

# 基于大数据平台的无人机技术研究概述

冯学伟 学号:2019520941

2020年5月20日

**摘要:**无人机凭借其低成本、高效率等优点，正成为提升农业植保科技含量的重要装备。针对目前无人机作业资源调度不合理、效率低下以及地主和飞服组织之间缺乏有效的沟通渠道等问题，基于互联网大数据云平台对无人机调度技术进行了研究。该平台一共分为 5 层，分别为数据采集层、网络传输层、平台层、访问层、用户层。根据不同用户的需求设计了不同的 APP，分别为“我是地主”和“我是飞手”，为地主与飞服组织之间搭起了沟通的桥梁，实现了飞手和种植户的供需无缝对接。以飞手为服务对象，提出基于回溯策略的无人机调度算法，在作业季内预计可提高飞手收益 20.1% 之中，为实现农业信息化做出了有益的尝试

**关键字:** 无人机; 大数据; 互联网; 调度平台

**Abstract:** The standard and high efficiency of the UAV itself are becoming important equipment for improving the scientific and technological content of agricultural plant protection. In view of the current unreasonable drone operation resource scheduling, low efficiency, and the lack of effective communication channels between the landlord and the flight service organization, the platform is divided into 5 layers, namely the data collection layer, network transmission layer, platform layer, and access layer, User layer. Different APPs have been designed according to the needs of different users, namely "I am the landlord" and "I am the flying hand", which establishes a coaxial communication between the landlord and the flying suit organization, realizing the flying hand and the grower's Seamless supply and demand. Taking the pilot as the service object, a drone scheduling algorithm based on the backtracking strategy is proposed, which is expected to increase the pilot's income in the 20.1 industry during the operating season, making a useful attempt to realize agricultural informatization.

**keywords**— UAV, the big data, the Internet, the Dispatching platform

## 1 scala学习心得

经历了几周的关于scala的学习，对编程的认知有更上了一个层次；下面我将就这一段时间的学习，来梳理出相关的学习内容：

### 1.1 函数式编程（fp）、面向对象编程（OOP）、面向过程编程（POP）

面向过程编程（POP）就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了。面向对象编程（OOP）是把构成问题的事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描述某个事物在整个解决问题的步骤中的行为。我们通过把大段代码拆成函数，通过一层一层的函数调用，就可以把复杂任务分解成简单的任务，这种分解可以称之为面向过程的程序设计。函数就是面向过程的程序设计的基本单元。

而函数式编程（Functional Programming），虽然也可以归结到面向过程的程序设计，但其思想更接近数学计算。我们需要先搞清楚计算机（computer）和计算（compute）的概念；在计算机的层次上，CPU执行的是加减乘除的指令代码，以及各种条件判断和跳转指令，所以，汇编语言是最贴近计算机的语言。而计算则指数学意义上的计算，越是抽象的计算，离计算机的硬件越远。对应到编程语言，就是越低级的语言，越贴近计算机，抽象程度低，执行效率高，比如C语言；越高级的语言，越贴近计算，抽象程度高，执行效率低，比如Lisp语言。

函数式编程就是一种抽象程度很高的编程范式，纯粹的函数式编程语言编写的函数没有变量，因此，任意一个函数，只要输入是确定的，输出就是确定的，这种纯函数我们称之为没有副作用。而允许使用变量的程序设计语言，由于函数内部的变量状态不确定，同样的输入，可能得到不同的输出，因此，这种函数是有副作用的。函数式编程的一个特点就是，允许把函数本身作为参数传入另一个函数，还允许返回一个函数。

### 1.2 Scala内置的数据结构

Scala内置的数据结构包括数组、元组、容器、序列、集合、映射、迭代器等。数组是一种可变的、可索引的、元素具有相同类型的数据集合；Scala提供了参数化类型的通用数组类Array[T]，其中T可以是任意的Scala类型，可以通过显式指定类型或者通过隐式推断来实例化一个数组。关于数组的操作有声明数组、多维数组的创建、数组元素的使用等。

元组是对多个不同类型对象的一种简单封装。定义元组最简单的方法就是把多个元素用逗号分开并用圆括号包围起来。使用下划线“\_”加上从1开始的索引值，来访问元组的元素，如果需要在方法里返回多个不同类型的对象，Scala可以通过返回一个元组来实现。

同时Scala提供了一套丰富的容器（collection）库，包括序列（Sequence）、集合（Set）、映射（Map）等，Scala用了三个包来组织容器类，分别是collection.mutable和immutable；collection封装了可变容器和不可变容器的超类或特质，定义了可变容器和不可变容器的一些通用操作。

序列（Sequence）是一种内部元素可以按照特定的顺序访问的容器。序列中每个元素均带有一个从0开始计数的固定索引位置，序列容器的根是collection.Seq特质。其具有两个子特质 LinearSeq和IndexedSeq。LinearSeq序列具有高效的 head 和 tail 操作，而IndexedSeq序列具有高效的随机存储操作。实现了特质LinearSeq的常用序列有列表（List）和队列（Queue）。实现了特质IndexedSeq的常用序列有可变数组（ArrayBuffer）和向量（Vector）。

Scala内部存在于列表相对的数据结构—集合，集合(set)是不重复元素的容器（collection）；而列表（List）中的元素是按照插入的先后顺序来组织的。“集合”中的元素并不会记录元素的插入顺序，而是以“哈希”方法对元素的值进行组织，所以，集合允许你快速地找到某个元素。

映射(Map)是一系列键值对的容器。键是唯一的，但值不一定是唯一的。可以根据键来对值进行快速的检索。Scala的映射包含了可变的和不可变的两种版本，分别定义在包mutable和immutable里。默

认情况下，Scala中使用不可变的映射。如果想使用可变映射，必须明确地导入mutable.Map。若想要获取映射中的值，可以通过键来获取。

迭代器（Iterator）不是一个容器，而是提供了按顺序访问容器元素的数据结构，迭代器包含两个基本操作：next和hasNext。next可以返回迭代器的下一个元素，hasNext用于检测是否还有下一个元素。除next和hasnext方法外，在对一个迭代器调用了某个方法后，不要再次使用该迭代器。

### 1.3 Spark运行架构特点

每个Application获取专属的executor进程，该进程在Application期间一直驻留，并以多线程方式运行tasks。Spark Application不能跨应用程序共享数据，除非将数据写入到外部存储系统；Spark与资源管理器无关，只要能够获取executor进程，并能保持相互通信就可以了，Spark支持资源管理器包含：Standalone、On Mesos、On YARN、Or On EC2；提交SparkContext的Client应该靠近Worker节点（运行Executor的节点），最好是在同一个Rack（机架）里，因为Spark Application运行过程中SparkContext和Executor之间有大量的信息交换；如果想在远程集群中运行，最好使用RPC将SparkContext提交给集群，不要远离Worker运行SparkContext；Task采用了数据本地性和推测执行的优化机制，关键方法：taskIdToLocations、getPreferredLocations。

### 1.4 RDD运行机制

一个 RDD 就是一个分布式对象集合，提供了一种高度受限的共享内存模型，其本质上是一个只读的分区记录集合，不能直接修改。每个 RDD 可以分成多个分区，每个分区就是一个数据集片段，并且一个 RDD 的不同分区可以保存到集群中不同的节点上，从而可以在集群中的不同节点上进行并行计算。

在Spark中，转换的作用是从现有数据集创建新数据集。转换是惰性的，因为它们仅在动作需要将结果返回到驱动程序时才计算。Spark采用textFile()方法来从文件系统中加载数据创建RDD，该方法把文件的URI作为参数，这个URI可以是：本地文件系统的地址；或者是分布式文件系统HDFS的地址；或者是Amazon S3的地址等等。

RDD操作包含转换操作、行动操作和惰性机制。对于RDD而言，每一次转换操作都会产生不同的RDD，供给下一个“转换”使用，转换得到的RDD是惰性求值的，也就是说，整个转换过程只是记录了转换的轨迹，并不会发生真正的计算，只有遇到行动操作时，才会发生真正的计算，开始从血缘关系源头开始，进行物理的转换操作；行动操作是真正触发计算的地方。Spark程序执行到行动操作时，才会执行真正的计算，从文件中加载数据，完成一次又一次转换操作，最终，完成行动操作得到结果；而所谓的“惰性机制”是指，整个转换过程只是记录了转换的轨迹，并不会发生真正的计算，只有遇到行动操作时，才会触发“从头到尾”的真正的计算。

## 2 大数据相关技术介绍

### 2.1 Apache Hadoop 架构

先来简要介绍 Apache Hadoop 架构、Hadoop 分布式文件系统（Hadoop Distributed Filesystem）DFS 和 MapReduce 并行编程模型。

Apache Hadoop 是阿帕奇软件基金会的一个项目,它在 2005 年才逐渐成型，虽然距离现在只过了短短 14 年，但它的发展却是日新月异的并逐渐开启了大数据研究与应用的热潮。Hadoop 最大的特点是不管是老机器还是新机器都可以构建 Hadoop 大数据平台，通过整合集群整体的存储和计算能力使其有更为强大的计算和存储能力。Hadoop 的集群存储和多处备份的机制保证了它有很高的可靠性，它可以在大数据运行的状态添加新机节点，这保证了它有很高的可伸缩性和可扩展性。Hadoop 能方便的实现大数据批量处理，实现大规模数据的分布式存储与并行计算。而且 Hadoop 是开源的，目前很多公司即使

其使用者也是贡献者，使 Hadoop 已成为非关系数据分析的主流技术。当然，关系数据库更容易使用，特别是 SQL 语句，所以为了降低大数据的使用门槛很多人开始致力于大数据平台上运行sql语句的研究如FaceBook为了在Hadoop集群上使用传统的sql语句，而开发了 Hive。如图1所示。主要包括以下模块：

- (1) Hadoop Common：支持其他 Hadoop 模块的常用实用程序。
- (2) Hadoop 分布式文件系统（HDFS™）：一种分布式文件系统，可提供对应用程序数据的高吞吐量访问。
- (3) Hadoop YARN：作业调度和集群资源管理的框架。
- (4) Hadoop MapReduce：基于 YARN 的系统，用于并行处理大型数据集

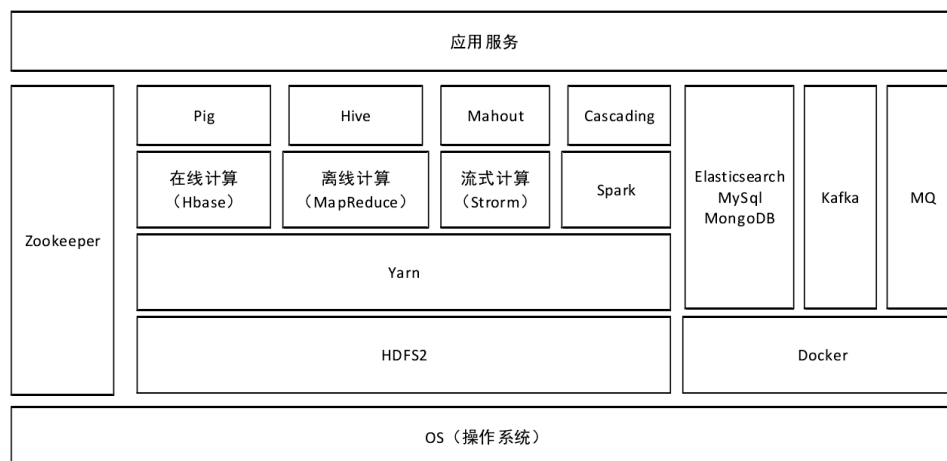


Figure 1: 大数据平台架构图

从图中可以看出在操作系统之上是 Hadoop 的 HDFS 分布式文件存储系统，yarn 在其上作为 Hadoop 资源管理器，它是一个通用资源管理系统，可为上层应用提供统一的资源管理和调度。在 yarn 的上面可以运行像 MapReduce、spark、Hbase、Storm、Pig、Hive、Mahout 等多种分布式计算。ZooKeeper 字面意思是动物管理员，而 pig、hive 又都是各种动物的名字，所有从名字就可以看出 ZooKeeper 是来管理和协调各种分布式应用的，可以使分布式应用更方便的进行分布式计算，在 Hadoop 中发挥着举足轻重的作用。不过不管是哪一种分布式计算，它们所使用的计算的基本思想都是 MapReduce。下面我们重点介绍一下 MapReduce 并行编程模型。

MapReduce 是一个编程模型。谷歌首先提出将其作为大规模数据处理的并行计算模型和方法。它用于大规模数据集（大于 1TB）并行计算。Hadoop 诞生于谷歌在国际会议上发表的两篇关于谷歌分布式文件系统和 MapReduce 的论文 [30]。其中 Map 译为映射，其实就是把任务进行拆分，把不同的任务分别配给不同的机器去执行。Reduce 译为归约其实就是把运算的结果进行进一步的计算再进行合并得到最终结果，其原理图如图2所示：

当向大数据平台提交 MapReduce 任务后，首先主控程序也就是 Master 会向把任务分配给工作机（slave），然后数据被分成多个片段分配给不同的工作机，工作机再对数据进行各种处理并把处理结果先写入工作机的本地磁盘中，然后再有 1 到 2 个工作机把处理结果进行汇总，再输出就是处理完成后的结果了。

## 2.2 Spark 架构

Apache Spark 诞生于 2009 年，因为 Hadoop 虽然开创了大数据开发的先河，但其启动慢，计算也慢，虽然听过增加集群的规模可以提高运算速度，但仍不尽如人意。为了克服 Hadoop 的缺点，在 2009

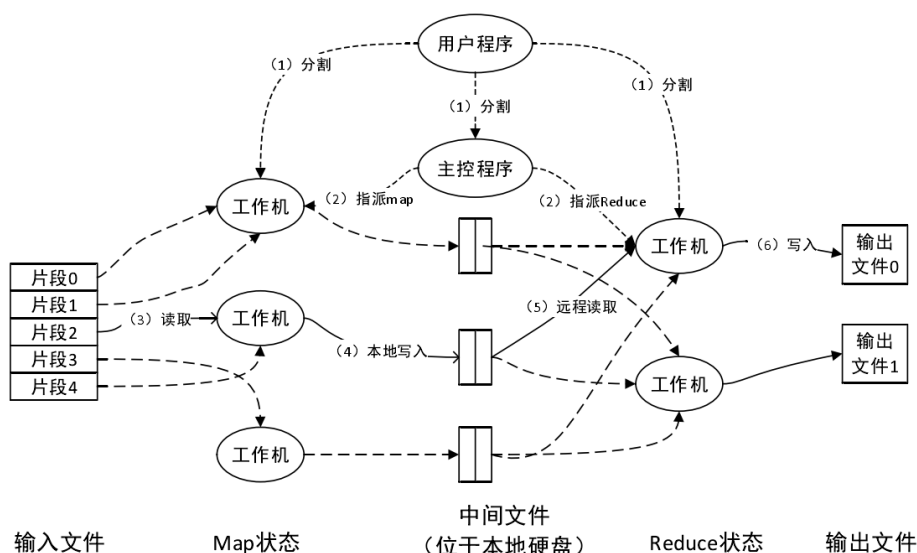


Figure 2: MapReduce 原理图

年 Spark 就诞生了，它作为一种新的集群计算系统，以其飞快的速度迅速就引起了研究和应用的热潮，它提供 Java, Scala, Python 和 R 中的高级 API，以及支持通用执行图的优化引擎。它还支持一组丰富的更高级别的工具，包括 Spark SQL 用于 SQL 和结构化数据的处理，MLlib 机器学习，GraphX 用于图形处理和 Spark Stream 用于流式数据的实时处理。

Spark 用到了一种被称为 RDD (Resilient Distributed Datasets) 的数据集，这是一种弹性分布式数据集，Spark 拥有丰富的 RDD 算子可以让用户像处理本地数据一样处理 Spark 大数据，因为 RDD 算子的丰富性可以让其处理大数据的方式更为灵活，并且避免了冗余的 MapReduce 操作，和冗余的磁盘 IO 通过缓存到内存，多线程执行等操作保证了 Spark 运算的速度。Spark 是在 Scala 语言中实现的，与 Hadoop 不同，Spark 和 scala 可以紧集成，scala 可以像本地收集对象一样轻松地操作分布式数据集。虽然 Spark 是为支持分布式数据集上的迭代作业而创建的，但它实际上是 Hadoop 的补充，可以在 Hadoop 文件系统上运行。

Spark 计算模型具有速度快、使用方便、统一分析引擎的优势。

### 3 基于大数据平台的无人机调度技术研究

无人机作为技术成熟、性能稳定的工具，服务越来越智能、高效、精准，而当无人机与物联网、与农业有效融合时，无疑将带来数据采集及应用的巨大突破。目前而言，全美 65% 的化学农药采用飞机作业完成喷洒，年处理 40% 以上的耕地面积；其中水稻施药作业 100% 采用航空作业方式；日本有 60% 的水稻田采用无人机飞防来完成防治的，近 40% 的农田植保防治是用无人机来完成的。中国耕地面积约为 1.33 亿 hm<sup>2</sup>，按照 40% 植保作业由植保无人机完成，则有 0.53 亿 hm<sup>2</sup> [4-5]。植保无人机在 2016 年以来，整体市场和产品可靠性都得到长足的发展。从 2018 年的态势来看，整个行业的销量有不会低于 3 万台。这样 1 个数字已经极大地超越了目前政府可能补贴的极限。这意味着整体用户市场开始接受植保无人机作为 1 个作业工具，承认它的效果和效率。随着无人机的普及，无人机在农业中也得到了广泛的应用，逐渐暴露出以下问题：一是现在平台没有整合气象、交通、维修等信息，没有专门为农户、飞服组织提供的个性化信息服务。飞服公司没有 1 个合适的平台联系种植户，出现了“找活难”的现象。二是信息不对称，在作业过程中，面对海量的农田作业信息，对于飞服组织而言，通常都是根据主观经验对选择的农田作业点进行调配，依据经验或盲目地选择作业农田。由于没有科学的调配方案，常常出现成本过高、收益较少的问题。

在利用农业机械提升工作效率方面，有许多学者在不同的方面开展了广泛的研究。文献 [19] 提出

了 1 种基于机主选择的农机调配模式，建立了以高收入低成本为目标的优化模型，实现了基于启发式优先级规则的农机调配算法。文献 [ 20 ] 提出了跨区域作业问题中带时间窗的农机调度问题，提出了基于智能遗传算法的农机调度策略。本研究是在张璠的农机跨区作业紧急调配算法适宜性选择的基础上 [ 21 ]，提出了基于回溯策略的单机多任务的无人机调度算法。在无人机不足时，基于最短距离优先的紧急调配算法性能更加有优势。在此算法基础上，结合回溯策略，找到无人机作业的最优规划路线，是本研究的核心。实现了飞手和种植户的供需无缝对接，为飞服组织提供最优益的调配方案，具有重要的现实意义和应用价值<sup>[23]</sup>。

### 3.1 系统的总体架构

平台自顶向下分为用户层、访问层、平台层、网络传输层、数据采集层 5 层结构。平台整体结构如3所示。

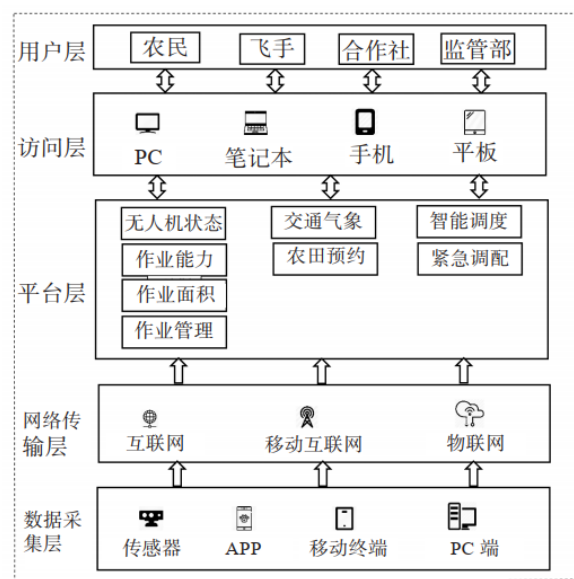


Figure 3: 系统结构图

**数据采集层：**大数据平台的数据采集来源主要有飞手和农户自主发布的信息（比如作物类型、作业时间窗、作业报价、面积等）、机载传感器向服务器传送的数据（比如无人机的位置等）、地图和天气接口获取的相关信息。机载传感器采集的数据要求实时性强的数据适合采用物联网的方式自动获取。

**网络传输层：**基于有无人机作业的地点不固定，环境多变等特性，有一些数据要求实时性较高，所以将移动互联网作为机载数据传输方式是最适合的。**平台层：**在平台运行过程中，会产生巨大的数据量，Spark 是专业用于处理分布式存储的大数据工具，也非常适用于数据的低延迟访问以及对性能和时间要求高的场景。用户可以使用多种不同的终端设备与大数据平台进行交互。

**访问层：**由于终端的多样性，使用 Web、App 等技术实现用户界面和数据的展示。

**用户层：**根据用户不同的需求，设定不同的角色，不同角色使用不同的 App。“我是地主”使用对象是农民，“我是飞手”使用对象是飞服公司。

### 3.2 数据采集

#### 3.2.1 机载数据单元采集数据

数据采集的方式有 2 种，分别是通过机载数据单元和客户端实现的。机载数据单元由 GPS、作业监控等若干传感器模块组成，采集来自北斗 /GPS 的无人机的地理位置信息、作业情况等信息，通过采集单元汇聚后经过移动无线网络发送到平台的接收端，采集到的数据发送到智慧平台的数据库

### 3.2.2 客户端采集数据

客户端分为  $B/S$  和  $App$  2 种方式, 目的是为了满足不同用户在不同的使用环境下的需求。农民通过“我是地主”  $App$  发布待作业的农田信息, 包括: 单价, 地块面积等, 设定农田中心点, 查询满足一定范围内的周边可用的无人机进行作业。

### 3.3 调度算法

智能调配要在时间期限内做出科学合理的决策, 规划出合理的调配方案。无人机利用移动互联网、北斗定位系统和物联网等信息技术, 为飞手和地主提供智能决策与信息化服务, 提供收益最高的方案规划。智能调配要在时间期限内做出科学合理的决策, 规划出合理的调配方案。无人机利用移动互联网、北斗定位系统和物联网等信息技术, 为飞手和地主提供智能决策与信息化服务, 提供收益最高的方案规划, 相关算法可查看文献[23]。

为了进一步验证算法的性能, 该文献模拟了 5 组无人机调度数据, 通过基于回溯策略调度算法计算出平均每台无人机在作业季内收益为 3 811.22 元。经过 5 次随机反复测试, 随机调度计算得出平均每台无人机在作业季内收益为 3 173.372 元。由此可见, 以飞手为服务对象, 平均每台无人机在作业季内预计可提高收益 20.1%。

该文献最终得出的结论有:

- (1) 基于互联网的无人机调度平台利用机载数据单元以及手机客户端等多种终端, 提供线上线下服务, 实现飞手和种植户的供需无缝对接;
- (2) 基于回溯策略的单机多任务无人机调度算法为飞服公司提供了 5 个最优候选方案, 并按收益最大、成本最低依次排列, 与传统的随机调度相比, 在作业季内每台无人机收益可预计提高 20.1%

## 4 基于气象大数据的天气预测的研究

随着社会的发展, 人们的出行方式也从步行, 骑马, 自行车, 汽车, 火车, 高铁, 飞机等不断演进。在交通运输领域目前除了高铁之外发展最为迅猛的就数无人机了, 比如近年来京东、顺丰已经开始进行无人机送货的尝试了。无人机相对于有人驾驶的飞机有很多先天性的优势。有人驾驶的飞机必须配备生命系统, 仪表盘、供氧系统、通讯、空调等, 无人机只需配备自动驾驶系统和通讯系统, 结构简单、制造成本低。另外, 由于精简了飞机上的设备, 使得重量降低, 核载率更高, 运营成本更低。

对于偏远山区隔山跑死马, 无人机有着先天的优势。无人机运输要解决的一个核心问题就输无人驾驶问题, 而解决无人驾驶问题的一个关键问题是规划无人机的飞行路径。本课题<sup>[24]</sup>将聚焦在未来无人机飞行时可能遇到的路径规划问题作为研究对象。天空中没有实体化的路线和标志, 而且对于飞机来说, 对飞行影响最大的因素就是天气根据雷一钧的《中国飞机失事统计》一文可以看到从事故中原因记载统计出: 不详 3 起, 人为失误 7 起, 恶劣天气 5 起, 飞机故障 5 起, 如图4所示。

可以看出恶劣天气对飞机坠毁的影响仅次于人为失误, 排名第二位, 如果使飞行自动化, 并避开恶劣天气, 这样就解决了第一和第二两大飞机失事因素, 极大的增强飞行的安全性。因此我将研究如何规划无人机航线躲开天气恶劣的地区使无人机能安全快速的到达目的地。基于现有气象大数据, 通过机器学习处理的方法准确预测天气, 在此基础上规划无人运输飞行器的最佳路线, 这将对未来无人机运输有重要意义。

### 4.1 数据处理

实验用到的数据来自天池大数据上的“未来已来——无人机路径优化大赛”的比赛数据。赛事举办方称这些数据是来源伦敦气象局的真实数据, 只是进行了脱敏处理。提供的是 5 天的 20 多个 G 的 10 个模型的风力预测数据和相应的真实天气数据。设计思路如下:



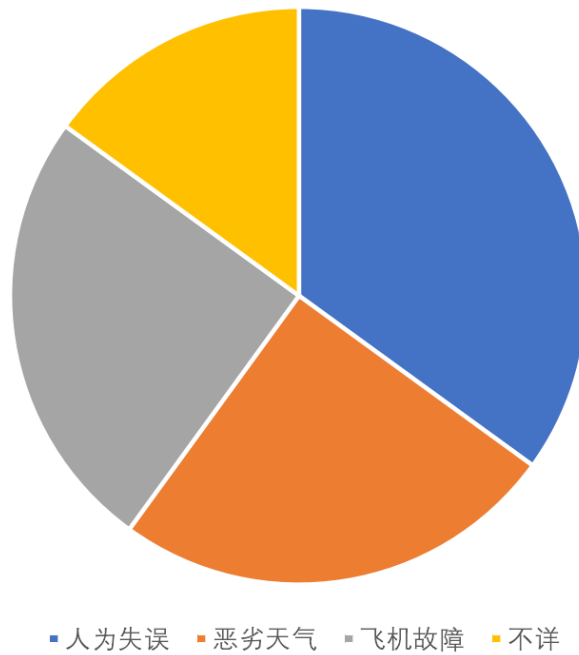


Figure 4: 失事原因

首先需要对原始的预测模型数据进行数据清洗，主要是一些缺失值和异常值的清洗，该文中我们用中值滤波器对数据处理，以消除数据中的缺失值和异常值的干扰。对于特征的提取主要是根据数据的实际意义，在原数据进行一番处理，得到新的特征，一般情况下原数据的统计信息、方差等都有可能成为新的特征。最终我们用到的特征有：原始数据特征、周围天气特征（上下左右）前后时间天气特征（前一个小时和后一个小时）、时间特征（离散特征），平均值特征；其次就是在数据清洗之后进行分析。

最后进行预测，预测方面可以有两种做法分类和回归。因为在最终规划路径的时候，我们会把区域上的点，分为安全区域和危险区域，所以只要保证安全区域的点是安全的飞机可在安全区域自由航行，也就更有可能找到更短的路径。但是做回归的话虽然可能找的安全区域系数更精确但在后续的路径规划的时候无人机飞行可能会因为过度考虑飞行路线的安全性而绕了很远的路这样反而使油耗增加，而且在空中呆的时间越久，那么在空中遇到危险天气的可能性也就越大，有点得不偿失。在进行预测时我们还不得不考虑一下样本均衡问题，例如如当一天的天气很好时大风区域会很小，或者某几天都是大风天，这样大风区域会很多，所以对于这样的天我们采用随机抽样的方法，来使用样本更均衡。综合考虑这些因素，为了更好的进行路径规划我们用二分类的方法作预测。

最后用数据结合各大预测模型，选取其中最优的几大预测方法得到的预测 F1 值（假定危险点为正例）对比

## 4.2 处理流程

本实验包括两部分，第一部分是进行天气预测，第二部分是进行无人能及航迹规划，两者以前以后前者是后者的铺垫和前提。实验的整体数据处理流程图如图5所示。

由于本实验的均是在虚拟机上完成的，搭建的大数据平台的单机内存受到本机内存的限制故而没有像真正的大数据平台那样拥有超大内存。所以本次实验并没有使用全量的数据，只使用了从全量数据集中截取的 400 多兆的数据。

该文献把程序打包并提交集群任务和单机任务进行对比实验，大数据集群对解决大数据运算的优势是可扩展性非常好，如果处理的数据量增大，需求的时间短，增加相应的机器到集群中便增加了集群的算力。对于无人机的路径规划算法，在基于传统 BFS 算法改造而得的时变的无人机路径规划算法和在此基础上结合 A\*算法添加路径评估函数，进一步改进而得的改进 BFS 算法和传统 BFS 进行比较，发现在



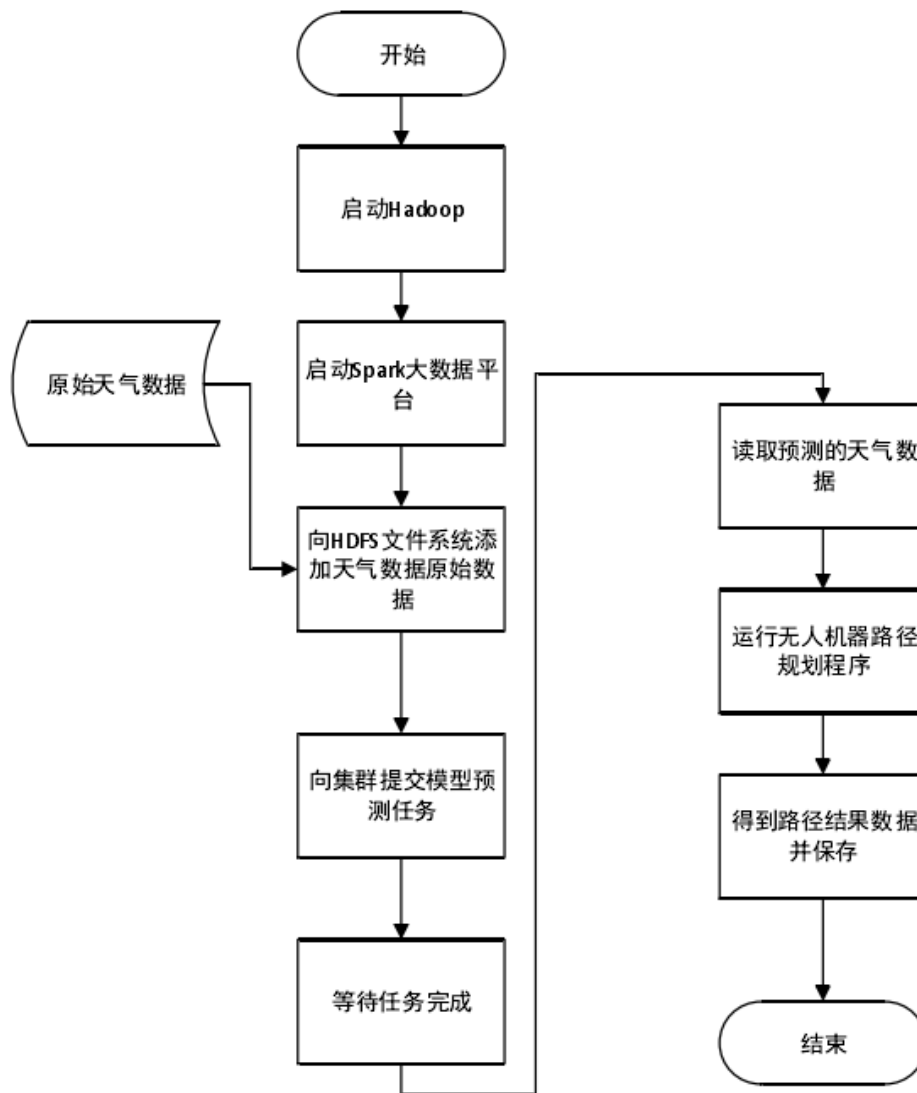


Figure 5: 数据处理整体流程图

预测的准确率上面有了很大的提高。当然改进算法也有一点不足，在规划出路径时用的时间有了一定的延长，但牺牲一点时间性能来获得路径代价的大幅降低是值得的。通过实验验证了该天气预测模型和路径规划算法的可行性与实用性和有效性。

## 5 结论

本文首先关于 Apache Hadoop, Spark, Hbase、Storm、Pig、Hive、Mahout 等技术进行一定的介绍，其次主要就文献[23-24]展开详细的叙述，文献[23]主要介绍了大数据平台的无人机调度技术在农业方面的研究，通过某种方式获取数据，输入到特定的算法内部借用大数据平台进行数据分析演算，从而可以验证该算法可以使得收益提高20.1%，尽述了大数据平台的使用；文献[24]主要介绍了基于气象大数据的天气预测，虽然已经有了很多研究方法可以借鉴，但每个方法都有它自身的局限性，故希望能综合使用各种预测方法，让各种预测方法取长补短，使最后的预测结果更加符合本课题的需求。天气预测一般包括风力、温度、湿度等多项指标的预测，由于文献[24]主要研究的天气数据时要服务于无人机的路径规划，该无人机在空中飞行时影响其飞行的主要因素是风力，所以这里主要预测的是风力数据，最后通过在数据集上应用该天气预测算法和路径规划算法并从时间效率和飞行代价等方面对实验结果进行了对比分

析，得出了可观的结论。

大数据平台在无人机领域的研究远远不止这两个方面，依然存在着很多大量的方向可以使得大数据和UAV更好地结合，对此需要我们不断深入学习，探索挖掘新思路。

## 参考文献

- [1] Foulds L R, Wilson J M. Scheduling operations for the harvesting of renewable resources [ J] . Journal of food engineering, 2005,70(3):281-292.
- [2] Basnet C B, Foulds L R, Wilson J M. Scheduling contractors' farm-to-farm crop harvesting operations [ J] . International Transactions in Operational Research. 2006,13(1):1-15.
- [3] Alio, Van Oudheusden D. In field logistics planning for crop harvesting [ J] . Engineering optimization, 2009, 41(2): 183-197.
- [4] Foulds L R. hay harvesting operations scheduling subject to probabilistic activity duration and machine failure [ J] . Journal of Agricultural studies. 2004, 41(2): 183-189.
- [5] 董玉玲. 浅谈植保无人机推广中存在的问题及对策
- [6] . 新疆农垦科技, 2016, 39(7):27-28.
- [7] 刘科, 马根众. 山东省农用植保无人机应用现状及建议 [ J] . 山东农机化, 2017(3):26-27.
- [8] 孔令亮, 陆金晶. 农业植保无人机发展现状浅析 [ J] . 江苏农机化, 2018(5):43-45
- [9] 胡善君. 浅谈计算机在植保无人机系统中的应用 [ J] . 科技创新与应用, 2017(13):49.
- [10] 蔡银杰, 孙娟, 丁晓辉, 等. 我国植保无人机发展现状与展望 [ J] . 世界农药, 2018,40(6):15-18+36.
- [11] 胡冲. 农机调配可视化系统研究 [ D] . 保定: 河北农业大学, 2012.
- [12] 王春山, 张璠, 滕桂法, 等. 智慧农机调配管理平台设计与实现 [ J] . 中国农机化学报, 2018,39(1):61-68.
- [13] 张璠. 农机调配策略研究 [ D] . 保定: 河北农业大学, 2012.
- [14] 王国新, 宁汝新, 王爱民, 等. 基于仿真的调度规则组合决策研究 [ J] . 北京理工大学学报, 2006,26(7):38- 41+69.
- [15] 易平, 李建军, 熊禾根. 基于多级优先规则的模具企业动态车间作业调度算法 [ J] . 模具工业, 2009,35(5):7-9.
- [16] 郭鹏程. 植保无人机调度平台的设计 [ J] . 山西科技, 2018,33(1):74-76.
- [17] 张燕, 滕桂法, 王春山, 等. 基于 Android 手机的农机调配系统 [ J] . 农机化研究, 2014,36(1):214-218.
- [18] 胡冲, 吴焕瑞, 滕桂法, 等. 农机调配多目标综合决策检验 [ J] . 农机化研究, 2013,35(3):46-49.
- [19] 周三琦. 农村信息化中农机调配策略及调配算法研究 [ D] . 郑州: 河南大学, 2016.
- [20] 张璠, 滕桂法, 马建斌, 等. 基于启发式优先级规则的农机调配算法 [ J] . 农业工程学报, 2012,28(10):78-85.
- [21] 王雪阳, 苑侗侗, 苑迎春, 等. 带时间窗的农机调度方法研究 [ J] . 河北农业大学学报, 2016,39(6):117-123.
- [22] 张璠, 滕桂法, 苑迎春, 等. 农机跨区作业紧急调配算法适宜性选择 [ J] . 农业工程学报, 2018, 34(5):47-53
- [23] 王昀, 滕桂法. 基于大数据平台的无人机调度技术研究[J]. 河北农业大学学报, 2019,42(6):134-138. DOI:10.13320/j.cnki.jauh.2019.0134.
- [24] 李建平. 基于气象大数据的无人机航线规划研究与应用[D]. 北京工业, 2019