1 当前代码处理逻辑详解

您的求解器采用了经典的**两层嵌套循环**结构：外层是Dinkelbach算法，负责处理分数目标；内层是列生成算法，负责求解在给定参数下的线性化问题。

1.1 外循环：Dinkelbach参数化算法 (`dinkelbach_solver.py`)

Dinkelbach算法的核心思想是通过引入一个参数 λ（代表我们对最终“日均飞行时间”的估计值），将非线性的分数目标函数转化为一系列线性的、减法形式的子问题进行迭代求解 [?, ?]。您的`dinkelbach_solver.py`完美地实现了这个过程：

1. **初始化 (Initialization)**：算法首先通过调用一次列生成（此时设置 $\lambda_0 = 0$）来获得一个高质量的初始解。这是一个非常有效的启发式策略，因为它首先专注于最大化飞行时间并最小化惩罚，为算法提供了一个良好的起点。

2. **计算初始 λ**: 利用初始解，计算出第一个比率 $\lambda_1 = \frac{\text{总飞行时间}}{\text{总值勤日}}$。

3. **迭代求解 (Iteration)**：进入主循环，在每一轮迭代 k 中：

   - **求解参数化问题**: 以当前的 λ_k 为参数，调用**列生成求解器**（‘ColumnGenerationSolver‘）来求解一个线性化的主问题。这个主问题的目标函数已经不再是分数形式，而是‘总收益 - λ $_k$ ∗ ;ı,‘bfflΘ

   - **获取新解**: 列生成求解器返回对于当前 λ_k 的最优解 x^k。

4. **收敛性检验 (Convergence Check)**：算法的核心判断依据是参数化目标函数 $F(\lambda_k) = (\sum C_p x_p^k) - \lambda_k (\sum d_p x_p^k)$ 的值。根据Dinkelbach理论，当这个值趋近于0时（即 `abs(F_lambda) < epsilon`），意味着当前的 λ_k 已经非常接近最优比率，算法收敛 [?, ?]。

5. **更新 λ**: 如果未收敛，则利用新解 x^k 计算出更优的下一轮比率 $\lambda_{k+1} = \frac{\sum C_p x_p^k}{\sum d_p x_p^k}$，然后进入下一次迭代。

1.2 内循环：列生成算法 (`attention_guided.py`)

对于外循环中给定的每一个固定的 λ_k 值，您的`ColumnGenerationSolver`都会被调用来求解一个大规模的线性规划问题。由于变量（所有可能的航线组合）数量巨大，您正确地使用了列生成方法 [?]。

1. **主问题与子问题**: 列生成将问题分解为**主问题 (Master Problem)** 和**定价子问题 (Pricing Subproblem)** [?]。

   - **主问题** (`master_problem.py`): 负责从一个已知的、有限的航线组合（列）集合中，选出最优的组合方案。

- **定价子问题** (`attention_guided.py`中的`PricingSubproblem`): 负责在所有可能的、尚未生成的航线组合中，寻找能够改善当前主问题解的新方案（新列）。

2. **迭代过程**:

   - **求解主问题**: `ColumnGenerationSolver`首先调用`MasterProblem`类，求解当前受限主问题（Restricted Master Problem, RMP）的线性松弛版本。
   - **获取对偶价格**: 求解RMP后，最重要的产出是每个航班覆盖约束的**对偶价格（Dual Prices）**，即Gurobi中的`.Pi`属性。这个价格在经济学上可以理解为"多满足一个航班约束所能带来的边际收益"。
   - **求解定价子问题**: 将对偶价格和当前的 λ_k 传递给`PricingSubproblem`。您的子问题求解器（`YourAttentionModel`）利用这些信息，去寻找具有**正检验数（Positive Reduced Cost）**的新航线组合。
   - **添加新列**: 如果找到了检验数为正的新列，则将其加入到主问题的变量集合中，然后重新求解主问题。
   - **收敛**: 重复以上步骤，直到您的Attention模型再也找不到任何检验数为正的新列。此时，内循环（列生成）收敛，表明对于当前的 λ_k，我们已经找到了最优的LP解。

3. **获取整数解**: 在列生成收敛后，您将所有变量类型改为整数（`GRB.BINARY`），并求解最终的混合整数规划（MIP）问题，从而得到一个高质量的整数解。

2 代码实现的数学模型提取

基于对您代码的分析，以下是其实现的数学模型的精确表达。

2.1 整体优化目标 (Original Fractional Problem)

您的代码框架所优化的**原始问题**是一个**纯分数规划**问题，其目标是最大化“每单位值勤日所能产生的净收益”：

$$\text{最大化} \quad Z = \frac{\sum_{p \in P}(w_{score} \cdot C_p - O_p) \cdot x_p - \sum_{f \in F} M_f \cdot s_f}{\sum_{p \in P} d_p \cdot x_p} \tag{1}$$

其中，w_{score} 是您代码中使用的权重，即‘1000‘。这个模型将所有惩罚项都放到了分子中，形成一个衡量“效率”的比率，这使得问题可以使用Dinkelbach算法进行精确求解。

2.2 主问题模型 (Master Problem)

在Dinkelbach算法的第 k 次迭代中，您的`master_problem.py`所求解的**受限主问题 (Restricted Master Problem, RMP)** 可以表述为:

- **集合与索引:**
  - P': 当前已生成的航线组合（列）的集合。
  - F: 所有航班的集合。
- **参数与权重:**
  - $w_{score} = 1000$: 日均飞时的得分权重。
  - C_p: 航线组合 p 的总飞行时间 (`total_flight_time`)。
  - d_p: 航线组合 p 的总飞行值勤日数量 (`total_duty_days`)。
  - O_p: 航线组合 p 自身附带的其他惩罚项之和 (`other_penalties`)。
  - M_f: 未覆盖航班 f 的惩罚成本 (`uncovered_flight_penalty`)。
  - a_{fp}: 二进制参数，如果航线组合 p 覆盖航班 f，则为1，否则为0。
  - λ_k: Dinkelbach算法当前迭代的参数，代表对“加权净收益/执勤日”这一比率的估计值。
- **决策变量:**
  - $x_p \geq 0$: 表示选择航线组合 p 的比例（在最终求解MIP时为二进制）。
  - $s_f \geq 0$: 表示航班 f 未被覆盖的比例（在最终求解MIP时为二进制）。
- **对偶变量:**

– π_f: 与航班 f 的覆盖约束相关联的对偶价格。

RMP模型公式:

$$
\begin{aligned}
\text{最大化} \quad Z_k = & \sum_{p\in P'} (w_{score} \cdot C_p - \lambda_k \cdot d_p - O_p) \cdot x_p \\
& - \sum_{f\in F} M_f \cdot s_f
\end{aligned} \tag{2}
$$

约束于:

$$\sum_{p\in P'} a_{fp} \cdot x_p + s_f = 1, \quad \forall f \in F \quad [\pi_f] \tag{3}$$

$$x_p \geq 0, \quad \forall p \in P' \tag{4}$$

$$s_f \geq 0, \quad \forall f \in F \tag{5}$$

2.3 定价子问题模型 (Pricing Subproblem)

您的attention_guided.py中的PricingSubproblem负责求解定价子问题。其目标是寻找一个当前主问题中不存在的、能最大化改善目标函数的新航线组合 $p^* \in P \setminus P'$。这等价于寻找一个具有**最大正检验数（Reduced Cost）**的列 [?]。

- **检验数 (Reduced Cost) 的计算:** 一个新航线组合 p^* 的检验数 $\bar{c}_{p^*}$ 是其在主问题目标函数中的原始系数，减去它所满足的各个约束的对偶价格之和。根据修正后的主问题模型 (2)，其检验数为：

$$\bar{c}_{p^*} = \underbrace{(w_{score} \cdot C_{p^*} - \lambda_k \cdot d_{p^*} - O_{p^*})}_{\text{新列的原始目标系数}} - \underbrace{\sum_{f\in p^*} \pi_f}_{\text{所覆盖航班的对偶价格之和}}$$

- **子问题模型:** 您的Attention模型实际上是在求解一个启发式的**资源约束最短路径问题 (Resource-Constrained Shortest Path Problem, RCSPP)** [?]。其目标可以表述为：

$$\text{最大化} \quad \bar{c}(p) = (w_{score} \cdot C_p - \lambda_k \cdot d_p - O_p) - \sum_{f\in p} \pi_f$$

约束于: p 是一个满足所有民航法规和合同约束的合法航线组合。

您的Attention模型通过学习历史数据和当前的对偶信息（π_f 和 λ_k），直接生成最有可能满足 $\bar{c}(p) > 0$ 的候选航线组合，从而高效地解决了这个NP-hard的子问题。

参考文献

[1] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, 1967.

[2] R. Baldacci, A. Lim, E. Traversi, and R. Wolfler Calvo. Optimal solution of vehicle routing problems with fractional objective function. *arXiv preprint arXiv:1804.03316*, 2018.

[3] G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*, volume 5. Springer Science & Business Media, 2006.