**无人机配送路径规划问题**

*姓名：吴凯璇 学号：2023202210061*

1. **问题重述**

根据要求，本报告通过合理设置参数值，将原问题具体调整如下：

无人机可以快速解决最后10公里的配送，本作业要求设计一个算法，实现如下图所示区域的无人机配送的路径规划。在此区域中，共有3个配送中心，任意一个配送中心有用户所需要的商品，其数量无限，同时任一配送中心的无人机数量无限。该区域同时有11个卸货点（无人机只需要将货物放到相应的卸货点即可），假设每个卸货点会随机生成订单，一个订单只有一个商品，但这些订单有优先级别，分为三个优先级别（用户下订单时，会选择优先级别，优先级别高的付费高）：

* 一般：3小时内配送到即可；
* 较紧急：1.5小时内配送到；
* 紧急：0.5小时内配送到。

我们将时间离散化，也就是每隔30分钟，所有的卸货点会生成订单（0-3个订单），同时每隔30分钟，系统要做成决策，包括：

1. 哪些配送中心出动多少无人机完成哪些订单；

2. 每个无人机的路径规划，即先完成那个订单，再完成哪个订单，...，最后返回原来的配送中心；

注意：系统做决策时，可以不对当前的某些订单进行配送，因为当前某些订单可能紧急程度不高，可以累积后和后面的订单一起配送。

目标：一段时间内（如一天），所有无人机的总配送路径最短

约束条件：满足订单的优先级别要求

假设条件：

1. 无人机一次最多只能携带3个物品；

2. 无人机一次飞行最远路程为20公里（无人机送完货后需要返回配送点）；

3. 无人机的速度为60公里/小时；

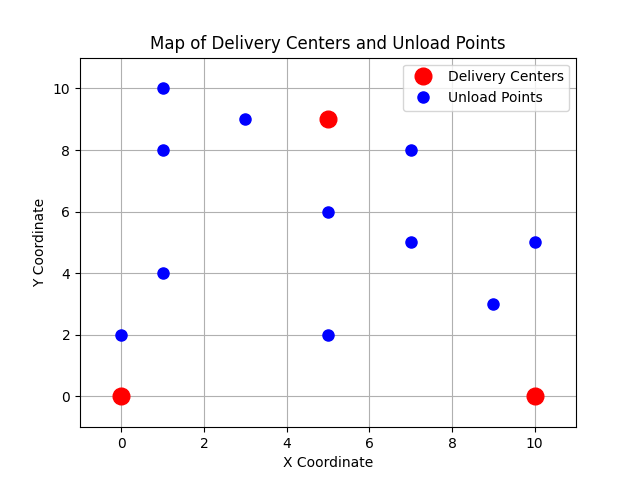
4. 配送中心的无人机数量无限；

5. 任意一个配送中心都能满足用户的订货需求；

1. **问题定义**

**2.1 配送中心和卸货点**

配送区域包含若干配送中心和卸货点。具体地，本系统中设定了三个配送中心，分别位于坐标 (0,0)、(5,9) 和 (10,0)。卸货点共有十一个，分布在不同的地理位置，如 (1,4)、(7,5) 等。如图1所示。



图表 1

此外，配送中心具备无限的货物供应能力和无人机配送能力，无需担忧货源或无人机数量不足的问题。

**2.2 订单的生成和优先级**

在此系统中，订单的生成是由每个卸货点随机产生的。具体来说，每个时间点，每个卸货点都有可能生成0至3个订单，订单生成的具体数量服从均匀分布。每个订单都有一个明确的优先级，分为“一般”、“较紧急”和“紧急”三类。

订单的优先级决定了配送的紧迫性，影响无人机的调度优先顺序。但是，在调度的角度看来，可以把某个订单的属性简化为 这样的一个元组，在某个决策时刻，此时参与决策的订单由两部分组成:

**2.3 假设**

* 在决策时刻，系统做完决策后，无人机马上出发派送所有被分配的订单。
* 派给无人机的订单，必是无人机能在航程内完成的订单。
* 在一个决策周期内，无人机能将完成派给它的所有任务。无人机的最远路程为20公里，速度为60公里/小时可知，完成最远航程所需时间0.33h，小于一个决策周期0.5h。
* 决策时刻的紧急订单定义为在，之后的一个决策周期内到达的订单。

**2.4 非紧急订单延迟处理策略**

**策略：**系统通过推迟当前的没搭载紧急订单的无人机的订单并与后续到来的订单一起处理可以形成更优的路径。

下面对该策略进行证明：

设在某个决策时刻有订单集合，记为搭载紧急订单的无人机的所有订单集合，为未搭载紧急订单的无人机的所有订单集合，为决策时刻的分配无人机订单最优策略，将划分为这两个集合。有：

为一映射，将订单集合和分配策略映射到最短路径距离空间，为订单集合在分配策略下能达到的最短距离。

由的性质，显然有，当且仅当为最优策略时取等号。

因此，对于给定的分配策略, 在决策时刻,有：

因此，系统通过推迟当前的没搭载紧急订单的无人机的订单并与后续到来的订单一起处理可以形成更优的路径。

**2.4 算法流程总结**

在决策时刻，

1. 将新生成的订单添加至上个决策时刻剩余的订单，形成总订单。
2. 对总订单使用遗传算法搜索订单分配策略
3. 发射搭载紧急订单的无人机并推迟当前的没搭载紧急订单的无人机的订单
4. 将剩余的订单传至下一决策时刻
5. **算法设计**

**3.1 遗传算法的框架**

本报告采用遗传算法 (Genetic Algorithm, GA) 来解决无人机配送路径优化问题。遗传算法是一种模拟自然界生物进化过程的搜索算法，通过种群的选择、交叉和变异操作来优化解。在本问题中，遗传算法用于寻找给定总订单的情况下最优的无人机配送路径策略，以最小化总飞行距离并满足订单的优先级约束。

算法流程大致如下：

1. **初始化**：生成初始种群，每个个体代表一种可能的配送路径策略。
2. **评估**：计算种群中每个个体的适应度。
3. **选择**：根据适应度选择优质个体进行繁殖。
4. **交叉和变异**：通过遗传操作产生新个体，增加种群多样性。
5. **替换**：用新生成的个体替换旧个体，形成新一代种群。
6. **终止条件**：重复执行上述步骤直到满足终止条件（如达到最大代数或适应度足够高）。

**3.2组成部分**

### ****a. 适应度函数 (****evaluate****)****

评估每个个体的质量。适应度函数计算一个解决方案的总飞行距离，这是决定无人机配送效率的关键指标。计算从配送中心出发，访问指定订单位置，最后返回配送中心的总距离。如果解决方案违反了载重或距离限制，则适应度函数会返回一个非常高的数值（惩罚），从而在选择过程中自然淘汰这些不可行的解决方案。

### ****b.个体创建函数 (****create\_individual****)****

初始化种群。每个个体代表一个潜在的解决方案，即一种特定的订单到无人机的分配方式。随机选择配送中心和一组订单来形成一个个体。它为算法提供了多样的起始点，有助于探索解空间。

### ****c. 遗传操作****

* **交叉 (crossover)**: 交叉操作用于产生新的个体，通过结合两个父代个体的特征来实现。两个解决方案交换部分无人机订单，以探索新的可能性和组合。同时调整每个无人机的订单列表，确保所有订单都恰好被处理一次。
* **变异 (mutate)**: 变异操作通过随机改变个体的某些部分（如改变配送中心或订单分配）来引入额外的多样性，防止算法过早收敛于局部最优解。

3**.3 伪代码展示**

Step1: 设置开始时间为当前时间，结束时间为开始时间加24小时,种群大小为50,遗传算法代数为100,交叉概率为0.4,变异概率为0.3, 订单列表为空.

Step2: 循环直到当前时间小于结束时间:

1.向订单列表添加新生成的订单

2.初始化遗传算法初始种群

3.为所有个体强制重置适应度

4.运行遗传算法，获取最优个体代表的订单分配策略和无人机路径

5.如果最优个体适应度为10000：

6.增加种群大小和代数，增加交叉和变异概率

7.转到2继续运行

8.初始化航班列表

9.如果无人机携带紧急订单（定义见第2章），则出发该无人机

10.从订单列表中删除这些订单

11.更新当前时间，重置遗传算法的参数

结束循环

1. **算法实现和结果分析**

**4.1使用的技术和工具**

本报告使用Python作为编程语言，使用DEAP (Distributed Evolutionary Algorithms in Python) 作为解决复杂优化问题的遗传算法库。

**4.2 参数设置**

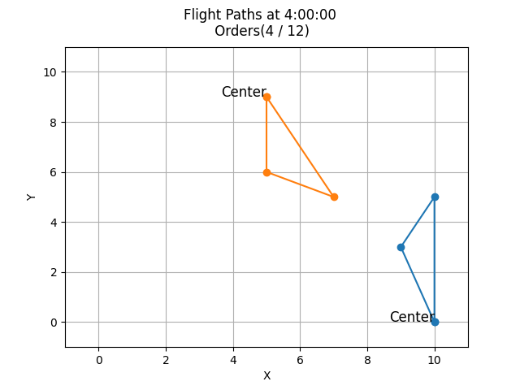
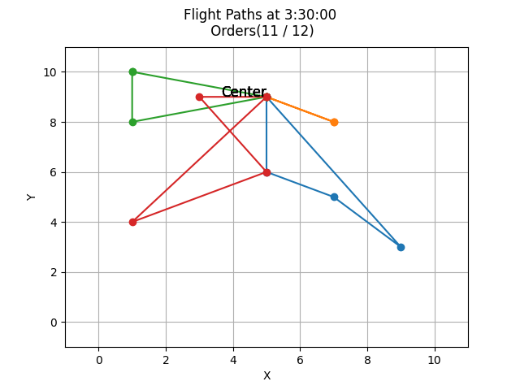
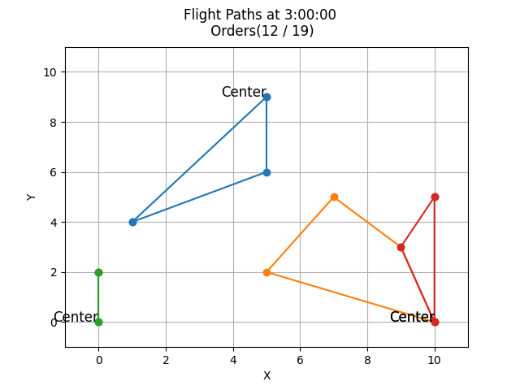
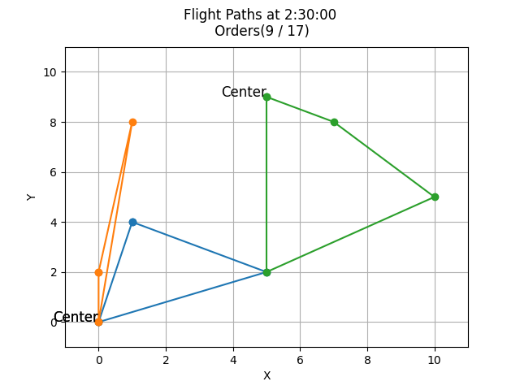
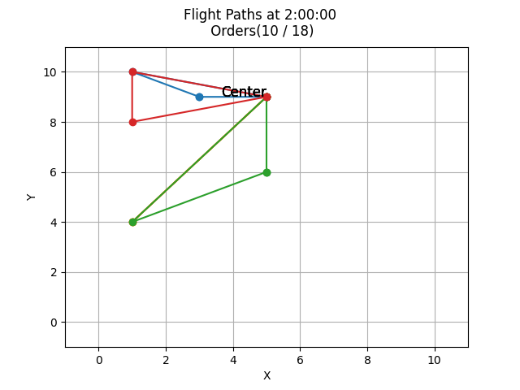
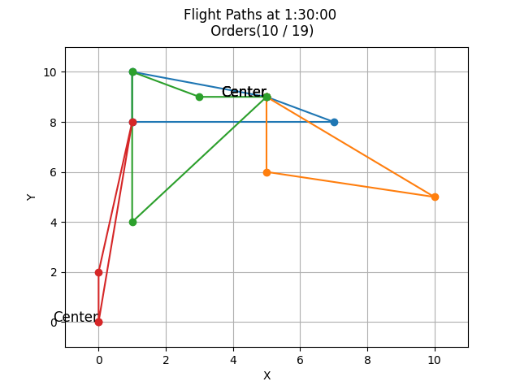
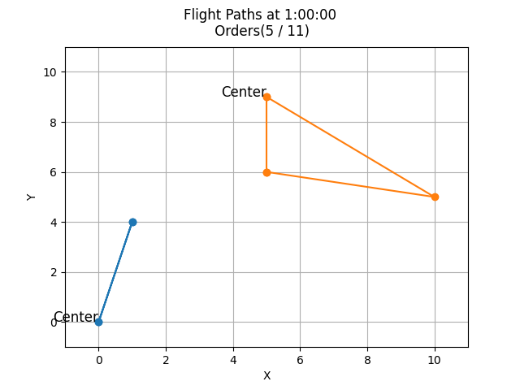
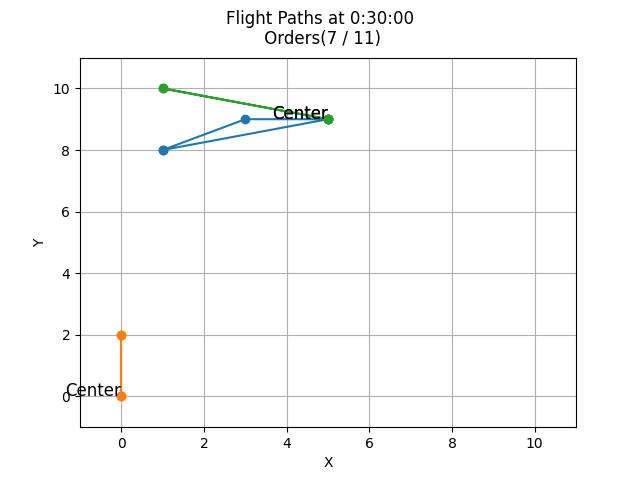
为确保遗传算法能有效运行并找到优化的解决方案，需要选择合适的算法参数。以下是本项目中使用的主要参数设置：

* **种群大小**：设置为50，这意味着每一代算法会评估和进化50个不同的配送方案。这个大小能够平衡算法的搜索能力和计算效率。
* **代数**：设定为100代。这允许算法有足够的迭代次数来逐渐改进解，同时避免过度运行导致的计算资源浪费。
* **交叉概率** (cxpb)：设置为0.4，这表示在每代中，40%的个体将通过交叉操作生成。
* **变异概率** (mutpb)：设置为0.3，即每代中有30%的个体会经历变异操作，增加种群的多样性。

这些参数的选择基于实验和前期的小规模测试，以确保算法能在合理的时间内给出解，同时维持较高的解的质量。

**4.3 运行结果**

运行遗传算法后，算法的执行结果可以通过绘制无人机的飞行路径和记录每代的最佳适应度值来展示。输出算法找到的最短飞行路径总距离，以及这些路径在地理上的布局。图2为一个4小时的模拟。



图表 2

**4.4表现分析**

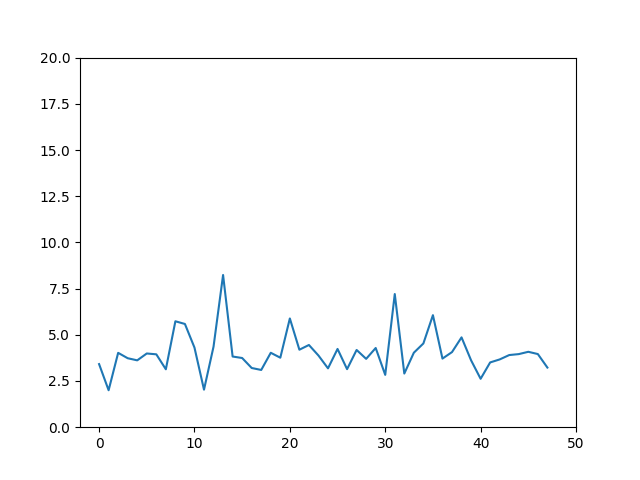
### ****a. 路径分析****

观察无人机的飞行路径图可以直观地了解到无人机是如何从一个配送点飞往另一个配送点的，以及它们是如何避免不必要的长距离飞行。

从遗传算法得到的策略上来看，算法能够有效地将订单分配给数量恰当的无人机，使得在时间和航程地约束下，尽可能使航线更加高效。

**b. 模拟过程中的总航程**

记录并绘制算法每一决策时刻的订单平均最短飞行距离。每一决策时刻的订单平均最短飞行距离在一定的范围内波动，说明算法的收敛效果和稳定性优秀。



图表 3 平均最短飞行距离

通过这些测试和分析，可以更好地理解算法的性能和可能的改进空间，从而对算法进行微调，以适应更广泛或更具挑战性的实际应用场景。

1. **讨论**

#### 5.1 算法性能的优势与局限性

**a. 优势:**

* **适应性强**: 遗传算法通过自然选择机制自我优化，适用于多变的配送环境和复杂的约束条件。
* **全局搜索能力**: 相较于传统优化算法，遗传算法不容易陷入局部最优解，更有可能找到全局最优或接近全局最优的解。
* **可扩展性**: 随着订单数量和卸货点的增加，算法能够通过调整种群大小和代数来适应更大规模的问题。

**b. 局限性:**

* **计算资源需求**: 遗传算法通常需要较多的计算资源，特别是在种群大和代数多的情况下。
* **收敛速度**: 在某些情况下，遗传算法可能需要较长时间才能收敛到最优解，特别是在解空间极大或解的质量差异较小的问题中。

#### 5.2 结论

本研究报告通过遗传算法解决了无人机配送路径优化问题，展示了算法在处理复杂系统约束和多变需求中的有效性。虽然存在一些局限性，但通过持续的研究和技术改进，遗传算法有望在未来的智能物流领域中发挥更大的作用。