

# 有关智慧农场的算法及研究

——基于 Python 算法下智慧灌溉程序设计

物理科学学院

李睿熙（组长）、刘众之、王楷震、易恩畅、郑宗翰

**摘要：**随着农业现代化的不断推进，如何高效合理地利用水资源进行灌溉已成为提高粮食产量和实现农业可持续发展的关键问题。本文设计了一种基于任务调度的灌溉优化算法，旨在合理安排灌溉任务，优化水资源分配，并考虑任务的持续性，确保每个灌溉任务在适当的时段完成，从而提升农业生产效率。

**关键词：**智慧农场、精准灌溉、节水灌溉、粮食安全

## 一、绪论

粮食安全被视作“国之大者”，乃国家安全的重要基石，关乎国民经济发展、社会稳定以及国家自立，属于全局性的重大战略问题。习近平总书记指出：“保障国家粮食安全是一个永恒的课题，任何时候这根弦都不能松。”然而，我国粮食需求持续攀升，供求紧平衡状态愈发紧张，国际形势复杂严峻，确保粮食安全这根弦必须始终紧绷。要保障粮食安全、牢牢端稳中国饭碗，就必须加速转变农业发展方式，大力推进农业现代化，坚决守住耕地红线。

我国水资源时空分布极不均匀、水旱灾害频发。我国独特的地理条件和农耕文明决定了治水对中华民族生存发展和国家统一兴盛至关重要。现在，我们国家不断推进城镇化进程，对水资源需求很大。水资源短缺已经成为了经济社会发展面临的严重安全问题。习近平总书记强调，推进中国式现代化，要把水资源问题考虑进去，以水定城、以水定地、以水定人、以水定产，从实现长治久安的高度和以对历史负责的态度做好这方面工作。

根据我国的粮食安全问题与水资源匮乏问题，本小组为提高粮食单产，节省耕地面积与水资源，设计了“智慧农场”节水算法。

## 二、信息调研

项目需要基于对相关农业数据的分析与建模。因此，数据的搜集和相关文献的查找成为了项目的重要组成部分。我们从国家统计局以及其他权威数据来源入

手，搜集了粮食产量、耕地面积和人口数量等关键信息，并利用这些数据进行程序设计和模型训练。

## （一）数据搜集

为了使模型更贴近实际，我们对多个数据源进行了系统的搜集和整理：

### 1. 粮食产量数据

我们首先前往国家统计局官网，通过检索功能获取了我国近五年粮食产量的条形图。通过进一步查阅历年发布的公告，我们整理出 2014 年至 2023 年的粮食产量数据。这些数据为后续的农业生产模型提供了基础支持。

### 2. 耕地面积数据

在对耕地面积数据的搜集过程中，我们发现国家统计局并没有按年发布详细的耕地面积数据。通过进一步检索，我们发现耕地面积的相关数据主要集中在每十年一次的全国农业普查中。根据第二次和第三次全国农业普查的数据，我们获得了 2006 年、2016 年和 2022 年的耕地面积数据，分别为 121775.9 千公顷、134921 千公顷和 12760.1 万公顷。虽然近年没有较大变化的单独统计，但这些数据为模型的土地利用部分提供了重要依据。

### 3. 人口数据

为了更好地模拟智慧农场的劳动力需求和人口资源配置，我们从第六次和第七次全国人口普查中提取了 2010 年和 2020 年全国人口数据，分别为 13.4 亿和 14.4 亿。这些数据为模拟农业劳动力分布和未来趋势预测提供了基础。

## （二）文献查找

在文献查找过程中，我们主要通过互联网搜索引擎（如 Bing）和学术资源库（如中国知网、国家统计局官网）获取相关的研究资料和政策文件。以下是我们参考的主要文献和数据来源：

### 1. 国家统计局官方网站

为获取粮食产量、人口普查和相关农业数据提供了最为权威的来源。

### 2. 自然资源部网站

帮助我们获取了耕地面积的相关数据，特别是关于农业普查中的统计信息。

通过这些权威的文献和数据，我们能够更精准地定义项目需求，并确保程序模型的设计基于可靠的现实数据。

### （三）总结

在数据搜集与文献查找的过程中，我们主要依靠国家权威机构发布的数据，确保了项目的准确性和真实性。通过对粮食产量、耕地面积和人口数据的综合整理，我们能够智慧农场项目提供充分的数据支持，并利用这些数据对模型进行有效的训练和调整。

此部分的数据搜集为程序模型的构建打下了坚实的基础，确保模型能够准确反映实际的农业生产和劳动力情况。

### 三、 数据分析

为了具体确定我们小组作业的选题与程序的实际应用方向，我们首先应用了 Excel、Python 等软件对搜集到的数据进行了处理，并得出了相应结论。

#### （一）全国的粮食产量数据

我们将全国的粮食产量数据导入 Excel 表格，以便后续的进一步分析。

地区	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
全国总计	112738.3	113340.5	113028.2	112219.6	117037	116064	116768	117631.5	118332.1	118968.5
北京	120.2	104.5	87.3	66.8	56	47	49	60.9	76.7	89.5
天津	345.8	350	357.3	351.2	350	339	350	373.5	376.7	390
河北	6332	6392.5	6327.4	6190.7	6539	6469	6389	6428.6	6443.8	6455.2
山西	3286.4	3287.2	3241.4	3204.4	3137	3126	3130	3138.1	3150.3	3161
内蒙古	5651	5726.7	5784.8	5757.8	6790	6828	6833	6884.3	6951.8	6984.7
辽宁	3235.1	3297.4	3231.4	3227.2	3484	3489	3527	3543.6	3561.5	3578.4
吉林	5000.7	5078	5021.6	5023.3	5600	5645	5682	5721.3	5785.1	5825.6
黑龙江	11696.4	11765.2	11804.7	11827.1	14215	14338	14438	14551.3	14683.2	14743.1
上海	164.9	161.9	140.1	118.7	130	117	114	117.4	122.8	127.2
江苏	5376.1	5424.6	5432.7	5406.4	5476	5381	5406	5427.5	5444.4	5458.9
浙江	1266.8	1277.8	1255.4	1282	976	977	993	1006.7	1020.4	1024.7
安徽	6628.9	6632.9	6644.6	6642.5	7316	7287	7290	7309.6	7314.2	7334.5
福建	1197.7	1193.2	1176.7	1179.4	834	822	834	835.1	837.6	841.1
江西	3697.3	3705.6	3686.2	3667.4	3721	3665	3772	3772.8	3776.4	3774.3
山东	7440	7492.1	7511.5	7447	8405	8313	8282	8355.1	8372.2	8387.9
河南	10209.8	10267.2	10286.2	10135.5	10906	10735	10739	10772.3	10778.4	10785.3
湖北	4370.3	4466	4436.9	4471.7	4847	4609	4645	4686	4689	4707
湖南	4975.1	4944.7	4890.6	4862.4	4748	4616	4755	4758.4	4765.5	4763.5
广东	2507	2505.8	2509.3	2500.1	2151	2161	2205	2213	2230.3	2229.5
广西	3067.7	3059.3	3023.6	2976.2	2802	2747	2806	2822.9	2829.3	2834.7
海南	417	375.6	360.4	348.3	286	273	271	271.4	273	273.6
重庆	2242.5	2234	2250.1	2239	2018	1999	2003	2013.2	2046.7	2025.9
四川	6467.4	6453.9	6453.9	6441.4	6266	6279	6313	6357.7	6463.5	6404
贵州	3138.3	3114.9	3113.3	3051.2	2740	2709	2754	2787.7	2788.7	2773.8
云南	4508.2	4487.3	4481.2	4446.1	4175	4166	4167	4191.4	4211	4243.2
西藏	169.3	176.6	176.6	184.8	183	184	182	187.2	192.6	194.6
陕西	3076.5	3073.5	3068.7	3045.3	3006	2999	3001	3004.3	3017.5	3023
甘肃	2842.5	2849.6	2814	2782.5	2645	2581	2638	2676.8	2699.8	2710.9
青海	280.1	277.1	281.1	278.6	281	280	290	302.4	303.5	304.9
宁夏	771.3	770.4	778.3	775.6	736	677	679	689.3	692.3	693.9
新疆	2255.9	2395	2401.1	2289.1	2220	2204	2230	2371.7	2433.9	2824.8

图 3.1：全国粮食产量（单位：万吨）

数据来源：国家统计局：<https://www.stats.gov.cn/sj/zxfb/>

由图表可知，从 2014 年到 2023 年，全国粮食总产量呈现稳步增长的趋势。

#### （二）全国耕地面积数据

将耕地数据面积获取后，我们发现数据数量级较大，并不利于研究与分析，因此，我们应用了 Excel 中的制图功能，将数据绘制为折线图。

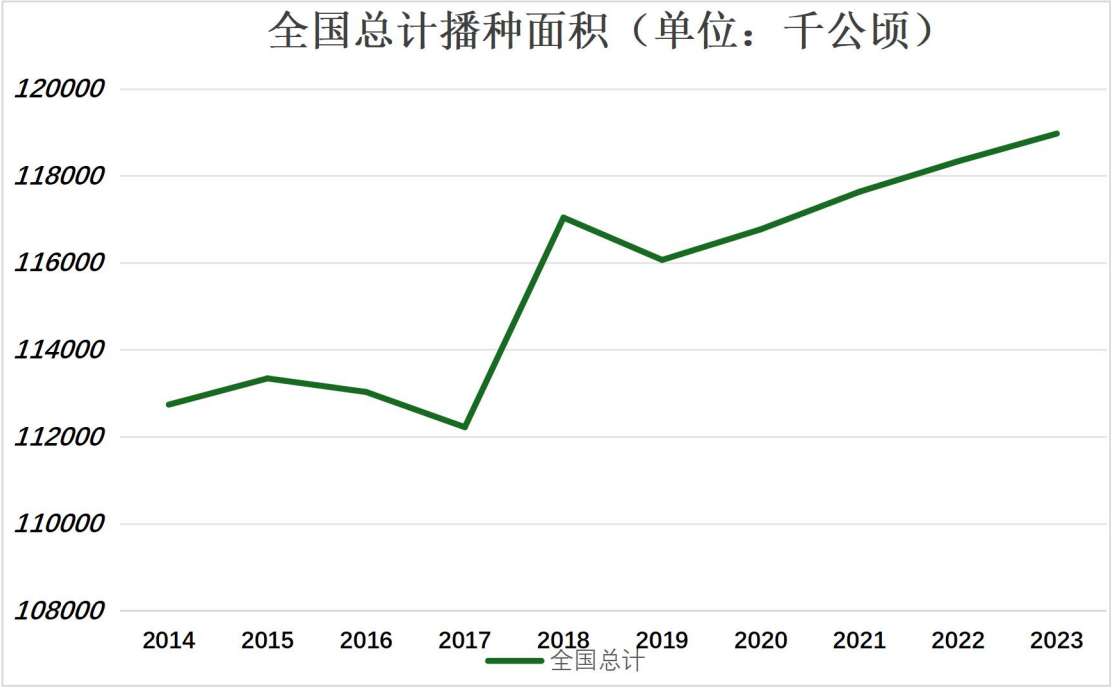


图 3.2：全国粮食播种面积变化

数据来源：[https://www.stats.gov.cn/sj/tjgb/nypcgb/qgnypcgb/202302/t20230206\\_1902100.html](https://www.stats.gov.cn/sj/tjgb/nypcgb/qgnypcgb/202302/t20230206_1902100.html)

由 Excel 所绘制的图表，我们可以很明显的发现，全国播种面积总体呈现波动上升趋势，表明农业种植面积在增加。

### （三）各地单产情况

由于各省的土地面积存在较大差异，仅仅应用播种总面积与粮食总产量并不能准确反映出各地粮食生产效率的准确情况。为此，我们应用 Python 编写程序（代码见附录），具体计算出各省的单产，以便进一步分析。

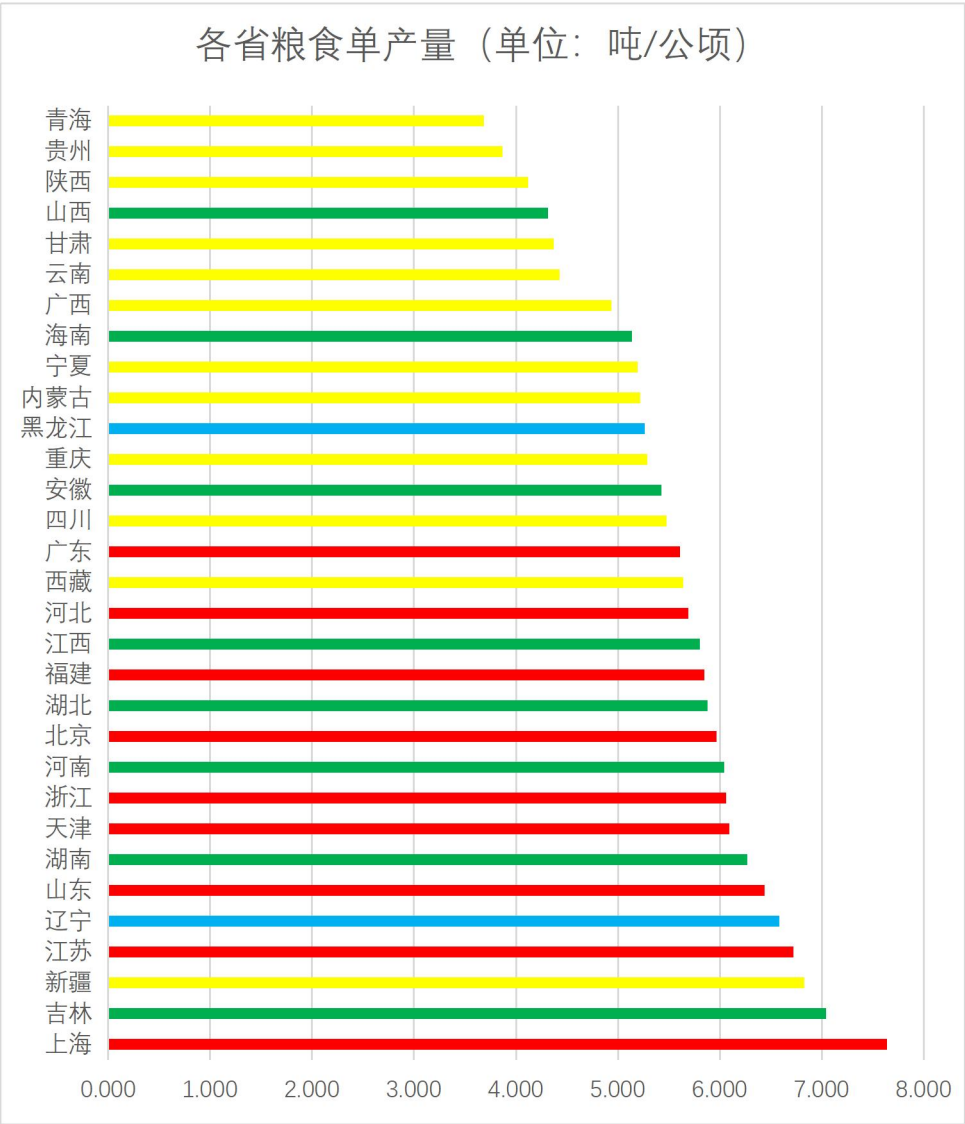


图 3.3：各省粮食单产量

图表来源：作者自制

在图表中，我们将不同地区的单产量使用不同的颜色表示（东部地区：红色；东北地区：蓝色；中部地区：绿色；西部地区，黄色）使数据更加可视化。我们可以明显发现，东部地区的粮食单产量要明显高于西部地区。

根据中国不同地区的粮食产量与种植面积数据，可以观察到明显的产量差异。中国的农业生产受地域、气候、水资源、土壤类型以及科技水平等多种因素的影响，各地区在粮食单产和产量上的差异体现了这些自然和人为因素的综合作用。

从上述分析可以看出，灌溉效率直接影响粮食单产。在一些水资源匮乏的地区（如西北地区），虽然种植面积扩大，但由于灌溉技术落后，水资源利用率低，导致产量未能显著提升。而在水资源相对丰富的地区（如长江流域），虽然总体

产量较高，但由于灌溉技术的滞后，部分地区依然存在资源浪费和产量提升缓慢的问题。

（四）未来粮食产量的预测

从国家统计局的官网上可以得到 2013 至 2023 年的全国粮食总产量，本文以表格形式展现。为了方便表示与计算，将 2013 年用年份代号 1 来表示，后续年份代号依次增加 1。（产量的单位是万吨）

年份	1	2	3	4	5	6	7	8	9	10	11
产量	60193	60709	62143	61623	61791	65789	66384	66949	68258	68653	69541

（表一）

对于全国粮食年产量  $y$  随年份  $x$  有明显的线性关系，于是可以进行一元线性拟合。不妨假设  $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ 。我们采用最小二乘法进行拟合。根据最小二乘法得到的回归直线能使距离平方和达到最小。我们就要使  $Q = \sum (y_i - \hat{y})^2 = \sum (y_i - \hat{\beta}_0 - \hat{\beta}_1 x)^2$  的值达到最小值。

令  $\frac{\partial Q}{\partial \hat{\beta}_0} = -2 \sum (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0$ ,  $\frac{\partial Q}{\partial \hat{\beta}_1} = -2 \sum x_i (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0$ ，解得：

$$\hat{\beta}_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

经过计算机的运算，我们不难根据表一的数据得到  $\hat{\beta}_1 = 1019.80, \hat{\beta}_0 = 58614.20$ 。

于是我们可以得到拟合直线的方程： $\hat{y} = 1019.80x + 58614.20$ 。

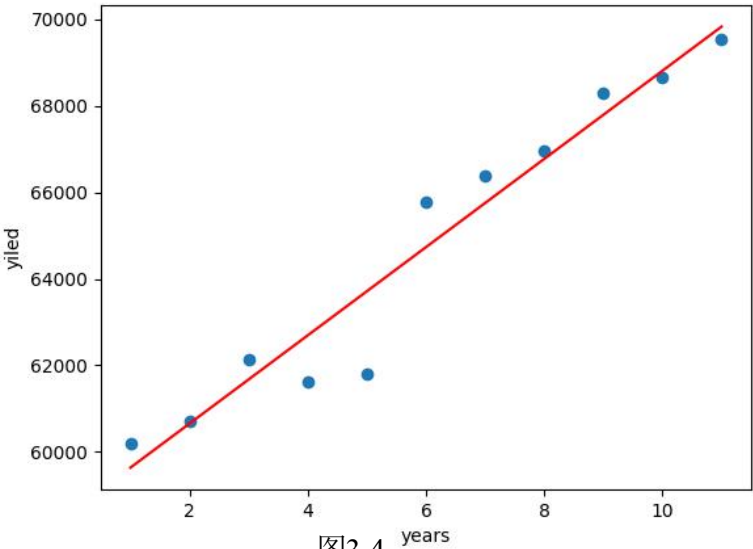


图3.4

上图 3.4 将拟合直线与散点图相结合，发现我们的估计较为精准。进一步地，考虑反映拟合有效程度的指标决定系数  $R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y}_i)^2} = 0.940224$ ，说明我们的拟合是比较到位的。

我们可以根据拟合的直线，进行相关的估计，例如：要估计 2024 年的粮食总产量，假设没有外界干扰因素，可以将  $x=12$  代入式中得到  $y=70851.8$  万吨。注意到，年份 4 与年份 5 数据出现了明显异常，我们可以进一步剔除这两年的数据，同上面的操作一样，得到更精确的结果，但本文不做进一步的运算。

## （五）结论

通过对中国不同地区粮食产量、科技水平与自然条件的分析，可以得出以下结论：

### 1. 地域差异导致产量差距

中国各地区的粮食单产差异较大，主要受到水资源条件、气候差异和农业科技水平的影响。特别是华北、黄河流域和西部地区，水资源紧张或分布不均，灌溉效率低下，限制了粮食单产的提高。

### 2. 智慧灌溉技术潜力巨大

智慧灌溉技术可以有效提升灌溉效率，减少水资源浪费，优化水分分布，在水资源匮乏的地区（如华北地区）。其应用不仅能提高粮食产量，还能推动农业科技的升级和土地的可持续利用。

### 3. 科技应用和灌溉管理提升产量

在农业科技相对落后的地区，如西部地区，传统灌溉方式和技术的限制导致产量低下。智慧灌溉可以作为一个关键技术，帮助这些地区提高水资源利用率，推动产量提升。

## 四、 算法分析

在大型农场的管理中，灌溉是非常重要的一环。而基于以上的数据，我们决定运用 Python 的自定义函数实现模块化、综合运用的灌溉算法，即我们只是提供了一种方案，具体的操作需要传感器的帮助和实践；因为自动灌溉装置当然需

要 IOT（物联网）模块，由于时间技术等各方面原因，我们无法实现利用传感器的方式实现自动灌溉，因此本文设计了一种简易的灌溉算法。

在大型农场的管理中，灌溉是非常重要的一环。因为自动灌溉装置当然需要 IOT（物联网）模块，由于时间技术等各方面原因，我们无法实现利用传感器的方式实现自动灌溉。而基于以上的数据，我们决定设计一种节水的灌溉方案，并运用 Python 程序给出灌溉时间与灌溉模拟。

（一）灌溉方案

设想一个有  $n$  块地的大型农场，每块地都种植了不同的粮食或经济作物。由于品种的差异，每种作物的需水量和灌溉时间也有所不同。因此，合理安排灌溉任务，既能保证作物的正常生长，也能节约用水和能源。在实际操作中，灌溉任务往往并非连续进行。例如，同一个任务可以在其规定的时间范围内分段完成。为了便于表示，我们将时间平均分为  $k$  个时间段，用一个整型数字来标识具体时间。将每个地块相伴的三元组定义为  $[x,y,z]$ ： $x$  为开始灌溉的时间点， $y$  为结束灌溉的时间点， $z$  为该任务所需的总灌溉时间单位。如  $[2,4,1]$  表示可以在时段 2~3 与时段 3~4 中任选一个进行灌溉。农场主需要根据不同地块种植的作物特点，通过统计分析来获取灌溉需求的相关数据。例如，假定农场主已经用 excel 表的形式将各地块所对应的三元组整理好。

地块	开始	结束	持续时间
1	2	4	1
2	4	7	1
3	3	7	3
4	6	9	2

（表 4.1）

灌溉需遵循以下原则：

原则 A. 任务重叠处理：若当前任务与前一个任务的灌溉时间段有重叠，则优先利用重叠时间完成灌溉需求。例如：任务 1 在时间段  $[2, 4]$ ，任务 2 在时间段  $[3, 7]$ ，那么时间段  $[3, 4]$  可以同时为 1 和 2 提供灌溉。

原则 B. 剩余时间后置原则：对于当前任务在重叠时间段未能完成的部分，将其安排在尽可能靠后的时间段。这样可以尽量延迟对其他任务的影响。



状态记录与更新：每个时间段需要记录哪些地块正在灌溉，以及哪些地块的任务已经完成。我们将使用一个矩阵表示灌溉状态：0 表示未完成灌溉；1 表示已完成灌溉。

最终输出结果包括两个部分：

- ①灌溉计划表：在每个时段需要为哪些地块开启灌溉系统。
- ②灌溉完成状态矩阵：展示各地块的灌溉完成情况。

（二）算法流程

本代码（见附录）运用了栈的数据结构对于灌溉时间段进行存储，下面将阐述编程的相关思想与代码的相关流程。

本代码用栈 `temp` 来储存已安排灌溉的时间段。依次遍历输入的每一个三元组，通过如下方式新增栈 `temp` 中的元素。假定栈 `temp` 中已有安排的灌溉时间段（如图 4.2.1），栈顶元素分量 `temp[top][1]=t`，新增任务所相伴的三元数组为  $[t_1, t_2, T]$ ，指针  $a$  依次遍历 `temp` 中的时间段，直到出现如图 4.2.2 或图 4.2.3 的情况（新增任务的开始时间第一次不晚于已有任务的开始时间），而之前已用自定义函数 `rank` 对结束时间进行升序排序，可以保证新增任务的结束时间不早于最后一个已有任务的结束时间。此时由原则 A 可知，重叠部分是共同灌溉的。假设新增任务在除去重叠时间后仍有剩余时间  $\Delta t$ ，由原则 B，如图 3.2.4，若  $T - t \geq \Delta t$ ，则可取  $[T - \Delta t, T]$  作为新增灌溉时段；如图 4.2.5，若  $T - t < \Delta t$ ，取  $[t, T]$  后对其余空余时间段进行类似的讨论，并从  $\Delta t$  中减去相应的已灌溉时间得到的  $\Delta t^*$ ，直至  $\Delta t^* = 0$ 。下面更新栈 `temp` 的元素，将被“覆盖”掉的元素从栈顶弹出，并合成新的栈顶元素 `temp[top]`。

重复操作后，即可得到应灌溉的时间段 `temp`。

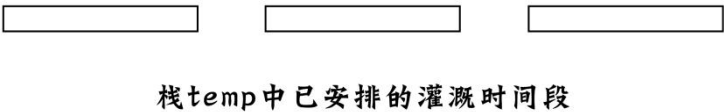


图-4.2.1

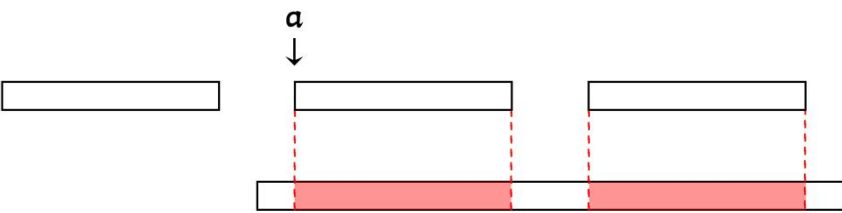


图-4.2.2 新增任务

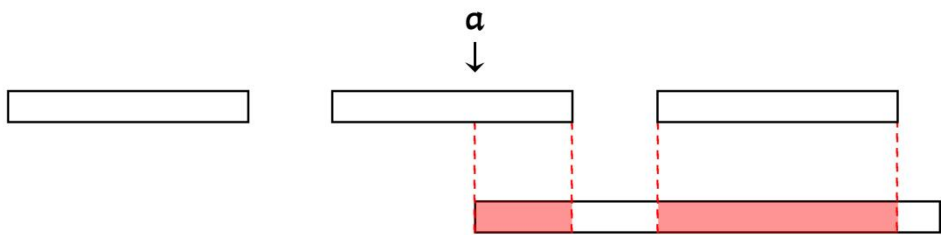


图-4.2.3 新增任务

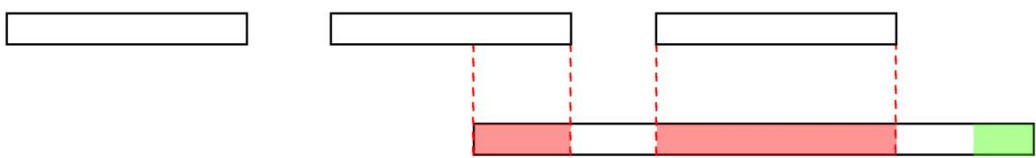


图-4.2.4

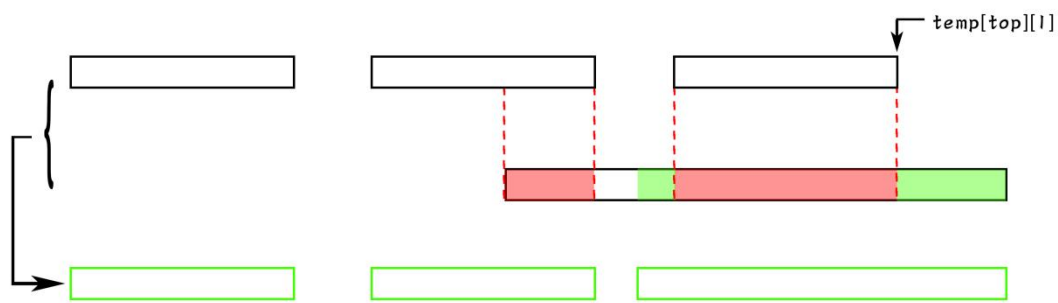


图-4.2.5

## 五、 反思与展望

在设计和实现上述灌溉算法时，我们的目标是通过高效的任务调度，合理分配有限的灌溉资源，以实现最优化的水资源管理。然而，尽管该算法能够在一定程度上解决灌溉任务的排期问题，仍存在一些缺陷，尤其是在农业应用场景中的局限性。同时，由于我们综合运用了数据处理与优化算法，因此我们的算法同样可以在其他领域中实现应用。

## （一）算法在农业应用中的缺陷

### 1. 无法结合天气状况等实际因素进行灌溉

该算法目前的设计仅依赖于任务的开始时间、结束时间和持续时间等基础参数进行排期调度。它假设每个灌溉任务都可以按计划执行，但忽略了一个重要的因素——天气状况。在实际农业灌溉中，天气变化（如降雨、温度变化等）对作物的水分需求有着至关重要的影响。如果没有考虑天气的变化，算法可能会导致水资源的浪费或作物因缺水而受损。因此，如果能加入天气传感器数据、土壤湿度监测等实时信息，算法的准确性和适用性将大大提高。

### 2. 对单一作物的农场适应性差

该算法的调度机制仅可以考虑不同作物对水分需求的差异。对单一作物而言，作物的水分需求量基本相同，并无法有效规划最佳灌溉方案。因此，在实际农业生产中，单一作物的农场可能无法很好地使用该算法。为了使算法适应更广泛的农业环境，可以根据地形与实际情况进一步调整算法与灌溉方式。

## （二）算法的推广应用

尽管该算法在农业灌溉中存在一些局限性，但其基本思想和框架在其他领域具有很好的推广潜力，尤其是在任务调度和资源优化的场景下。以下列举几个潜在的应用领域。

### 1. 生产与制造领域

在制造业中，生产任务通常有一定的开始时间、结束时间和持续时间。任务调度算法可以帮助工厂根据设备的空闲时间安排生产任务，以最大化生产效率并减少设备空闲时间。

### 2. 物流和运输管理

在物流行业，任务调度算法可以用来安排不同的运输任务，安排运输车辆的出发和到达时间，避免交通冲突和提高运输效率等任务都可应用该算法。比如，多个货物需要按时从一个仓库运送到多个目的地，任务调度可以优化每辆车的行程，减少空载行驶。

### 3. 项目管理

在大型项目管理中，通常有多个任务需要在特定时间段内完成。调度算法可以帮助项目经理安排各个任务的开始和结束时间，确保所有任务能够按照预定的

时间表进行，并且避免任务冲突。比如，建筑项目中的不同施工任务，需要根据资源可用性和优先级进行合理安排。

#### 4. 云计算和资源管理

在云计算环境中，多个虚拟机可能需要使用计算资源（如 CPU、内存等），调度算法可以帮助管理这些虚拟机的资源使用时间。通过合理的时间段分配，避免资源浪费，最大化资源的利用率。

#### 5. 能源调度

电力系统需要确保发电机在不同的时间段能够满足用电需求。调度算法可以帮助优化发电机的启停时间，确保电力系统的高效运行，同时避免过度发电或能源浪费。

## 六、 附录

### （一）参考文献

- [1]刘冕. 守住耕地保护红线筑牢粮食安全底线[N]. 南昌日报, 2024-06-25 (002).
- [2]张晶晶. 信息技术在农田水利工程节水灌溉中的发展措施[J]. 农机市场, 2024, (08):66-67.
- [3]国家统计局. (2023). 中国粮食产量数据. 国家统计局数据库. <https://www.stats.gov.cn/sj/zxfb/>
- [4]国家统计局. (2023). 人口普查数据. 国家统计局数据库. [https://www.stats.gov.cn/sj/zxfb/202303/t20230301\\_1919256.html](https://www.stats.gov.cn/sj/zxfb/202303/t20230301_1919256.html)

### （二）文中所用 Python 代码

#### 1. 最小二乘拟合直线代码

```
import pandas as pd
import matplotlib.pyplot as plt
path="F:\大学\作业\统计学导论\数据.xlsx"
df=pd.read_excel(path)
x=[1,2,3,4,5,6,7,8,9,10,11]
y=[]
for i in range(0,11):
    y.append(df.at[i,"全国粮食总产量"])
sxy=0
sx=0
sy=0
sx2=0
for i in range(0,11):
    sxy=sxy+x[i]*y[i]
    sx=sx+x[i]
```

```

sy=sy+y[i]
sx2=sx2+x[i]*x[i]
b1=(11*sxy-sx*sy)/(11*sx2-(sx)**2)
b2=sy/11-b1*sx/11
print(b1,b2)
z=[]
for i in range(0,11):
    z.append(b1*x[i]+b2)
fz=0
fm=0
ypingjun=sy/11
for i in range(0,11):
    fz=fz+(y[i]-z[i])**2
    fm=fm+(y[i]-ypingjun)**2
R2=1-fz/fm
print(R2)
plt.plot(x,z,color="red")
plt.xlabel("years")
plt.ylabel("yiled")
plt.scatter(x,y)
plt.show()

```

## 2. 计算各地单产代码

```

start=2014
suoyin=[]
while start<=2023:
    suoyin.append(start)
    start=start+1
import pandas as pd
import matplotlib.pyplot as plt
path1="F:\程序\粮食数据\各省份播种面积.xlsx"
path2="F:\程序\粮食数据\各省份年总产量.xlsx"
df1=pd.read_excel(path1)
df2=pd.read_excel(path2)
area=input()
lenth=31
num=0
for i in range(1,1+lenth):
    if df1.at[i,"地区"]==area:
        num=i
average=[]
for i in suoyin:
    temp=df2.at[num,i]/df1.at[num,i]
    average.append(temp)
plt.plot(suoyin,average)
plt.show()

```

## 3. 灌溉优化代码

```

def rank(lst):#按结束时间升序排列
    n=len(lst)
    for i in range(0,n-1):
        for j in range(n-1,i,-1):
            if lst[j][1]<lst[j-1][1]:
                lst[j],lst[j-1]=lst[j-1],lst[j]
    return lst
def rank_single(lst):
    n=len(lst)
    for i in range(0,n-1):

```

```

        for j in range(n-1,i,-1):
            if lst[j]<lst[j-1]:
                lst[j],lst[j-1]=lst[j-1],lst[j]
    return lst
def irrigate(task,k):
    n=len(task)
    temp=[[0,0] for i in range(n)]
    top=-1
    task=rank(task)
    schedule=[[[] for i in range (k+1)]
    for time in task:
        start=time[0]
        end=time[1]
        period=time[2]
        a=0
        while a<=top and start>temp[a][1]:
            a=a+1#升序排列保证 temp[a][1]>st[top][1]
        if a<=top:
            for i in range(a+1,top+1):
                period=period-(temp[i][1]-temp[i][0])
                for j in range(temp[i][0],temp[i][1]):
                    schedule[j].append(time[3])
            if start<temp[a][0]:
                period=period-(temp[a][1]-temp[a][0])
                for j in range(start,temp[a][0]):
                    schedule[j].append(time[3])
            else:
                period=period-(temp[a][1]-start)
                for j in range(start,temp[a][1]):
                    schedule[j].append(time[3])
        if period>0:
            t=end
            while t-temp[top][1]<=period and top!=-1:
                period=period-(t-temp[top][1])
                t=temp[top][0]
                top-=1
            top+=1
            temp[top]=[t-period,end]
            for j in range(t-period,end):
                schedule[j].append(time[3])
    return ([temp,schedule])#temp 中元素的格式是[开始, 结束]

```

```

import pandas as pd
path="F:\大学\作业\程序设计\分块时段 示例.xlsx"
df=pd.read_excel(path)
k=int(input())
task=[[0,0,0,0]for i in range(len(df))]
for i in range(len(df)):
    task[i][0]=df.at[i,"开始"]
    task[i][1]=df.at[i,"结束"]
    task[i][2]=df.at[i,"持续时间"]
    task[i][3]=i+1
schedule=irrigate(task,k)[1]
temp=irrigate(task,k)[0]
for i in range(len(schedule)):
    x=list(set(schedule[i]))
    schedule[i]=x
start=[0 for i in range(len(task))]

```

```

print(schedule)
process=[]
for i in schedule:
    if len(i)!=0:
        for j in i:
            task[j-1][2]-=1
        for j in range(len(task)):
            if task[j][2]==0:
                start[task[j][3]-1]=1
        process.append(start)
print(process[len(process)-1])

```

### (三)、小组分工

成员	分工
李睿熙	课题甄选+算法设计+代码编写+PPT 制作+论文撰写+展示与答辩
郑宗翰	课题甄选+数据收集与整理+可视化流程图制作+PPT 制作+展示与答辩
刘众之	课题甄选+数据收集+算法设计+PPT 制作+论文撰写+展示与答辩
王楷震	课题甄选+数据收集+文献查找+PPT 制作+论文撰写+展示与答辩
易恩畅	课题甄选+文献综述+可视化构建+PPT 制作+论文撰写+展示与答辩