讲述 A\*算法的搜索过程,最后通过一个例子演示了其搜索状态的变化。

#### 3.1.2 图论简介

##### ➤ 图的定义

图 $G = (V, E)$ 是一个二元组,其中  $V$  表示图中顶点的有穷集合,  $E$  表示图中边的有穷集合<sup>[19]</sup>。集合  $V$  中的元素称为顶点或结点,集合  $E$  中的元素称为边。图中每个顶点都有一个邻接点集合,例如对于顶点  $u$  的邻接点集合可表示为  $Adj[u]$ ,它是所有与顶点  $u$  有边相连的顶点集合。如下图所示,顶点集合  $V = \{A, B, C, D, E\}$ ,边集合  $E = \{a, b, c, d, e, f, g\}$ ,为了叙述的准确性通常不将边表示为  $a, b \dots$ ,例如边  $a$  可以表示为边  $(A, B)$ ,  $b$  可以表示为边  $(B, C)$ ,顶点  $A$  的邻接点集合为  $Adj[A] = \{B, E\}$ ,顶点  $B$  的邻接点集合为  $Adj[B] = \{A, E, D, C\}$ 。

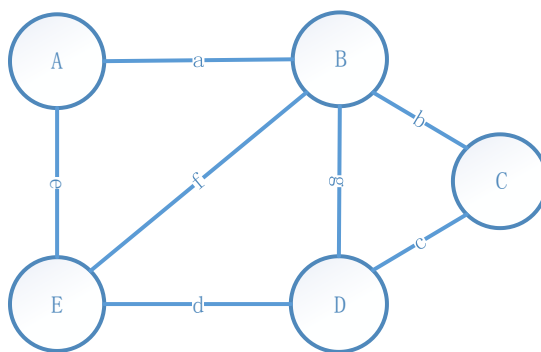


图 3.1 例图



图通常包含有向图和无向图两种，由于本课题讨论的 A\*算法只使用无向图所以有向图不再赘述。无向图的性质有：

- 1) 边集合  $E$  是由顶点的无序对构成。例如边  $(A,B)$  和边  $(B,A)$  是指同一条边  $a$ ，即边的顶点是无序的。
- 2) 图中不允许存在自环。每条边的两个顶点必须是不同的顶点，例如边  $(A,A)$  是不存在的。

#### ➤ 权重图

权重图是指图中的每条边都有一个权重。边的权重通常是一个边集到实数集的函数  $\omega: E \rightarrow R$ 。例如图  $G = (V, E)$  是一个权重图，设其权重函数为  $\omega$ ，则边  $(u, v) \in E$  的权重可以表示为  $\omega(u, v)$ 。在不混淆的情况下通常可以将边的权重称为边的代价。

#### ➤ 路径及其权重

路径是图中的一个顶点序列。设图  $G = (V, E)$  中顶点  $u$  到  $u'$  的路径为  $p(u, u') = \langle v_0, v_1, v_2, \dots, v_k \rangle$ ，其中  $u = v_0$ ， $u' = v_k$ ，且有  $(v_{i-1}, v_i) \in E$ ， $i=1, 2, \dots, k$ 。

路径权重是指构成该路径的所有边的权重之和。在图  $G = (V, E)$  中结点  $u$  到结点  $u'$  的路径  $p$  的权重为：

$$\omega(p) = \sum_{i=1}^k \omega(v_{i-1}, v_i) \quad (3.1)$$

由上述定义可以得出该路径中包含了顶点  $v_0, v_1, \dots, v_k$  以及边  $(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$ ，路径总长度为  $k$  即路径中边的数目。如果从顶点  $u$  到  $u'$  有一条路径  $p(u, u')$ ，则称顶点  $u$  到  $u'$  是可达的。

前面探讨了路径及其权重，下面引入最短路径及其权重。如果图  $G = (V, E)$  中结点  $u$  到结点  $u'$  可达，则其最短路径的权重  $\delta(u, u')$  为：

$$\delta(u, u') = \min\{\omega(p): u, \overset{p}{\rightsquigarrow} u'\} \quad (3.2)$$

故而从结点  $u$  到结点  $u'$  的最短路径为：任何从结点  $u$  到结点  $u'$  的路径中路径权重  $\omega(p) = \delta(u, u')$  的路径。从此处可以看出最短路径并非一条。

#### ➤ 网格地图

本系统搜索算法使用的是网格地图，网格地图是一个矩阵。地图中的单元格可以抽象为一个质点，每个质点只具有坐标属性，即每个质点由平面坐标和高度三个量唯一表示一个质点。

### 3.1.3 A\*算法原理

#### ➤ 估价函数模型

前面讨论过 A\*算法是启发式搜索算法，就是运用与需求解问题相关的启发性知识，在搜索过程中对搜索到的点进行搜索代价评估，得到一个最优的位置，然后从这个最优位置继续向下搜索，直到搜到目标点为止。在启发式搜索中位置估价函数非常重要，不





同的估价函数算法的工作效率不同。在 A\*算法中位置估价函数为：

$$f(n) = g(n) + h(n) \quad (3.3)$$

$f(n)$ 表示顶点  $n$  的总估计代价， $g(n)$ 表示从源点  $s$  到顶点  $n$  的实际路径代价，启发函数 $h(n)$ 表示从顶点  $n$  到目标点  $t$  的启发代价<sup>[20]</sup>。文献[21-23]讲述了估价函数模型并分析了启发函数的特性。

### ➤ 估价函数性质

由于 A\*算法是可采纳的，可设源点  $s$  到目标点  $t$  是可达的，且 $p'(n_i, n_j) \subseteq p(s, t)$ ，路径 $p(s, t)$ 为源点  $s$  到目标点  $t$  的一条最短路径。若有函数 $h(n)$ 满足 $h(n_i) - h(n_j) \leq c(n_i, n_j)$ 且 $h(n) \leq h'(n)$ ， $c(n_i, n_j)$ 为从顶点 $n_i$ 到 $n_j$ 的实际代价， $h'(n)$ 为顶点  $n$  到目标点的实际代价，则有 $f(n_i) \leq f(n_j)$ ，此时称启发函数 $h(n)$ 是相容的。

证明：

$$\because h(n_i) - h(n_j) \leq c(n_i, n_j)$$

$$\because h(n_i) - h(n_j) + g(n_i) \leq c(n_i, n_j) + g(n_i)$$

$$\text{又} \because f(n_i) = g(n_i) + h(n_i)$$

$$\therefore f(n_i) - h(n_j) \leq c(n_i, n_j) + g(n_i)$$

$$\therefore f(n_i) \leq c(n_i, n_j) + g(n_i) + h(n_j)$$

$$\text{又} \because g(n_j) = c(n_i, n_j) + g(n_i)$$

$$\therefore f(n_i) \leq g(n_j) + h(n_j)$$

$$\text{又} \because f(n_j) = g(n_j) + h(n_j)$$

$$\therefore f(n_i) \leq f(n_j)$$

综上所述，如果从源顶点  $s$  到目标顶点  $t$  是可达的且启发函数 $h(n)$ 相容，则估价函数 $f(n)$ 单调递增。则当 A\*函数在扩展结点时估价函数的值不减少，那么最先找到的路径的代价一定是最小的（路径是最优的），而且从理论上说已访问过的结点以后都不需要再访问。如果搜索算法一定能找到最优路径则称启发函数 $h(n)$ 是可采纳的。

### ➤ 估价函数设计

估价函数的设计并不是一成不变的，也没有固定的规律可循，通常是实际问题实际分析。并不是所有的启发函数都是相容的，但启发函数一定是可采纳的，否则算法就不



能找到最优路径。

设顶点  $n$  的坐标为  $(x_n, y_n)$ ，目标点  $t$  的坐标为  $(x_t, y_t)$ ，搜索地图是网格地图而不是数学意义上的图，一般来说启发式函数有如下几种：

- 1) 曼哈顿距离。平面直角坐标系中线段在两个坐标轴上投影长度的总和，即线段在  $X$  轴的分量加上线段在  $Y$  轴的分量之和。

$$h(n) = |x_t - x_n| + |y_t - y_n| \quad (3.4)$$

- 2) 欧几里得距离。平面直角坐标系中两点间的直线距离。

$$h(n) = \sqrt{(x_t - x_n)^2 + (y_t - y_n)^2} \quad (3.5)$$

- 3) 契比雪夫距离。平面直角坐标系中线段在两个坐标轴上投影长度最大者。

$$h(n) = \max(|x_t - x_n|, |y_t - y_n|) \quad (3.6)$$

上述启发函数都是固定的简单的距离计算，不利于算法动态调整搜索方向，可以给启发函数  $h(n)$  乘以一个动态权值来改善算法的效率和搜索方向。设动态权值函数为  $K(n)$  则改进后位置估价函数为：

$$f(n) = g(n) + K(n) \times h(n) \quad (3.7)$$

$K(n)$  的值可以通过分析待解决的问题加上实验确定，大多数情况下  $K(n)$  是一个常量值， $K(n)$  用来改善  $f(n)$  中  $g(n)$  与  $h(n)$  所占的比重，从而改善算法的搜索效率。

## 3.2 A\*算法详解

### 3.2.1 A\*算法基本操作

在介绍 A\* 算法前需要说明 A\* 算法使用的松弛操作，松弛操作即对某点的扩展。对图  $G = (V, E)$  中的每个顶点  $u$  都有三个与其相关的属性  $u.f$ ,  $u.g$ ,  $u.h$  以及  $u.p$ ，分别为点  $u$  的估计代价，源点  $s$  到  $u$  的实际代价， $u$  到目标点  $t$  的启发代价以及顶点  $u$  的前驱结点。可以使用下面的算法对图中的点进行初始化：

**INITIALIZE-SOURCE( $G, s, t$ )**

1. for each vertex  $u \in G.V$
2.  $u.f = \infty$
3.  $u.g = \infty$
4.  $u.h = h(u)$
5.  $u.p = NIL$
6.  $s.g = 0$
7.  $s.f = s.g + s.h$

在经过上述初始化操作后源点  $s$  的路径估计代价为源点的启发代价且小于无穷大，其他点的路径估计代价和实际代价为无穷大，用启发式函数  $h(n)$  算出所有点的启发代价值，



所有点的前驱结点都为空。

对一条边 $(u,v)$ 的松弛操作包含如下步骤：

- 1) 测试源点  $s$  到点  $v$  的实际代价是否可以减小。将  $u.g$  加上边 $(u,v)$ 的权重 $\omega(u,v)$ ，然后与  $v.g$  比较，如果前者比后者小则转到下一步。
- 2) 修改  $v.g$  与  $v.f$  的值。将  $v.g$  修改为  $u.g$  加上边 $(u,v)$ 的权重 $\omega(u,v)$ ， $v.f$  修改为  $v.g$  加上  $v.h$ ，将点  $v$  的前驱结点  $v.p$  修改为顶点  $u$ 。转下一步。
- 3) 修改集合  $Q$ 。如果顶点  $v$  不在  $Q$  中则将顶点  $v$  加入到  $Q$  中。

松弛造成伪代码如下所示：

**RELAX-EDGE**( $u, v, \omega, Q$ )

1. if  $v.g > u.g + \omega(u, v)$
2.      $v.g = u.g + \omega(u, v)$
3.      $v.f = v.g + v.h$
4.      $v.p = u$
5.     if  $v \notin Q$
6.          $Q = Q \cup \{v\}$

当算法成功找到一条最优路径时需要将这条路径打印出来，伪代码 **PRINT-PATH** 可以打印从源点  $s$  到  $t$  的路径中的所有结点。算法在搜索过程中会保存其前驱结点的信息，当成功找到目标点后，可以根据其前驱结点的信息回溯到源点，从而打印其路径。对 **PRINT-PATH** 伪代码稍作修改便可将源点到目标点的路径信息保存到一个向量中。

**PRINT-PATH**( $G, s, t$ )

1. if  $t == s$
2.     print  $s$
3. else if  $t.p == NIL$
4.     print "error"  $t$  "no parent"
5. else **PRINT-PATH**( $G, s, t.p$ )
6.     print  $t$

### 3.2.2 A\*算法基本流程

A\*算法通过启发函数评估搜索过程中点的代价，通过比较选择最优的点进行扩展，直到搜索到目标点为止。A\*算法在运行过程中维护两个顶点集合 **CLOSE** 表和 **OPEN** 表。**OPEN** 表存放其估计代价已经计算但是尚未被扩展的点，**CLOSE** 表存放已经被扩展的点。算法重复从 **OPEN** 表中选择路径估计代价最小的顶点  $u$ ，并将点  $u$  存入 **CLOSE** 表中，然后对与顶点  $u$  相连的边进行松弛操作。在下面的算法伪代码中 **OPEN** 表使用最小优先队列实现，其关键值为点的路径估计代价  $f$ ，**CLOSE** 表使用一般性容器即可。算法伪代码如下所示：



### **A-STAR-SEARCH( $G, s, t, \omega$ )**

1. **INITIALIZE-SOURCE( $G, s, t$ )**
2. **OPEN** = { $s$ }
3. **CLOSE** =  $\emptyset$
4. **while** **OPEN**  $\neq \emptyset$
5.      $u = \text{POP-MIN}(\text{OPEN})$
6.     **CLOSE** = **CLOSE**  $\cup \{u\}$
7.     **if**  $u == t$
8.         **PRINT-PATH( $G, s, t$ )**
9.         **break**
10.    **for each vertex**  $v \in G.\text{Adj}[u]$  **and**  $v \notin \text{CLOSE}$
11.         **RELAX-EDGE( $u, v, \omega, \text{OPEN}$ )**

上述伪代码中过程 **POP-MIN** 是从 **OPEN** 表中弹出路径估计代价  $f$  最小的点  $u$ 。上述伪代码第 1 行对图中所有点进行初始化，第 2 行将 **OPEN** 表初始化为源点  $s$ ，第 3 行将 **CLOSE** 表初始化为空。第 5~6 行弹出 **OPEN** 表中路径估计代价最优的结点  $u$ ，并将其并入 **CLOSE** 表。第 7~9 行判断点  $u$  是否是目标点  $t$ ，如果是则打印源点  $s$  到目标点  $t$  的路径并退出循环。第 10~11 行对所有与  $u$  相邻的点  $v$  进行松弛操作。第 4 行中判断 **OPEN** 表是否为空，如果 **OPEN** 表为空还未找到合理的路径说明源点  $s$  到目标点  $t$  不可达。

为了更清晰的说明算法的执行流程，图 3.2 所示的例子详细展示了算法执行时的中间状态。本例中采用网格地图，图中深蓝色的方块表示障碍区域，障碍区域不能通过，算法将从方块  $s$  寻找到一条到  $t$  的最优路径。图中每个方格中有三个数字一个箭头。方格中左上角的数字表示路径估计值即  $f$  值，左下角的数字表示实际代价值即  $g$  值，右下角的数字表示启发值即  $h$  值，右上角的箭头表示寻路方向，箭头末端指向其前驱结点，黑色方块表示已经放入 **CLOSE** 表的点，浅色方块表示 **OPEN** 表中的点。(a)中表示的是算法第一次从 **OPEN** 表中取得源点  $s$  并扩展后的场景。(b)~(e)为每次成功执行 **while** 循环后的场景，由于中间场景比较多只取了其中的几个具有代表性的场景。(f)为最后一次寻找到目标点  $t$  时的场景。当寻找到目标点  $t$  后根据图中的箭头反向搜索就可以找到这条最短路径。

需要说明的是，为了简单明了的展示算法执行过程中各个变量的变化情况，图中每向相邻方块移动一步其代价为 1，只能水平或者垂直移动。图中启发函数  $h(n)$  使用的是曼哈顿距离。从图中(d)可以看出某一时刻路径估计代价最优的点有多个，这时最优点的选择并没有特定的法则，可以选择为最近找到的最优点或者选择最先找到的最优点也可，当最优点有多个选择时选择任何一个都不会影响最终的最优路径，算法能在所有最优路径中选择其中的一条，从图中(f)也可以看出只是选择了多条合理路径中的一条。

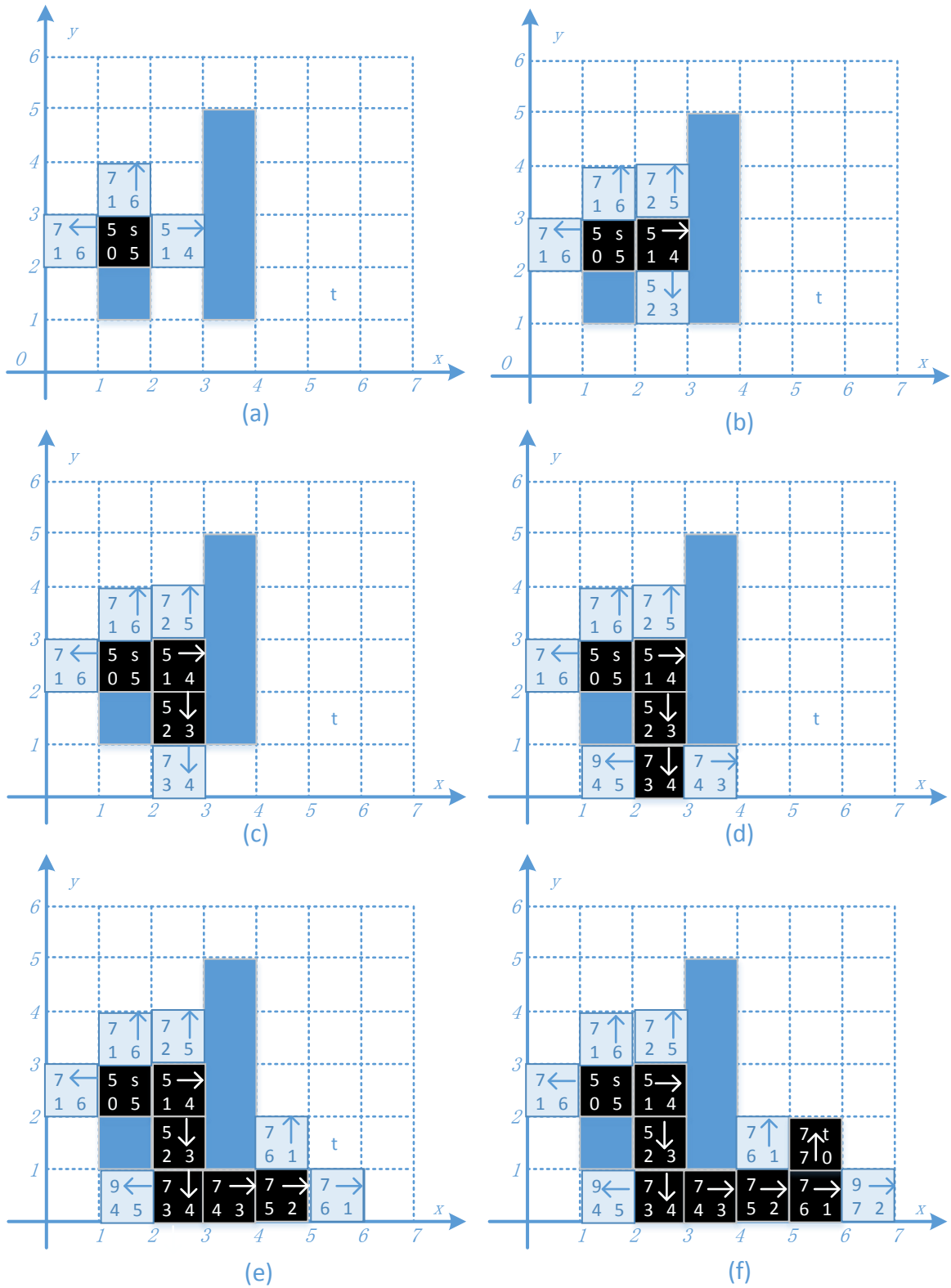


图 3.2 A\*算法执行过程



## 4 系统概要设计

### 4.1 概述

本课题的任务是完成电力线路规划演示软件的设计和实现工作，同时在现有航迹规划算法的基础上研究并改进一种适用于电力线路选择的航迹规划算法。本系统的主要设计目的是综合电力线路选择的各种要求，结合计算机技术为传统电力线路选择提供服务。由于电力线路的选择分为纸上选择和实际勘测两步，本系统主要是用信息化手段替代传统纸上选线，为实际勘测提供精确的数据资料。系统设计的主要任务是：

- 1) 界面简洁，操作简单。电力线路规划系统在人机交互方面应当直观，灵活，便捷，易用，符合用户使用习惯。
- 2) 系统各个模块间高内聚，低耦合，功能完善。系统各个模块高效完成其对应的功能并与其它模块合作在用户的操作下规划出一条合理的路径，各个模块间互不影响。

### 4.2 系统总体设计

#### 4.2.1 系统模块结构

本系统能根据电力线路规划的目标点，起始点，路径规划限制，规划环境约束等要求自动规划出路径结果，同时能对规划结果进行显示并将规划结果按照指定文件格式输出。系统按照工程的方式组织每次规划中的文件，即一次规划任务产生不同的规划相关文件。系统的总体模块结构如图 4.1 所示：

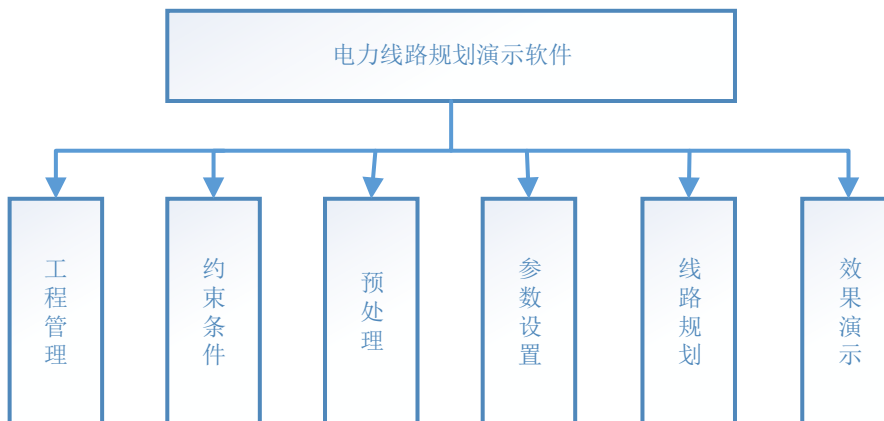


图 4.1 总体模块结构

本系统皆以菜单的方式实现各个功能模块，这些功能模块是：

- 1) 工程管理。本模块包含的子模块有新建工程文件，打开工程文件，关闭工程，关闭系统。
- 2) 约束条件。本模块包含画起始点，画终止点，画威胁区，画禁止区四个子模块。





- 3) 线路规划。本模块包含参数设置和线路规划两个子模块。
- 4) 预处理。本模块包含初始化组合图，生成组合图两个子模块。
- 5) 效果演示。本模块没有子模块。

#### 4.2.2 系统使用流程

本系统各个模块分工合作，其使用流程如下图所示：

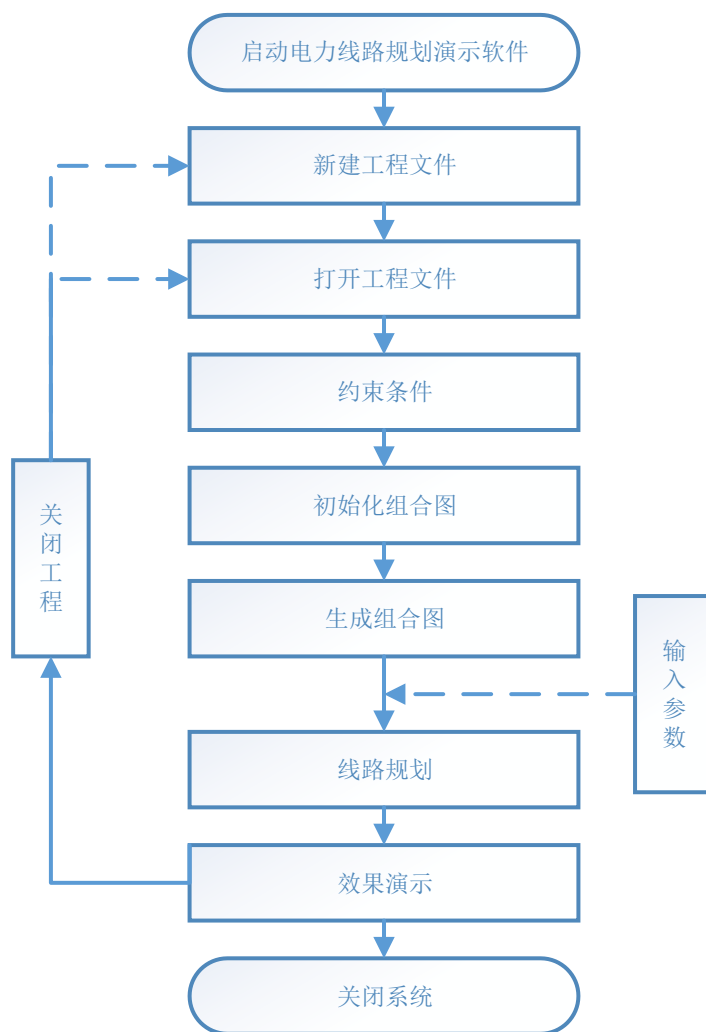


图 4.2 系统使用流程图

如图 4.2 所示，当第一次启动系统时，工程管理模块中新建工程文件子模块将读取 DEM 文件并生成高程图；打开工程文件子模块读取高程图文件以便转将其转换为灰度图；约束条件模块在系统的可视化界面上指定起止点和约束区域；生成组合图模块读取高程图的信息并综合约束信息生成一张组合图以及任务文件；线路规划模块则读取任务规划文件并依据高程图和组合图中的信息进行规划并输出规划结果；效果演示模块则读取规划结果并将其显示在可视化地图上。系统的各个模块相互独立，又分工协作，任何一个模块的任务未完成都不能进入到下一个模块。



### 4.2.3 系统总体界面

为了实现友好的人机交互操作，提高可视化效果，本系统参照一般软件的设计思路将系统主体管理界面设计为下拉菜单的方式。界面的构成部件有：系统标题区，主功能菜单区，功能快捷按钮区，地图浏览区，消息显示区。界面设计如下图所示：



图 4.3 系统总体界面

系统采用下拉菜单方式管理各个模块，启动时整个界面充满屏幕。系统标题区显示系统的名称：电力线路规划演示软件；主功能菜单区的每个菜单将各个模块综合在一起，分为：工程管理菜单，约束条件菜单，线路规划菜单，效果演示菜单，预处理菜单。快捷按钮区是各个菜单中子菜单的工具栏按钮；地图浏览区是用来显示可视化地图，用户指定的起止点，约束区以及最终规划结果的区域；消息显示区并未包含在主界面之内，消息显示区是一个单独的模态对话框，只有当系统有消息显示时才会出现，消息显示区有两种，第一种为常规信息显示对话框，第二种是时间计数对话框，下小节将会对这两种对话框详细描述。

## 4.3 系统模块设计

### 4.3.1 工程管理菜单

一般系统的工程管理菜单都会生成工程管理文件，这个文件记录了相应工程的各种信息。当关闭工程后再次打开工程时，工程管理菜单会从工程管理文件中读取记录的信息，然后自动加载用户的设置，不需要用户管理工程用到的各种文件，从而改善用户体验。由于本系统文件不多且流程简单，所以本系统工程管理菜单并不具备真正的工程管理能力，本模块相当于一个数据处理模块，负责把 DEM 数据转换为高程图数据，然后将高程图数据转换为可视化灰度图。本菜单由新建工程文件，打开工程文件，关闭工程，退出系统等子菜单构成。工程管理菜单界面设计如下所示：

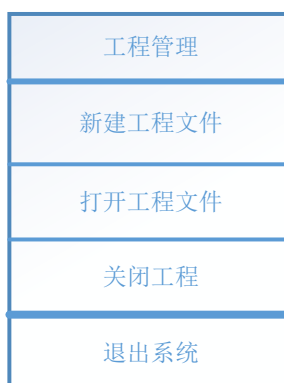


图 4.4 工程管理菜单界面

工程管理菜单结构图如下所示：

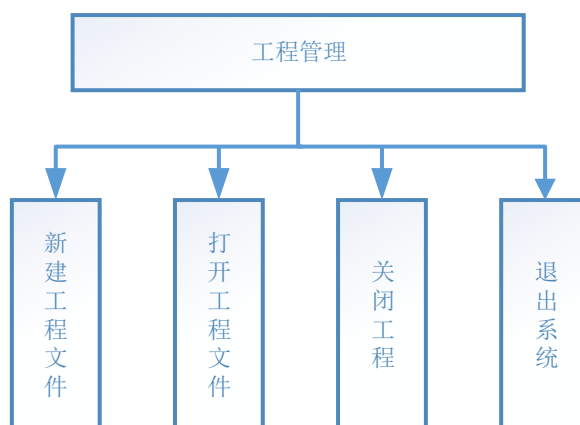


图 4.5 工程管理结构图

工程管理菜单数据流程图如下图所示：



图 4.6 工程管理数据流程图

如图 4.6，工程管理菜单中各个子菜单的输入输出如下所述：

- 1) 新建工程文件子菜单输入为 DEM 地形数据，输出为高程图数据。
- 2) 打开工程文件子菜单输入为高程图数据，输出为可视化地图数据（灰度图）。
- 3) 关闭工程子菜单无确切输入输出数据，关闭当前打开的工程文件，释放当前工程占用的资源。
- 4) 关闭系统菜单关闭整个系统，清理系在运行期间所申请的任何资源。

#### 4.3.2 约束条件菜单

约束条件菜单用于指定线路规划算法的起止点，规划约束区域。为了方便后续组合

图的生成将禁止区设计为矩形，威胁区设计为圆形。本菜单有画起始点，画终止点，画威胁区，画禁止区四个子菜单组成。

约束条件菜单界面设计如下：

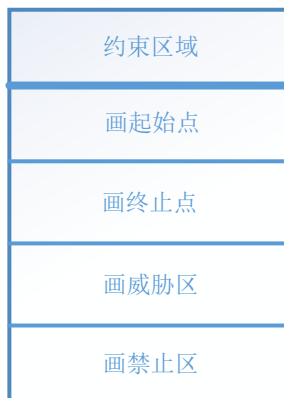


图 4.7 约束条件菜单界面

约束条件菜单结构如下图所示：

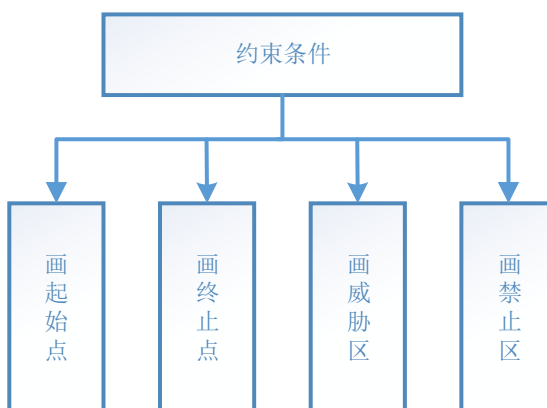


图 4.8 约束条件菜单结构

约束条件菜单数据流程图如下图所示：

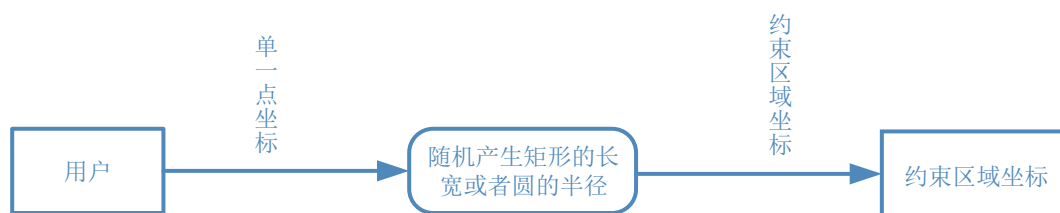


图 4.9 约束条件数据流程图

当用户选中约束条件中某个子菜单后，可在地图上单击一个点，系统记录下这个点的坐标，并依据这个坐标生成相应的约束区域坐标，具体入下：

- 1) 当用户选择画起止点或是终止点时，用户单击点的坐标就为相应的起止点坐标。
- 2) 当用户选择画威胁区时，用户单击点的坐标为圆心，系统随机产生圆的半径。
- 3) 当用户选择画禁止区时，用户单击点的坐标为矩形左上角坐标，系统随机产生矩形的长宽。



### 4.3.3 预处理菜单

预处理菜单用于生成规划时使用的组合图数据，所谓组合图就是标明规划地图内每个点是否在威胁区、禁止区内或者点不在任何规划约束区内。当线路规划算法要判断某个点是否可以扩展时就需要用到组合图中的信息。预处理菜单包含初始化组合图和生成组合图两个子菜单。

预处理菜单界面设计如下：

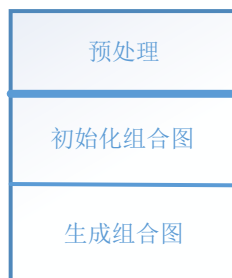


图 4.10 预处理菜单界面

预处理菜单结构如下所示：

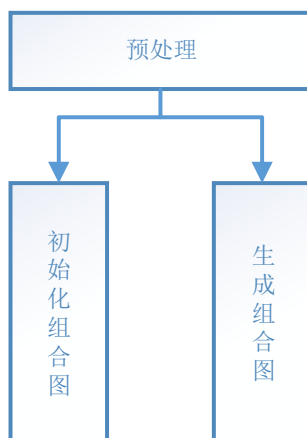


图 4.11 预处理菜单结构

预处理菜单数据流程图如下所示：

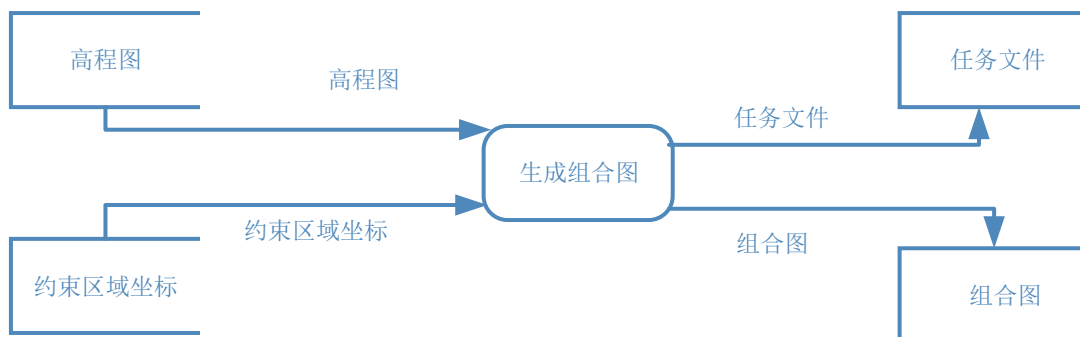


图 4.12 预处理数据流程图

如图 4.12，预处理菜单中各个子菜单的输入与输出描述如下：

- 1) 初始化组合图。输入为一个组合图文件名和高程图数据，输出为根据高程图数据初



始化的组合图。

- 2) 生成组合图。输入为初始化的组合图和约束区域坐标，输出为包含约束区域信息的组合图以及规划任务文件。

#### 4.3.4 线路规划菜单

线路规划菜单根据指定的起止点，约束区域及其他规划要求规划出一条合理的线路。线路规划菜单包含输入参数和 A\*线路规划两个子菜单。

线路规划菜单界面设计如下所示：

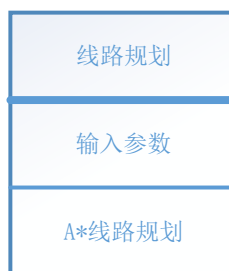


图 4.13 线路规划菜单界面

输入参数界面设计如下：

两点间最小距离	<input type="text"/>
两点间最大距离	<input type="text"/>
两点间最大角度	<input type="text"/>
两点间最大坡度	<input type="text"/>
<div><input type="button" value="确定"/> <input type="button" value="取消"/></div>	

图 4.14 输入参数界面

线路规划菜单结构如下所示：

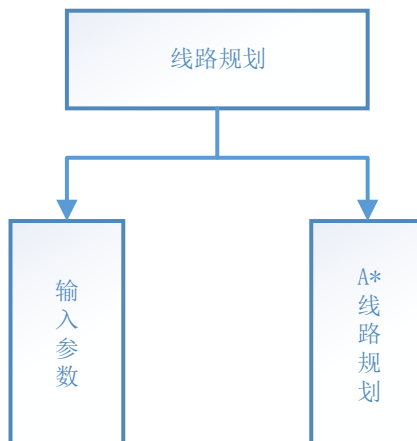


图 4.15 线路规划结构





线路规划菜单数据流程图如下所示：

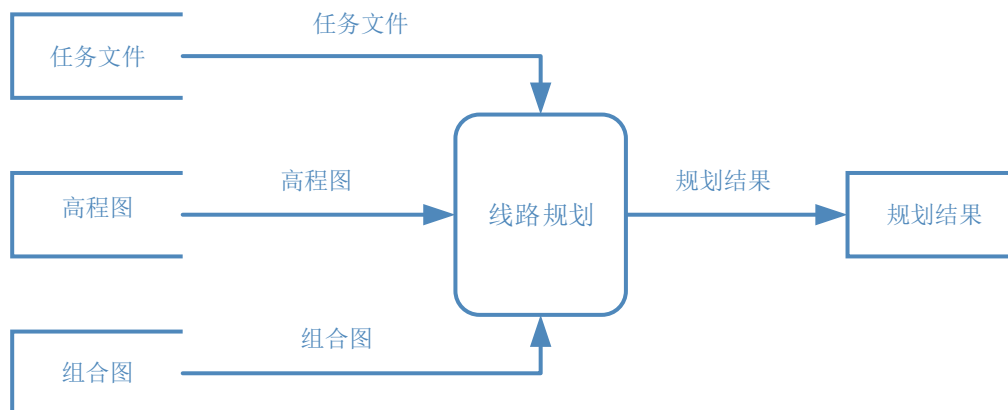


图 4.16 线路规划数据流程图

如图 4.16，线路规划菜单中各个子菜单输入与输出描述如下：

- 1) 输入参数子菜单输入的参数包含相邻塔杆间的距离、爬升和下滑坡度以及转弯角度，当选择输入参数子菜单时会弹出一个参数输入对话框，用户输入的这些参数会被保存在系统相应的变量中，作为 A\*线路规划算法的参数被算法使用。
- 2) A\*线路规划子菜单输入为任务文件，高程图和组合图，输出为规划结果，规划结果是一系列点的集合，这些点构成了一条电力线路。

#### 4.3.5 效果演示菜单

线路规划完成后会把规划的线路保存成固定格式的文本文件，效果演示就是对规划结果的二维动态演示，即把结果中的各个点连线显示在可视化地图上。效果演示菜单只有效果演示一个子菜单。

效果演示菜单界面设计如下：



图 4.17 效果演示菜单界面

效果演示菜单结构图如下所示：

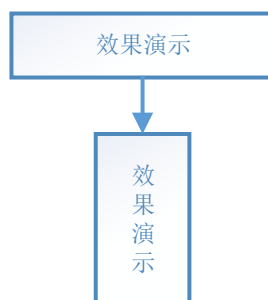


图 4.18 效果演示结构图

效果演示数据流程图如下所示：

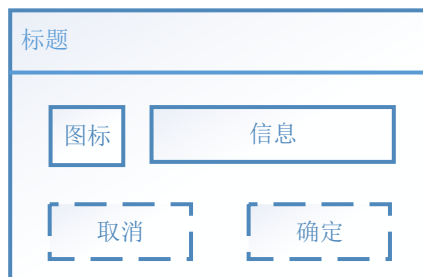


图 4.19 效果演示数据流程图

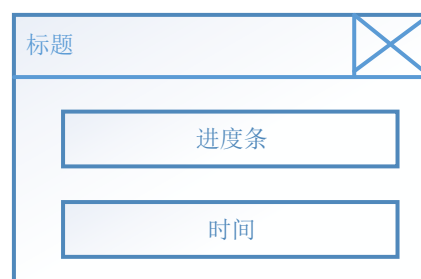
如图 4.19，效果演示输入文件为线路规划结果，输出为可视化的规划结果，规划结果与起止点，约束区域一起显示在可视化地图上。

#### 4.3.6 消息对话框

信息显示对话框在每个模块中都有可能出现，主要用于提示系统出现的错误，向用户显示某项操作是否成功，显示某个操作所消耗的时间或是在执行某种操作前取得用户的选择信息。两种消息显示对话框如下图所示：



常规信息对话框



时间对话框

图 4.20 信息显示对话框

如图 4.20，常规信息显示对话框的取消和确定按钮是可选的，二者中可以只出现一个，时间对话框的右上角设计有取消按钮。常规信息显示对话框是用于向用户显示一般的信息，比如某个模块是否执行成功或者询问用户是否执行某项操作。时间对话框用于显示某项操作的执行进度，比如线路规划模块非常耗时于是向用户显示当前规划已经用了多少时间，每个时间对话框都关联了一个耗时的操作，用户可以单击取消按钮取消这项耗时的操作。



## 5 系统详细设计

### 5.1 工程管理设计

#### 5.1.1 模块说明

本模块是系统的基础模块，后续各项操作均在本模块基础之上。本模块执行过程中主要是以对文件的读、写、保存为主要操作，读取原始 DEM 地形数据进行转换，并将转换后的数据保存在高程图文件中。本模块由新建工程文件，打开工程文件，关闭工程等模块构成。

#### 5.2.2 新建工程文件

新建工程文件子模块负责完成 DEM 地形数据到高程图数据的转换，因为原始的 DEM 地形文件并不能满足系统规划的需要，所以为了便于规划系统使用了统一的内部文件格式(\*.map)。

##### ➤ DEM 地形图数据格式

DEM 地形图数据文件是文本文件，分为两部分：文件头，文件数据。

- 1) 文件头。本部分只有一个整数  $n$ ，这个整数表示整个地图有  $n$  个点。
- 2) 文件数据。本部分由  $n$  行组成，每行由 4 个浮点数组成，前两个数据为点的经纬度，第三个数据为点的高度，第四个数据为高度的误差。

##### ➤ 高程图文件格式

转换的高程图(\*.map)的格式也包含文件头信息，文件数据两部分。

- 1) 文件头信息。本部分包含了地图左上角的经纬度坐标，数据的行数，数据的列数，最低海拔，最高海拔，精度以及无效数据。
- 2) 文件数据。本部分包含的是地形的海拔数据，文件被设计为一个网格模型，每个单元格存储地形中一个点的海拔，数据长度为 16 位。每个点的坐标加上文件头信息中左上角的经纬度坐标就是点的原始经纬度坐标。

##### ➤ 程序算法

- 1) 程序通过弹出一个文件选择对话框获取需要转换的 DEM 地形文件路径字符串，然后判断获取的字符串是否为空，如果为空则退出。
- 2) 弹出一个文件选择对话框获取转换后的高程图文件路径字符串，然后判断获取的字符串是否为空，如果为空则退出。
- 3) 将上述两个文件路径字符串传递给数据转换线程。
- 4) 数据转换线程首先打开 DEM 文件读取数据，然后按照高程图文件的格式将数据写入到高程图文件中。

程序逻辑流程如下所示：

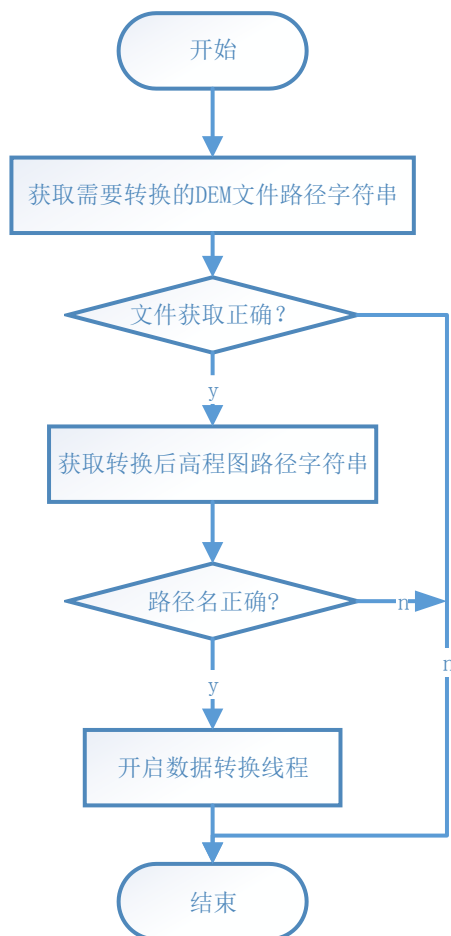


图 5.1 新建工程文件逻辑

### 5.2.3 打开工程文件

打开工程文件子模块对系统自定义的高程图文件实现打开浏览功能，即可视化。输入为高程图文件，输出为 RGB 图或灰度图。

#### ➤ 灰度图格式

灰度图是一个像素矩阵，其行数和列数与高程图中的行列数相等，矩阵中的每个像素对应高程图中的每个点的海拔。海拔数据到 RGB 数据的转换公式为：

$$RGB = ((elevation - minElevation) \div (maxElevation - minElevation)) \times 255 \quad (5.1)$$

从公式(5.1)中可知每个像素点 RGB 值的三个分量都为同一个值，即点的海拔和地形中最小海拔的差值与地形中最小海拔和最大海拔的差值的比例乘以 255。RGB 值为 0 代表此点的海拔与最低海拔相同，255 代表此点的海拔与最高海拔相同，在灰度图中显示出来则为越白的区域海拔越高，越黑的区域海拔越低。

#### ➤ 程序算法

- 1) 打开需要浏览的高程图文件。
- 2) 读取高程图文件的文件头和文件数据。
- 3) 申请用于保存灰度图数据的存储空间。



- 4) 按照公式(5.1)和灰度图的格式将高程图数据转换为灰度图存储在申请的空间中。
- 5) 将转换后的数据交由界面窗口部件显示。

程序逻辑流程如下所示：

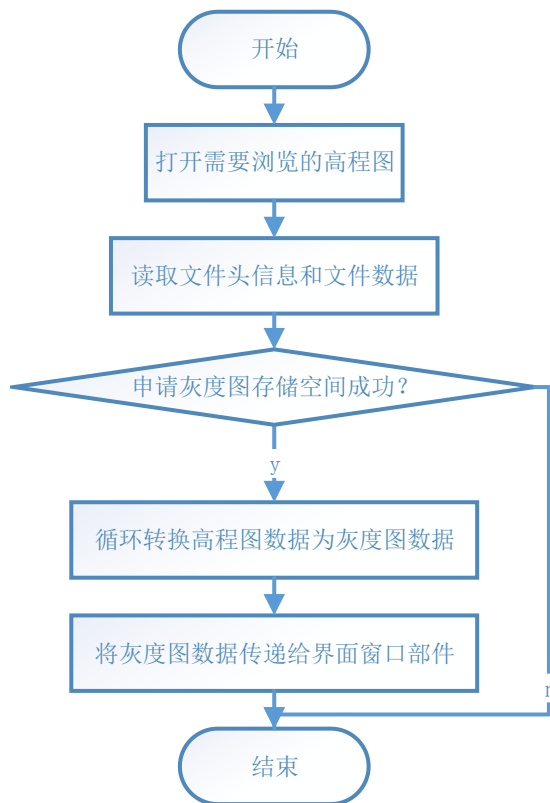


图 5.2 打开工程文件逻辑

#### 5.2.4 关闭工程

关闭工程子模块关闭当前正在使用的工程，设置系统界面部件的状态为初始状态，释放工程打开期间所申请的资源，关闭工程打开期间打卡的文件。程序逻辑流程如下：

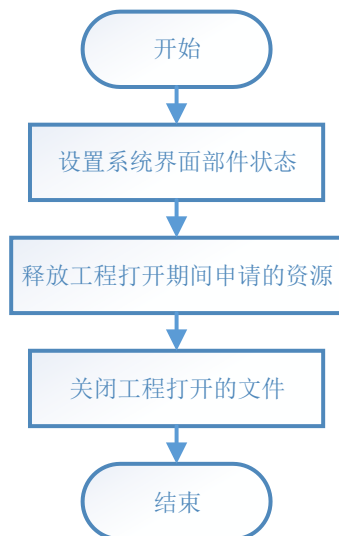


图 5.3 关闭工程逻辑

## 5.2 约束条件设计

### 5.2.1 模块说明

为了方便系统处理和模拟真实环境中的障碍区域，约束区域有两种：威胁区，禁止区。威胁区电力线路可以通过但是每个威胁区都有威胁等级，不同的等级线路通过的代价不一样。禁止区则禁止线路通过。本模块由画起止点，画威胁区，画禁飞区组成等子模块组成。

为了区分威胁区和禁止区，将禁止区设置为长度和宽度在 10 至 50 像素的矩形，威胁区设置为半径在 20 到 50 像素的圆形。由于用户随机设置约束区域所以每个约束区域有可能和前一个区域重合，如果威胁区与威胁区或者禁止区域与禁止区域重合则重合部分的约束性质不变。如果威胁区和禁止区重合则重合区域的约束性质以禁止区为主，即线路不能通过重合区域。约束区形状域如下图所示：

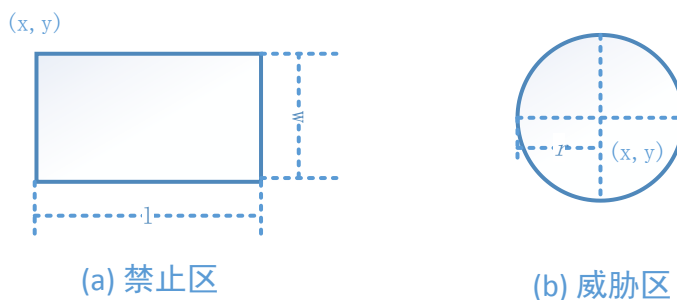


图 5.4 威胁区域

如图 5.4 所示，(a)为禁止区， $(x,y)$ 为左上角坐标， $l$  为长， $w$  为宽；(b)为威胁区， $(x,y)$ 为圆心坐标， $r$  为圆的半径。

### 5.2.2 程序算法

- 1) 用户约束条件菜单中的任意子菜单，程序保存用户的选择。
- 2) 当用户在系统主界面上单击鼠标时：
  - a) 若用户选择的是画起始点或者画终止点，系统在主界面上以用户点击点为中心画一个等边三角形，并保存单击点的坐标。
  - b) 若用户选择的是画威胁区，系统在主界面上以用户单击点为圆心，以一个大于 10 小于 40 的随机数为半径画一个圆，并将这个圆的坐标保存到威胁区链表中。
  - c) 若用户选择的是画禁禁止区，系统在主界面上以用户单击点为左上角，以两个大于 10 小于 50 的随机数为长和宽画一个矩形，并将这个矩形的坐标保存到禁止区链表中。

### 5.2.3 模块流程

本模块的流程图如下所示：



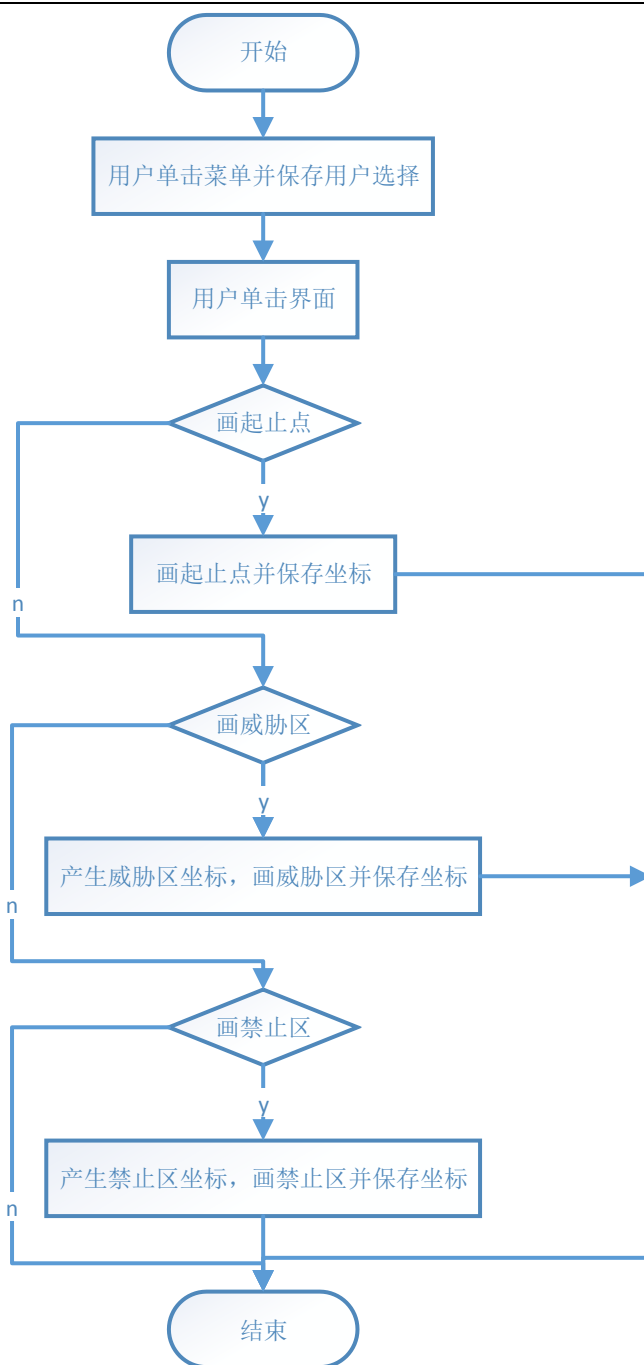


图 5.5 约束区域逻辑

## 5.3 预处理设计

### 5.3.1 模块说明

在规划过程中需要不断的从高程图中读取数据，不断的判断某个点是否在威胁区或者禁止区中。例如当算法扩展某个点时需要判断这个点的高度是否合法，需要判断这个点是否在某个约束区域中。通常我们可以用一个循环判断某个点是否在禁止区或者威胁区中，算法的时间复杂度为 $\theta(n)$ ， $n$ 为约束区域的个数。当威胁区和禁止区的数目非

常大，而判断的点非常多时将严重影响算法的性能。故而设计组合图将某个点是否在约束区内的详细信息予以记录，当要判断某个点时只需根据点的坐标便可在组合图中获得点的详细信息，时间复杂度为 $\theta(1)$ 。组合图以文件的形式保存，并不常驻内存，当需要取得组合图中的信息时，系统用 Windows 内存映射的方法取得组合图中的相应数据。

### 5.3.2 组合图编码

组合图用 32 位编码高程图中每个点的约束信息。低 16 位用于表示该点是自由规划区，禁止区，威胁区。高 16 位用于表示某个点在威胁区中的编号。当某点在自由规划区或者禁止区中时高 16 位没有意义。当产生威胁区时系统按顺序将每个威胁区坐标存入一个向量中，高 16 位编号即代表某个点所在威胁区在向量中的下标。当低 16 位为 1 时表示自由规划区，为 2 时表示禁止区，此时高 16 位无意义，当低 16 位未 4 时表示威胁区，高 16 位为相应编号。每种区域的编码如下图所示：

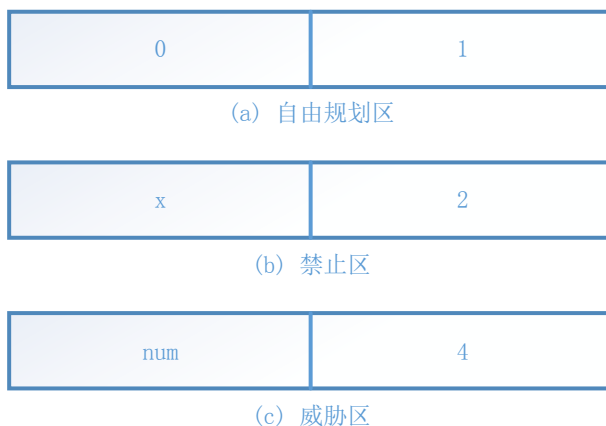


图 5.6 威胁区编码

如图 5.6，(a)为自由规划区的编码，前两个字节为 0，后两个字节为 1。(b)为禁止区编码，前两个字节的值不确定，当威胁区与禁止区重合时  $x$  为威胁区编号  $num$ ，否则为 0。(c)为威胁区编码，前两个字节为威胁区的编号  $num$ ，后两个字节为 4 代表是威胁区。

### 5.3.3 程序算法

本模块分为两步：初始化组合图，生成组合图。

- 1) 初始化组合图时将所有点的信息初始化为自由规划区。
- 2) 生成组合图时分为将威胁区写入组合图和禁止区写入组合图两步。
  - a) 写入威胁区信息时先求出每个圆的外接矩形，然后依次判断外接矩形中的每个点是否在圆中，如果在圆中则将相应点设置为威胁区，并将这个点所在威胁区的编号写入组合图。
  - b) 写入禁止区信息时循环将每个禁止区中的点设置为禁止区即可。

### 5.3.4 模块流程

本模块处理逻辑流程如下所示：

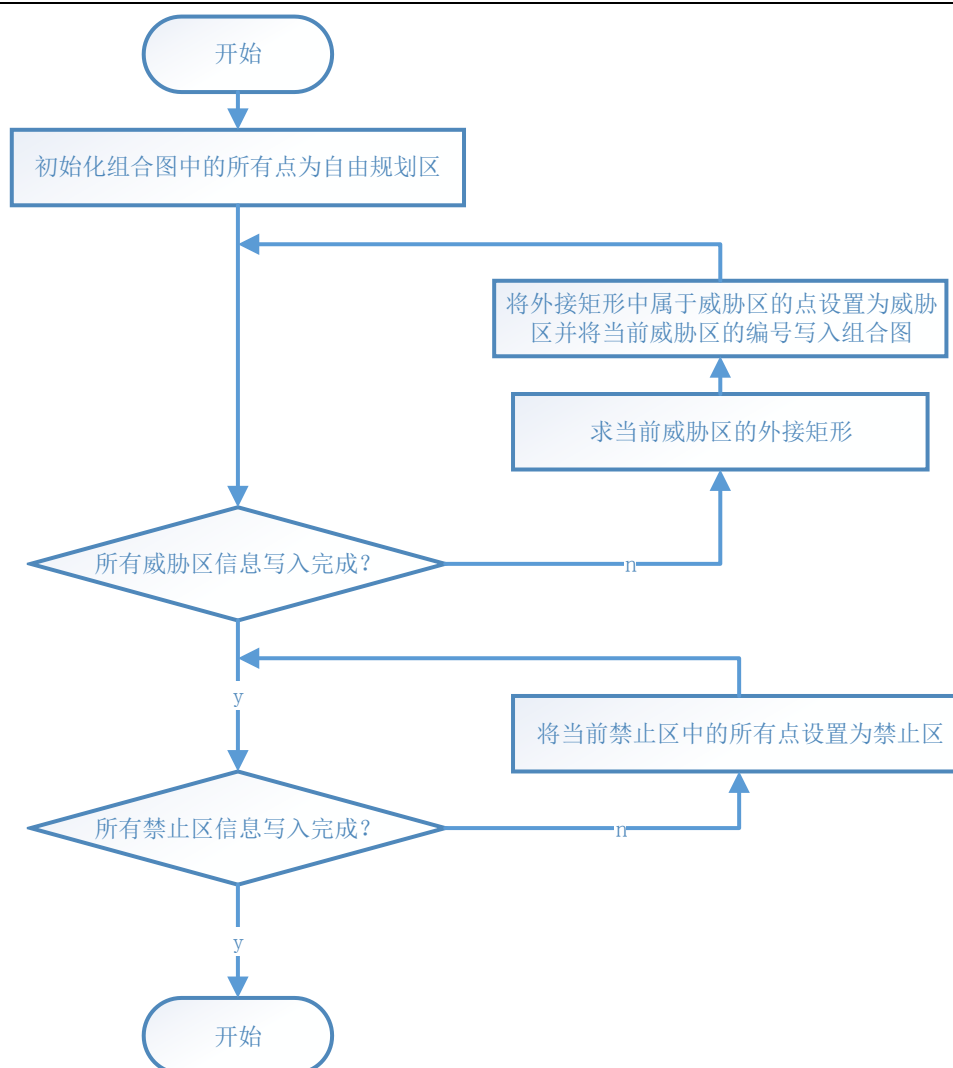


图 5.7 预处理逻辑

## 5.4 线路规划设计

### 5.4.1 模块说明

本模块是系统的核心模块，在前面的各种处理基础上规划出最终的路径。线路规划算法的设计优劣直接影响系统的性能和用户体验。算法在用户指定的起止点，威胁区信息和禁止区信息上规划出一条满足电力线路规划约束的航迹。通过研究现有航迹规划算法，以及综合电力线路规划约束条件，为保证算法的效率本模块采用稀疏 A\* 算法进行搜索。文献[24]研究了综合各种路径代价影响下的线路规划方法，文献[25-27]讨论了改进数据结构对算法进行优化，文献[28-31]讨论了通过改进搜索策略对算法进行优化。

### 5.4.2 程序算法

#### ➤ 计算搜索区域

稀疏 A\* 算法在搜索过程中将结点分为三类：已扩展的结点存放在 *CLOSE* 表中，待

扩展的结点存放在 *OPEN*，尚未扩展的结点。传统 A\* 算法扩展点时会扩展与所有与待扩展点相邻的点，稀疏 A\* 算法将塔杆间最小距离  $L_{min}$ ，最大距离  $L_{max}$ ，最大转弯角度  $\alpha$  考虑在内先进行平面规划，当平面规划符合条件后进行高度规划，即扩展点需满足最大爬升角度  $\beta$ 。将上述规划约束综合后形成一个以  $L_{max}$  为半径，圆心角为  $2 \times \alpha$  的扇形有限规划区域，为缩小搜索规模再次将搜索区域等分  $S$  份，整个搜索区域如图 5.8 所示。

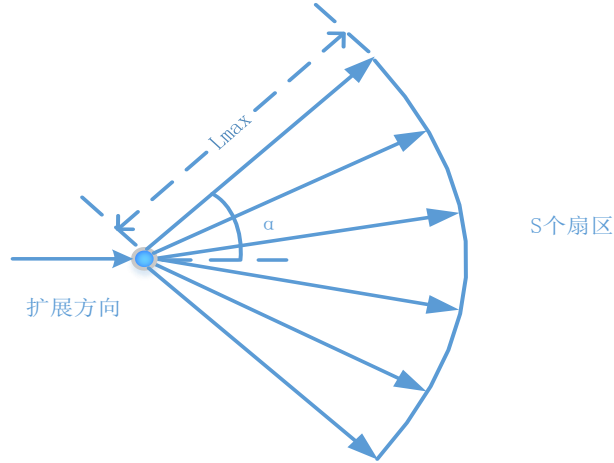


图 5.8 搜索区域

### ➤ 扩展结点

如图 5.9 所示，算法在搜索过程中扩展点时不需要遍历该区域内的每个结点，只需考虑区域内的具有代表性的点即可。假设把搜索区域均匀划分为  $S$  个扇区，每个扇区中只取与当前结点距离  $l$  满足  $L_{min} \leq l \leq L_{max}$  的点，算法并不扩展每个扇区中所有满足条件的点，算法逆时针遍历每个扇形中第一条边上满足条件的点，并计算这些点的代价。根据稀疏 A\* 算法每个扇区只保存代价最小的点，则每个结点产生的扩展节点共有  $S$  个。

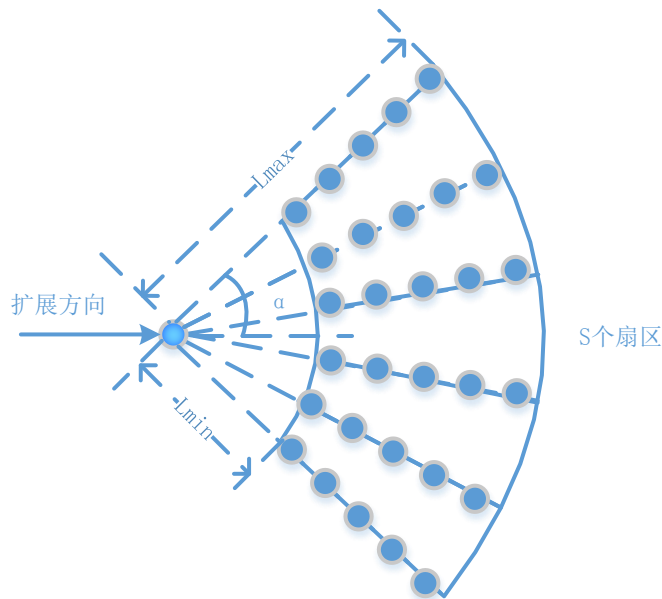


图 5.9 扩展扇形中的点

### ➤ 扩展结点合法性判断

算法根据当前待扩展点  $n$  找到一个尚未扩展的点  $m$  时需要判断结点  $m$  是否在威胁区或禁止区中，如果结点  $m$  不在禁止区中，则算法根据结点  $m$  的坐标从组合图中取得  $m$  的高程值，然后判断从当前待扩展点  $n$  到结点  $m$  的爬升或下滑角是否满足要求，通过两点间的高度与距离可以算出爬升下滑角，其计算公式如下所示：

$$\beta' = \tan^{-1}(|h_n - h_m| \div \sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}) \quad (5.2)$$

如果有  $\beta' \leq \beta$  则结点  $n$  满足约束条件，此时判断点  $n$  与  $m$  是否可以建立航迹，即结点  $n$  与结点  $m$  连线之间是否所有点都不在禁止区中，如果结点  $n$  与结点  $m$  连线之间有任何某个点在禁止区中则结点  $m$  都必须舍弃，算法流程如下图所示：

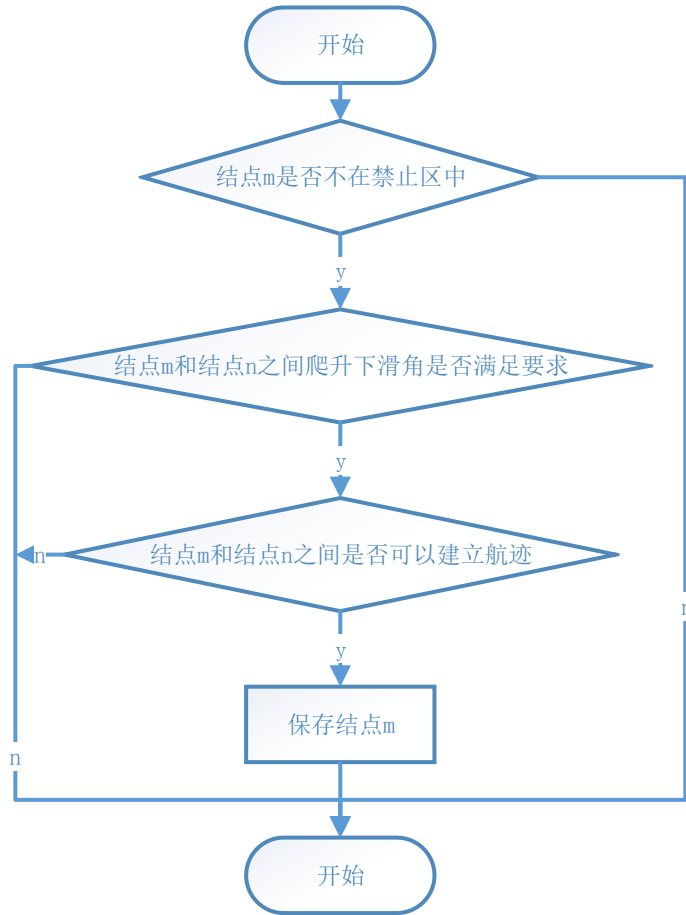


图 5.10 扩展结点合法性判断

### ➤ 计算扩展方向与扩展角

算法根据当前待扩展点  $s$  与目标点  $t$  确定扩展方向，以结点  $s$  与结点  $t$  的连线为角平分线，等分一个角度为  $2 \times \alpha$  的角，点  $s$  到  $t$  的射线的方向就为扩展方向。扩展方向射线的倾斜角为：

$$\theta = \tan^{-1}(|y_t - y_s| \div |x_t - x_s|) \quad (5.3)$$

扩展区域角度的下限  $\alpha_{min}$  为：

$$\alpha_{min} = \theta - \alpha \quad (5.4)$$

扩展区域角度的上限 $\alpha_{max}$ 为:

$$\alpha_{max} = \theta + \alpha \quad (5.5)$$

扩展区域角度上下限满足关系:

$$\alpha_{max} - \alpha_{min} = 2 \times \alpha \quad (5.6)$$

扩展方向与扩展角度的关系如下图所示:

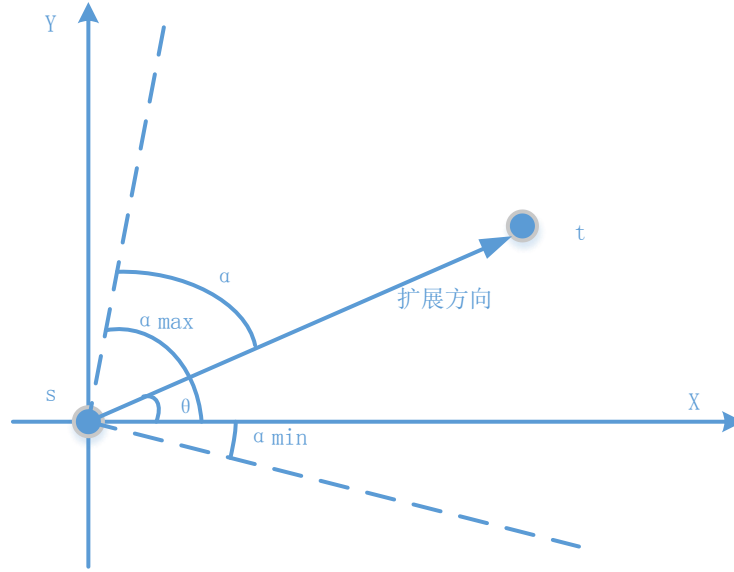


图 5.11 扩展方向与扩展角度

### ➤ 扩展点代价计算

代价函数是 A\*算法的核心, A\*算法本身并不是特别复杂, 它的复杂度主要体现在代价函数的设计, 代价函数主要是用于控制算法的搜索方向和搜索效率, 使算法“带有方向”的搜索。代价函数的设计需要综合考虑线路规划的各种约束条件, 这样搜索形成的路径才具有实用性。根据公式(3.3)可知代价函数分为两部分: 实际代价 $g(n)$ , 启发代价 $h(n)$ 。

当前结点  $n$  到其前驱结点  $p$  的已知代价包括点  $n$  到点  $p$  的距离代价 $d_{pn}$ , 点  $n$  和点  $p$  的高度代价 $h_{pn}$ , 点  $n$  到点  $p$  连线经过威胁区的代价 $thr_{pn}$ 。故而结点  $n$  的实际代价为已知代价与其前驱结点  $p$  的实际代价  $p.g$  的和。如下所示:

$$n.g = d_{pn} + h_{pn} + thr_{pn} + p.g \quad (5.7)$$

距离代价为 $d_{pn}$ 为欧几里得距离, 高度代价为两点间的高度差。设源点  $s$  到结点  $n$  的路径为 $p(s, n) = \langle s, n_1, n_2, \dots, n \rangle$ , 其长度为  $k$ , 则  $n.g$  还可表示为:

$$n.g = \sum_{i=1}^k (d_i + h_i + thr_i) \quad (5.8)$$

其中 $d_i$ 表示  $p(s, n)$ 中第  $i$  条边 $(n_{i-1}, n_i)$ 的距离代价,  $h_i$ 表示第  $i$  条边的高度代价,  $thr_i$ 表

示第  $i$  条边的威胁代价。

启发代价包含当前点  $n$  到目标点  $t$  的距离代价  $d_{nt}$ ，点  $n$  和目标点  $t$  的高度代价  $h_{nt}$ ，以及当前点  $n$  和目标点  $t$  连线之间经过威胁区的代价  $fbd_{nt}$  和禁止区的代价  $thr_{nt}$ 。

$$n.h = d_{nt} + h_{nt} + fbd_{nt} + thr_{nt} \quad (5.9)$$

其中  $fbd_{nt}$  为当前点  $n$  与目标点  $t$  的连线经过禁止区的长度和禁止因子的积。

如果路径  $p$  经过了某个威胁区  $i$  则其威胁代价计算公式为：

$$thr = \mu \times (r_i - d_i) \quad (5.10)$$

其中  $\mu$  为威胁因子， $r_i$  威胁区  $i$  的半径， $d_i$  为路径  $p$  到威胁区圆心的距离，所以如果路径离圆心越近威胁代价越大。如图 5.2 所示为威胁区与规划路径相交的场景，图中路径  $p(n_1, n_2)$  经过圆 1 的圆心，与圆 2 和圆 4 相交，与圆 3 不相交。根据公式(5.10)则线路  $p(n_1, n_2)$  的威胁代价为： $thr = \mu \times [(r_1 - d_1) + (r_2 - d_2) + (r_4 - d_4)]$ 。

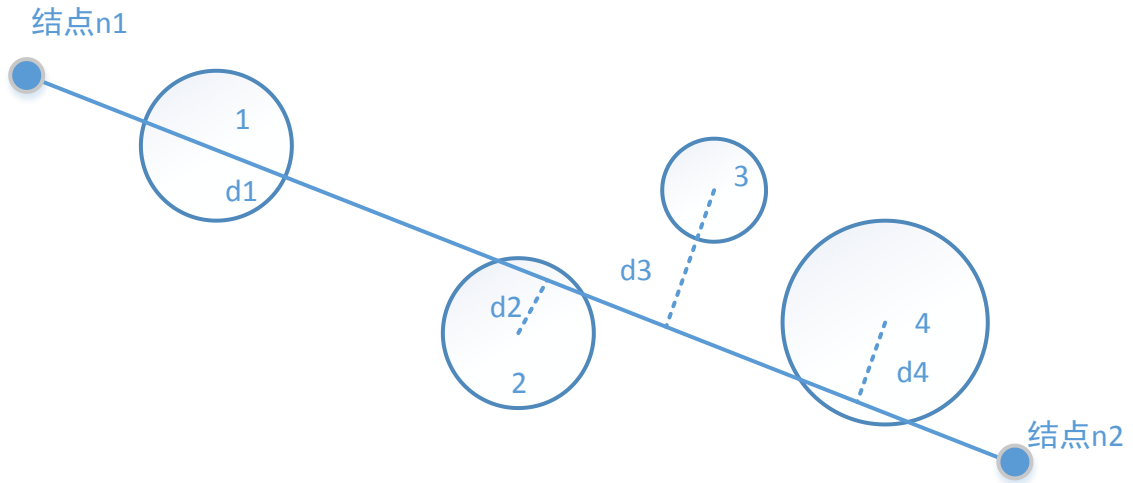


图 5.12 威胁区与规划线路相交

当计算待扩展点  $n$  的启发代价时需计算点  $n$  与目标点  $t$  经过的禁止区代价，设点  $n$  与目标点  $t$  的连线经过了禁止区  $i$  则其禁止代价计算公式为：

$$fbd = \sigma \times d_i \quad (5.11)$$

其中  $\sigma$  为禁止因子， $d_i$  为点  $n$  到目标点  $t$  连线进过禁止区  $i$  的长度。如图 5.13 所示为点  $n$  和目标点  $t$  连线与禁止区相交的场景，则其禁止区代价为  $fbd = \sigma \times (d_{ab} + d_{cd})$ 。

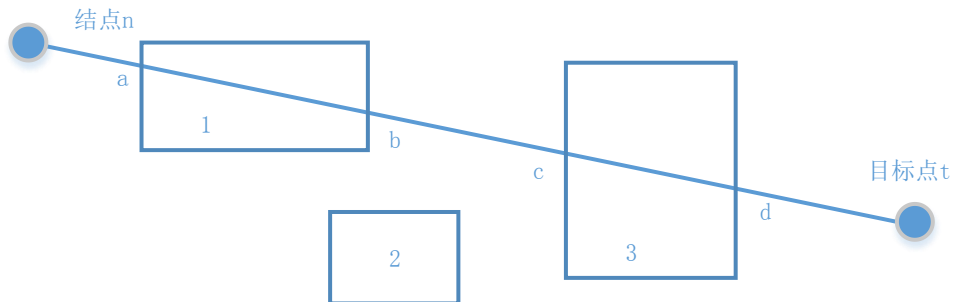


图 5.13 待扩展点和目标点连线与禁止区相交





### 5.4.3 数据结构设计

#### ➤ 图的存储

网格地图在平面坐标系下按照  $x$  坐标和  $y$  坐标将平面划分为无数个网格，在特定境下可以将每个网格抽象为一个只具有  $x$  与  $y$  坐标以及一个属性  $z$  的质点，整个地图由这些质点按照其坐标排列成一个矩阵。本模块将地图设计为一个二维数组的形式， $map[n][m]$  表示整个地图有  $n$  列  $m$  行， $map[x][y]$  表示坐标  $(x,y)$  处点的高程值。由于整个地图可能非常大，程序使用 Windows 内存映射加载地图数据。

#### ➤ 结点数据结构

算法在搜索过程中会产生很多中间结点，根据第 3 章的知识每个结点包含的信息有：点的  $x$  和  $y$  坐标，点的高程值  $z$ ，从源点到当前点的实际代价值  $g$ ，从当前点到目标点的代价预估值  $h$ ， $g$  值和  $h$  值的总和  $f$ ，当前点的前驱结点  $p$ 。本模块将结点信息设计为一个 C 结构体 *PointInfo*，当搜索到一个新的点时就申请一块 *PointInfo* 大小的内存存放点的信息。

#### ➤ OPEN 表设计

*OPEN* 表用于存储算法在搜索过程中搜索到的点，在本模块中将 *OPEN* 表设计为一个链表，链表中的点按照  $f$  值从小到大排序， $f$  值最小的为表头。*OPEN* 表中存放结点信息结构体的首地址，*OPEN* 表的设计为 *std::list<PointInfo \*>OpenList*。当有新的结点加入 *OPEN* 表时，首先查看表中是否存在某点的坐标与其相同，如果存在且其  $f$  值比新的点  $f$  值大则删除原来的点，并将新的点插入 *OPEN* 表中，如果不存在则直接按照  $f$  的大小插入 *OPEN* 表中。

#### ➤ CLOSE 表设计

*CLOSE* 表存放算法已经扩展过了的点，与 *OPEN* 表相似，*CLOSE* 也保存点信息结构体的首地址。*CLOSE* 表的设计为 *std::list<PointInfo \*>CloseList*，当从 *OPEN* 表中取得一个最优点后可以直接放入 *CLOSE* 表中。

### 5.4.4 算法流程

前面几各小节主要讲述了算法如何缩小搜索区域，如何在搜索区域中扩展一个点，如何计算待扩展点的估计代价。由于第 3 章已经详细的描述了 A\* 算法的执行流程，所以本节将不在赘述，本节将把上节讨论的子步骤综合起来简明扼要的概述算法的主要流程。算法的主要流程如下所述：

- 1) 清空 *OPEN* 表和 *CLOSE* 表，将源点  $s$  的各个代价设置为 0 放入 *OPEN* 表。
- 2) 当 *OPEN* 表不为空时循环从 *OPEN* 表中取出代价值最小的点  $n$ ，并将点  $n$  加入 *CLOSE* 表中。
- 3) 判断目标点  $t$  是否在点  $n$  的搜索区域内
  - a) 如果目标点  $t$  在点  $n$  的搜索区域内则判断点  $n$  与点  $t$  是否可以直接连接（ $n$  与  $t$

的连线不经过威胁区和禁止区)。

- b) 如果目标点可以连接则找到一条路径, 搜索结束。
  - c) 如果目标点不在  $n$  的搜索区域内或者不可连接则继续。
- 4) 将点  $n$  扩展。
- a) 计算点  $n$  的搜索扇形区域。
  - b) 将点  $n$  的扇形搜索区域均分为  $S$  个扇区, 然后计算每个扇区中可以扩展的点  $m$ 。
  - c) 如果点  $m$  不在  $OPEN$  表中则将点  $m$  加入  $OPEN$  表, 回到第(2)步继续搜索。

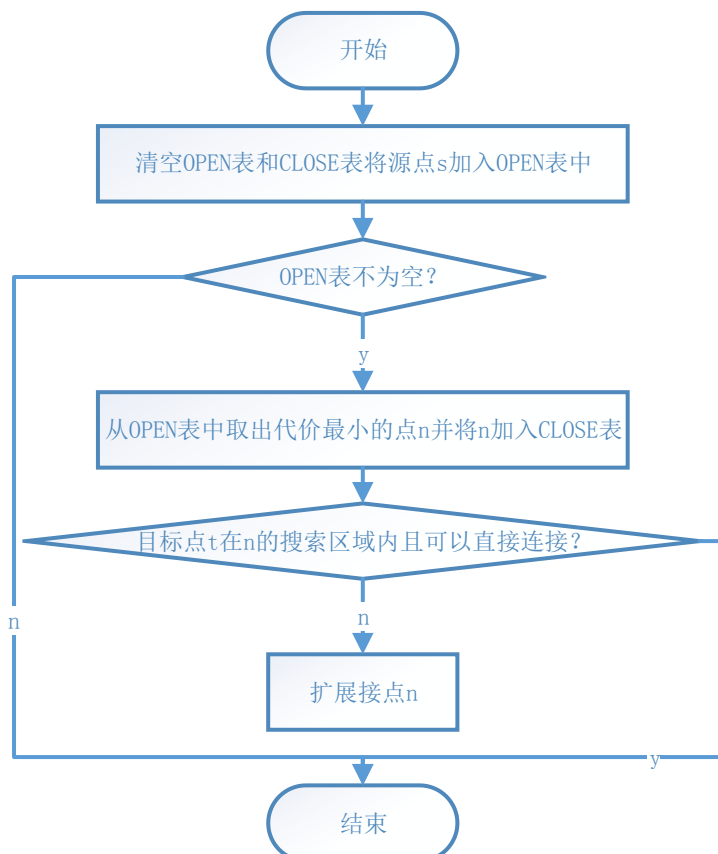


图 5.14 A\*算法主流程

## 5.5 效果演示设计

线路规划模块规划线路结束后会生成一个规划结果文件, 结果文件包含一系列的点, 每个点包含有经纬度和高度值。这个结果文件可视性很差不利于分析系统规划的结果是否符合要求, 于是需要把这个规划结果在地图上演示。本模块首先读取文件中每个点的经纬度, 然后将每个点的经纬度转换为灰度图中的坐标即屏幕坐标, 最后将这些点以折线的方式画在灰度图上显示给用户。



## 6 实验结果演示

### 6.1 系统主界面

系统启动时的主界面将会充满整个屏幕，如图 6.1 所示：

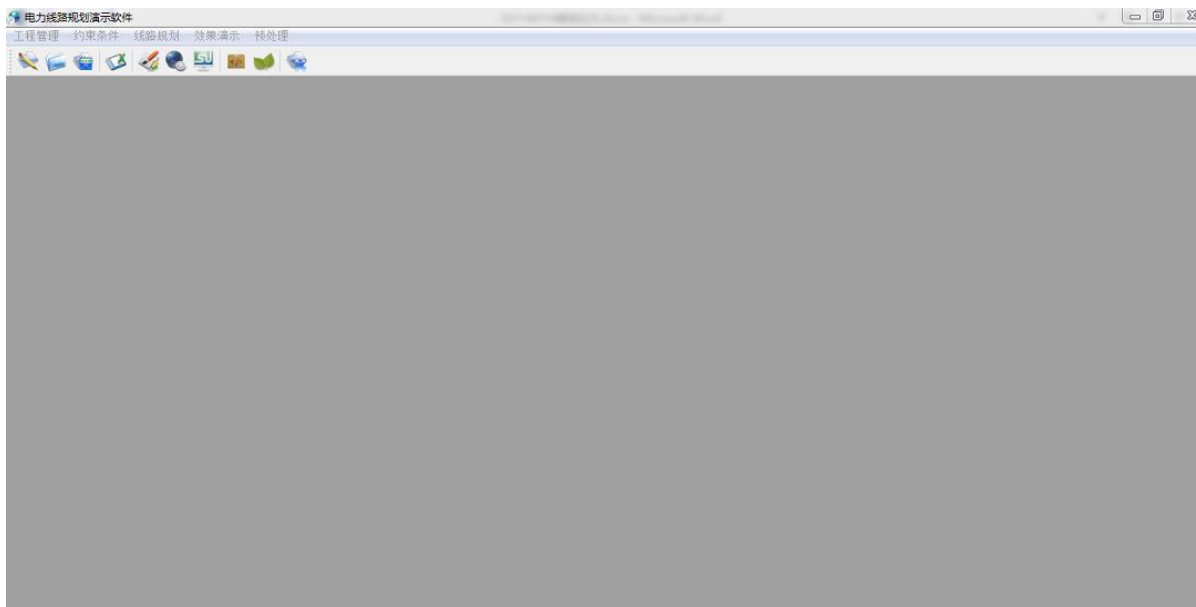


图 6.1 系统启动界面

### 6.2 工程管理功能

#### 6.2.1 新建工程文件

新建工程文件菜单完成原始 DEM 地形数据到高程图数据的转换。如图 6.2 和 6.3 所示，本模块会让用户输入原始 DEM 的文件名以及转换后高程图的文件名。

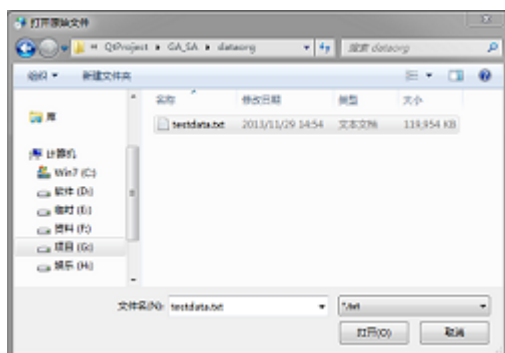


图 6.2 打开原始 DEM 文件

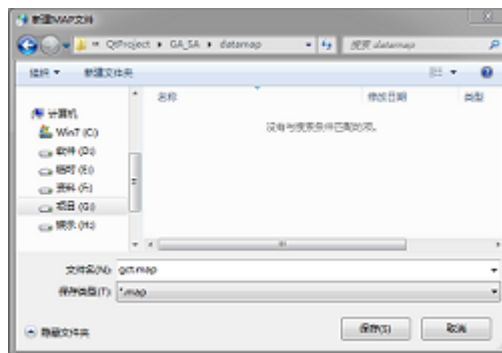


图 6.3 新建高程图文件

新建工程模块在得到原始 DEM 地形数据的文件路径，以及新建高程图数据文件名后，将把原始地形数据转换为规则的高程图数据以便后续处理，由于数据量大转换时转换工作在后台进行，主界面会提示转换进度。转换进度及成功提示如图 6.4 和 6.5 所示。

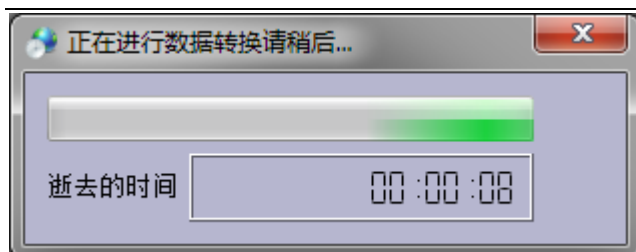


图 6.4 数据转换进度



图 6.5 数据转换成功提示

### 6.2.2 打开工程文件

打开工程文件菜单读取高程图数据并将其转换为可视化的灰度图显示在主界面上，图 6.6 为打选择高程图文件名对话框，图 6.7 为高程图转换为灰度图后的系统主界面。

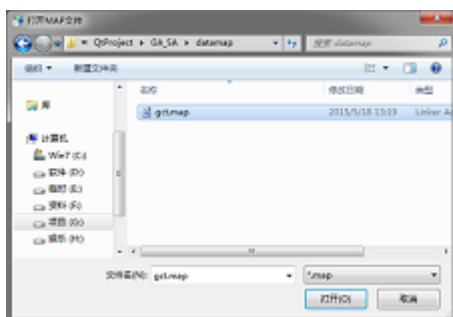


图 6.6 打开高程图

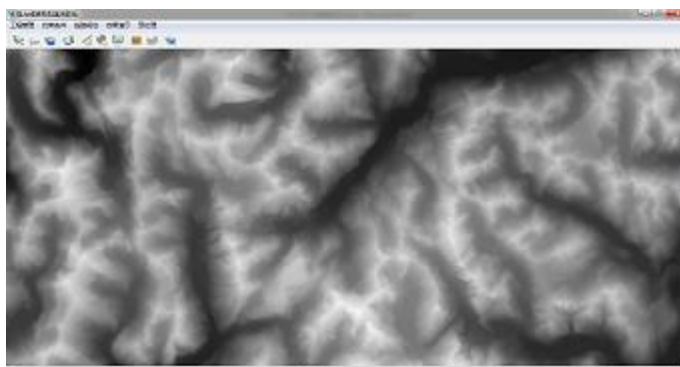


图 6.7 高程图转换后的灰度图

### 6.2.3 关闭工程

关闭工程子模块会清理工程打开期间所申请的资源并将系统主界面设置为图 6.1 所示的启动时的状态，这里就不在赘述。

## 6.3 约束条件功能

如图 6.8 所示，图中三角形为起止点，圆形为威胁区，矩形为禁止区。



图 6.8 起止点与约束区域



## 6.4 预处理功能

预处理模块负责将约束条件模块中的约束区域写到组合图中，生成工作在后台完成，系统主界面显示进度。如图 6.9 和图 6.10 所示为生成组合图时的进度条以及成功提示。

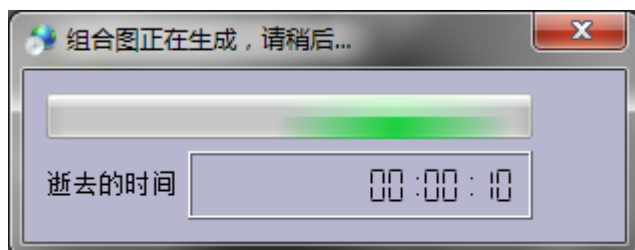


图 6.9 组合图生成进度

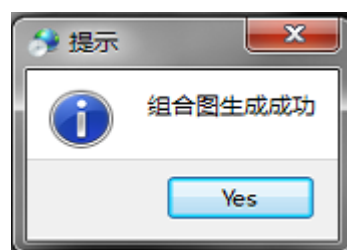


图 6.10 组合图生成成功提示

## 6.5 线路规划功能

### 6.5.1 输入参数

输入参数子模块将设置线路规划过程中用到的各种约束参数。输入参数界面如图 6.11 所示。



图 6.11 输入参数界面

### 6.5.2 线路规划

本模块首先读取规划任务文件，然后在后台进行规划，系统主界面显示规划操作所耗时间。如图 6.12 和图 6.13 所示为读取任务文件名对话框，以及规划任务进度条。

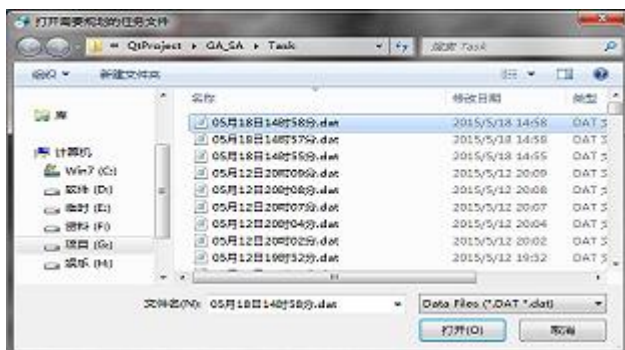


图 6.12 打开规划任务文件

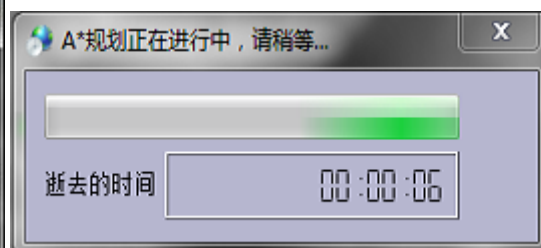


图 6.13A\*规划进度



## 6.6 规划结果演示

为了直观的判断规划结果是否正确，效果演示模块将规划结果显示在系统主界面的灰度图上。为了说明算法的正确性，将线路规划模块在不同的约束区域中运行，并演示其规划结果，实验使用的数据大小为  $1500 \times 1500$  的 DEM 地形数据，即整个地图中有 2250000 个点。前面讨论到威胁区线路可以通过但是有一定的代价，所以算法规划的路径应该尽量避开威胁区，即离威胁区中心尽量远。如图 6.14 所示，线路避开了威胁区。

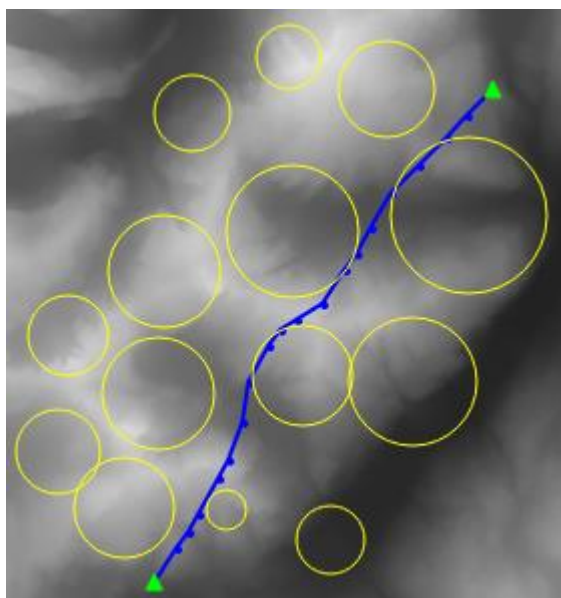


图 6.14 在威胁区中规划路径

图 6.15 演示了算法在禁止区中找到的一条最优路径，线路不能穿过禁止区。如图所示，算法所规划的路径有效的避开了其中的禁止区域。

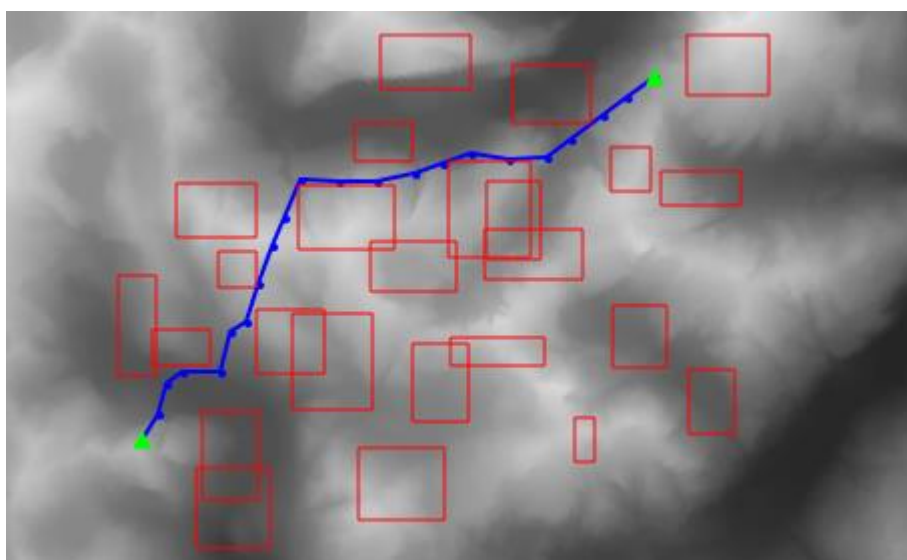


图 6.15 在禁止区中规划路径



如图 6.16 演示了在综合约束区域中，算法规划的路径。从图中可以看出算法规划的路径依然有效的避开了禁止区域，在无法避免穿过威胁区时，线路采取了尽量远离威胁区中心的措施。

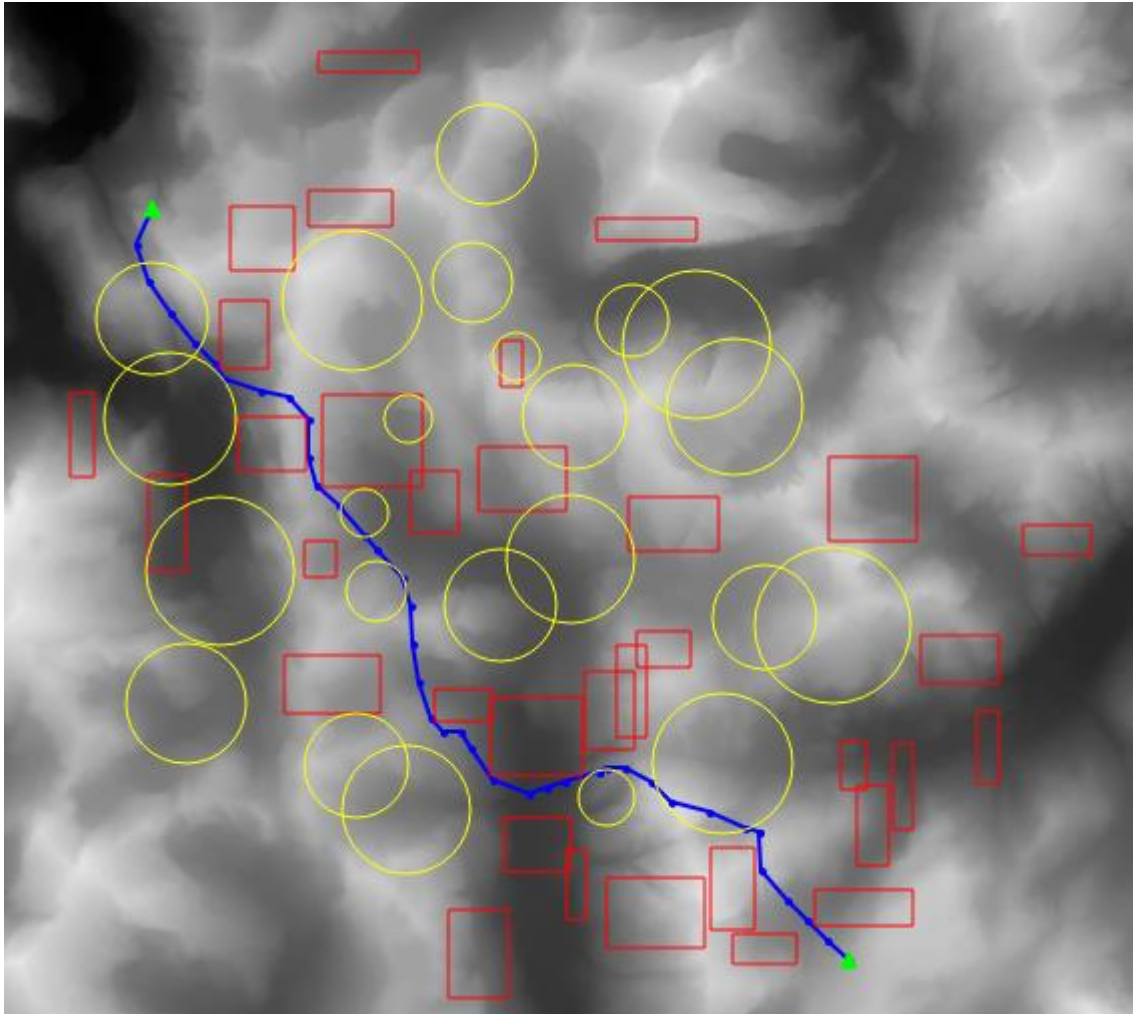


图 6.16 在综合约束区域中规划路径





## 7 总结与展望

### 7.1 论文主要内容总结

电力线路规划是电力系统的重要组成部分，由于人工选线的局限性使得利用信息化手段规划电力线路成为研究重点。电力线路规划综合了规划任务，规划地理环境以及电力架线与维修的要求等约束条件，从而规划出一条从起始点到目标点的最优可行路径。本系统根据 DEM 地形数据，在 Windows 平台下用 C/C++ 语言与 QT 图形界面完成了电力线路规划演示软件的设计与实现。重点讨论了 A\* 算法的理论基础与启发函数的原理，在电力线路规划约束条件下改进并实现了可用于电力线路规划的 A\* 算法，同时验证了算法的正确性。本文主要探讨了以下内容：

- 1) 探讨了电力线路规划的国内外研究现状，介绍了使用 A\* 算法完成线路规划的优缺点。
- 2) 完成了电力线路规划演示软件的设计，程序实现及测试工作，对程序中的关键数据结构进行了分析与设计。
- 3) 讨论了 A\* 算法的基本原理，改进并实现了适用于电力线路规划约束条件的 A\* 搜索算法。
- 4) 设计并实现了线路规划演示软件的人机交互界面。

### 7.2 不足与展望

由于作者能力和时间有限，对本课题所涉及的内容探讨还不够全面和深入，探讨方法也有不甚合理之处，还有待进一步改善：

- 1) 软件人机界面设计还不够合理，软件很多设计没用考虑用户的使用习惯，还需改进。
- 2) 规划算法的效率还有待提高，由于组合图的设计不够合理使得组合图比高程图还大，所以当数据比较大时，组合图不能全部装入内存，对文件的读写严重影响了算法的效率，可以考虑去掉组合图改用算法判断点是否在约束区域中。
- 3) 算法规划的线路平滑度不够，这是由于本系统算法弱化了结点转弯控制造成的，对线路的平滑处理还需进一步研究。
- 4) 在线路规划过程中高度规划的设计不够合理，在高度规划过程中由于只是判断了相邻两点的爬升下滑角是否合理，这只能局部优化线路，若将最终的规划结果综合来看有可能出现高度参差不齐的情况。



## 参考文献

- [1] 杨华壮, 刘权锋. 中国电力工业发展概述[J]. 中国科技博览, 2011(32):376.
- [2] 孙庆海. 浅谈中国电力事业的发展与规划[J]. 房地产导刊, 2014(35):189.
- [3] 辜体仁. 电力行业信息化发展及现状介绍[A]. OA'2011 办公自动化国际学术研讨会论文集(C), 北京. 2011:1-6.
- [4] 王艳丽. 输变电工程项目送电线路最优路径选择研究[D]. 保定: 华北电力大学, 2012.
- [5] 康健民, 袁敬中, 肖少辉, 赵俊生, 刘震. 基于分层模型的输电线路选线算法设计[J]. 电力建设, 2012, 33(4):6-10.
- [6] 杜全维, 王全心. 基于卫星遥感技术的电力选线系统设计与实现[J]. 科技资讯, 2009(28):22-23.
- [7] Miguel V, Hector G, Sarmiento. Image processing application maps optimal transmission routes[J]. IEEE Computer Applications in Power, 1996, 3(1):33-39.
- [8] West N A, Dwolatzky B, Meyer A S. Terrain-based routing of distribution cables[J]. IEEE Computer Applications in Power, 1997, 4(1):42-47.
- [9] Salman A, Hamid E, Valadan Z M. A new method for path finding of power transmission lines in geospatial information system using raster networks and minimum of mean algorithm [J]. World Applied Sciences Journal, 2008, 3(2):269-277.
- [10] Volkan Y, Recep N. Developing a geospatial model for power transmission line routing [C] //Proceedings of the 3rd Africon Conference. Ezulwini Valley, Swaziland:IEEE, 1992:59-64.
- [11] Schmidt A J. Implementing a GIS methodology for siting high voltage electric transmission lines[J]. Papers in Resource Analysis, 2009, 11(2):58-63.
- [12] Ward J, Fellow I, Ted G, et al. A New method for public involvement in electric transmission-line routing[J]. IEEE Trans On Power Delivery, 2009, 24(4):33-38.
- [13] 刘春霞, 王家海. 基于空间分析的送电线路选线方法[J]. 测绘工程, 2008, 17(3):28-30.
- [14] Yao J F, Lin C, Xie X B, et al. Path planning for virtual human motion using improved A\* star algorithm[C] //Proc. Of the 7<sup>th</sup> International Conference on Information Technology: New Generations, 2010:1154-1158.
- [15] Kang H, Lee B, Kim K. Path planning algorithm using the particle swarm optimization and the improved dijkstra algorithm[C] //Proc. Of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application, 2008: 1002-1004.
- [16] Ju M Y, Cheng C W. Smooth path planning using genetic algorithms[C] //Proc. Of the 9th World Congress on Intelligent Control and Automation, 2011:1103-1107.
- [17] Zhao S G, Li M. Path planning of inspection robot based on ant colony optimization algorithm [C] //Proc. of the International Conference on Electrical and Control Engineering, 2010:1474-1477.



- [18] Shakiba R, Najafipour M, Salehi M E. An improved PSO-based path planning algorithm for humanoid soccer playing robots[C] //Proc. Of the 3rd Joint Conference of AI & Robotics and the 5<sup>th</sup> RoboCup Iran Open International Symposium, 2013:1-6.
- [19] Thomas H.Cormen, Charles E.Leiserson, Ronald L.Rivest, Clifford Stein 著. 算法导论(第2版)[M]. 潘金贵, 顾铁成, 李成发, 叶懋译. 北京: 机械工业出版社, 2006. 9:667.
- [20] 李泉扬, 周德云, 冯琦. 基于分级规划策略的 A\*算法多航迹规划[J]. 系统工程与电子技术, 2015(2): 318-322.
- [21] 钟敏. A\*算法估价函数的特性分析[J]. 武汉工程职业技术学院学报[J], 2006, 18(2): 31-33.
- [22] 王德春等. 基于 A\*算法的舰船最佳航线选择[J]. 青岛大学学报, 2005, 18(4):10-13.
- [23] 杨素琼等. 基于 A\*算法的地图路径搜索的实现[J]. 铁路计算机应用, 2001, 9(4):8-11.
- [24] 李季, 孙秀霞. 基于改进 A-Star 算法的无人机航迹规划算法研究[J]. 兵工学报, 2008, 29(7):788-792.
- [25] 乐阳, 龚健雅. Dijkstra 最短路径算法的一种高效率实现[J]. 武汉测绘科技大学学报, 1999(03):209-212.
- [26] 王开义, 赵春江, 胥桂仙. GIS 领域最短路径搜索问题的一种高效实现[J]. 中国图象图形学报. A 2003(08):951-956.
- [27] 陆锋, 卢冬梅, 崔伟宏. 交通网络限制搜索区域时间最短路径算法[J]. 中国图象图形学报. A 1999(10):849-855.
- [28] 程林, 王美玲, 张毅. 一种基于 SuperMaP GIS 的改进 Dijkstra 算法[J]. 地球信息科学学报, 2010(5):649-653.
- [29] 杨琰, 廖伟志, 李文敬, 杨文, 李杰. 基于 Petri 网的顾及转向延误的最优路径算法[J]. 计算机工程与设计, 2013(10):3643-3647.
- [30] 杜立智, 张晓龙. 一个高效的 3SAT 到 Hamilton 环转化方法[J]. 南京理工大学学报(自然科学版), 2013(4):506-510.
- [31] 许岩峰, 王巍. 浅谈突发事件应急物资调度[J]. 科技广场, 2012(6):210-213.