

## 一种限界优化方法求解航班着陆调度问题

冯小荣<sup>1,2</sup>, 张 帅<sup>1,2</sup>, 丘东林<sup>1</sup>, 王兴隆<sup>1,2</sup>

(1. 中国民航大学空中交通管理学院, 天津 300300; 2. 民航飞联网重点实验室, 天津 300300)

**摘要:** 航空运输需求持续增长与枢纽终端区空域资源紧张的情况日益凸显, 本文提出了一种限界优化的动态规划方法 (Dynamic programming approach to limit optimization, DPALO) 求解终端区航班着陆调度问题 (Arrived landing problem, ALP)。首先建立了时间窗约束的航班着陆调度的离散化数学模型, 推导了固定顺序下求解 ALP 的递推公式, 并结合 ALP 问题特点, 限界优化航班时间窗, 并证明了所提方法不影响模型最优值的求解。其次, 运用精英遗传算法、粒子群算法、线性循环交换和线性循环插空等方法调整航班序列, 以期求得较优解。最后在 OR-Library 数据集进行验证, 实验结果表明, 采用精英遗传算法调整航班着陆序列, DPALO 的计算结果优于已知最优解 (Best known values, BKV)、仿生算法 (Bionic algorithm, BA) 和位移决策算法 (Displacement decision algorithm, DDA), 与细胞自动机优化方法 (Cellular automaton optimization, CAO)、紧致子序列算法 (Compact subsequence algorithm, CSA) 和滚动时域-混合粒子群优化-局部搜索算法 (Rolling horizon framework hybrid particle swarm optimization local search algorithm, RH-HPSO-LS) 的结果相近; DPALO 的时间效率在小样本数据集上时间效率达到毫秒级, 在大样本数据集上相较于 CSA、CAO 和 RH-HPSO-LS 分别提升了 76.88%、89.11% 和 78.28%。

**关键词:** 航班着陆调度; 时间窗约束; 动态规划; 遗传算法; 粒子群算法

**中图分类号:** V355; U8

**文献标志码:** A

**文章编号:** 1005-2615(2024)06-1024-12

## A Limit Optimization Method for Flight Landing Scheduling Problems

FENG Xiaorong<sup>1,2</sup>, ZHANG Shuai<sup>1,2</sup>, QIU Donglin<sup>1</sup>, WANG Xinglong<sup>1,2</sup>

(1. Air Traffic Management College, Civil Aviation University of China, Tianjin 300300, China;

2. Key Laboratory of Internet of Aircraft, Tianjin 300300, China)

**Abstract:** The continuous growth of air transportation demand and the tightness of airspace resources in the terminal area of a hub are becoming more and more prominent. A dynamic programming approach to limit optimization (DPALO) is proposed to solve the arrived landing problem (ALP). First, a discrete mathematical model of flight landing scheduling with time window constraints is established, and a recursive formula for solving ALP with a fixed order is derived. The flight time window is optimized by combining the ALP problem characteristics with the constraints, and it is proved that the proposed method does not affect the solution of the optimal value of the model. Second, elite genetic algorithms, particle swarm algorithms, the linear loop swapping and the linear loop interpolation are applied to adjust the flight sequences and then to finding an optimal solution. Finally, validation is performed on the OR-Library dataset. The experimental results show that using the elite genetic algorithms to adjust the flight landing sequence, DPALO outperforms the best known values (BKV), the bionic algorithm (BA) and the displacement decision algorithm (DDA)

**基金项目:** 国家自然科学基金面上项目 (62173332); 国家自然科学基金重点项目 (U2133207); 中央高校基本科研业务费-自然科学重点项目 (3122023050); 中央高校基本科研业务费专项基金项目 (3122020051)。

**收稿日期:** 2024-06-11; **修订日期:** 2024-09-13

**通信作者:** 王兴隆, 男, 教授, 硕士生导师, E-mail: xinglong1979@163.com。

**引用格式:** 冯小荣, 张帅, 丘东林, 等. 一种限界优化方法求解航班着陆调度问题[J]. 南京航空航天大学学报, 2024, 56(6): 1024-1035. FENG Xiaorong, ZHANG Shuai, QIU Donglin, et al. A limit optimization method for flight landing scheduling problems[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2024, 56(6): 1024-1035.

and obtains similar results to those of the cellular automata optimization approach (CAO), the tight subsequence algorithm (CSA), and the rolling horizon framework of hybrid particle swarm optimization local search algorithm (RH-HPSO-LS). The time efficiency of DPALO achieves milliseconds in time on the small sample dataset, and it is improved by 76.88%, 89.11%, and 78.28% on the large sample dataset in comparison to CSA, CAO, and RH-HPSO-LS, respectively.

**Key words:** flight landing scheduling; time window constraints; dynamic programming; genetic algorithm; particle swarm optimization (PSO)

2023 年,民航全行业共完成运输总周转量 1 188.3 亿吨千米、旅客运输量 6.2 亿人次、货邮运输量 735.4 万吨,同比分别增长 98.3%、146.1% 和 21%,行业运输生产呈现稳健恢复的良好局面<sup>[1]</sup>。2023 年,全国客运航空公司共执行航班 467.17 万班次。其中,正常航班 410.19 万班次,平均航班正常率为 87.80%。较 2022 年下降 7.18%,全国客运航班平均延误时间为 10 min,比 2022 年增加 6 min。据文献[2]统计跑道资源紧张是造成这终端枢纽机场产生延误主要原因。

为了防止航班在进近的过程中受到前序航班的尾流影响,相邻的航班要满足最小尾流距离间隔。国际民用航空组织(International Civil Aviation Organization, ICAO)基于最大起飞质量将航班分类,其中最大起飞质量小于 7 t 为轻型机,7~136 t 为中型机,136 t 以上为重型机。前序航班产生的尾流会影响后续航班的起飞与降落。基于 ICAO Doc 4444 中的有关规定,中国民航局(Civil Aviation Administration of China, CAAC)制定了尾流间隔标准(CCAR-93-R5)<sup>[3]</sup>。进近时航空器的间隔需同时满足尾流间隔和飞机安全间隔;若无尾流间隔要求,则需满足飞机安全间隔即可。2010 年,Balakrishnan 等<sup>[4]</sup>将尾流间隔转换为尾流时间间隔,如表 1 所示。本文所讨论的问题就是在满足尾流时间间隔的同时,找到最合适的机型排序,从而缩短航班起降时间,减少着陆代价。

表 1 尾流时间间隔			
Table 1 Tailrace time interval			
前机	后机		
	重型	中型	轻型
重型	96	157	196
中型	60	69	131
轻型	60	69	82

根据文献[5-6],航班着陆受最早着陆时间与最晚着陆时间的限制,且需满足航班间的最小尾流时间间隔。最早着陆时间是飞机以最大空速飞行,可以最早着陆的时间;而最晚着陆时间是飞机可以保持(盘旋)的最大允许时间。航班在空中飞行,会

有不同的成本产生:例如航班想要提前着陆,会使目标机场变得更加繁忙,增加管制员的压力,导致人力成本增加;若航班延迟着陆,则会增加燃油消耗。因此每架航班着陆时存在一个最优着陆时间。如图 1 所示,航班在最优时刻着陆的代价最小,视为 0。偏离目标着陆时间所产生的代价与偏差量成正比。

图 1 中  $T_i$ 、 $E_i$ 、 $L_i$  分别表示航班  $i$  的最优着陆时间、最早着陆时间和最晚着陆时间,  $g_i$ 、 $h_i$  表示航班相对于最优着陆时间提前或者延误的单位时间代价<sup>[7]</sup>。

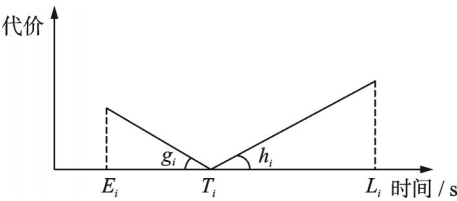


图 1 航班着陆时间与代价关系  
Fig.1 Flight landing time versus cost

为了解决航班着陆调度问题,已有许多学者提出解决该问题的算法。Beasley 等<sup>[8-9]</sup>构建了 OR-Library 数据集,数据集中给定了最早着陆时间,最优着陆时间,最晚着陆时间等概念,确定从利益角度出发,建立整数规划模型优化目标。Pinol 等<sup>[10]</sup>考虑了静态飞机着陆问题中的多跑道情况,采用散点搜索和生物算法,实现了跑道的有效利用,但是该算法只能在小样本的数据中找到最优解,在较大规模数据上效果不理想;Yu 等<sup>[11]</sup>将飞机着陆调度问题表述为一个需要实时解决的约束优化问题,提出了一种元胞自动机优化方法来解决飞机着陆调度问题,对于少于 100 架航班的计算时间极短,但是当超过 150 架航班时,计算时间将会大幅增加;许军海等<sup>[12]</sup>运用遗传算法对机场终端区进行航班排序,虽然结果得到了优化,但是算法效率较低;王满超<sup>[13]</sup>根据飞机着陆调度问题的混合整数规划模型,设计了独特的粒子群算法,充分利用了粒子群算法解决连续问题的优势,相对于其他离散解法降低了问题的复杂度;Girish<sup>[6]</sup>提出了一种滚动视界框架下的混合粒子群优化算法,用于解

决飞机着陆问题;Chen等<sup>[14]</sup>将飞行调度问题表述为一个整数线性规划,然后转化为一个多目标优化问题,提出了一种不确定性感知的飞行调度算法,对给定跑道上航班的到达/离开顺序进行排序;Rodríguez-Díaz等<sup>[15]</sup>提出了以最小化噪声影响和最少总燃料消耗为目标,以尾流间隔和位置移动限制为约束的双目标优化模型,并使用巴拉哈斯机场的数据进行实验验证;张军峰等<sup>[16]</sup>构建PMS(Point fusion system)多目标排序与调度模型,提出基于多目标帝国竞争算法(Imperialist competitive algorithm, ICA),与先到先服务(First come first served, FCFS)相比有较大的提升;Jiang等<sup>[17]</sup>研究了到达和出发航班的联合调度问题,提出了“到达航班均衡延误时间”和“出发航班均衡满意度”评价指标,针对到达和出发的4种联合运营交通场景,建立了相应的到达和出发航班联合调度的双层规划模型,更好地平衡公平性和效率,使得调度结果更加合理;冯小荣等<sup>[7]</sup>建立了时间窗约束的航班着陆优化模型,提出一种基于紧致子序列的算法求解固定顺序下航班着陆调度问题,在小样本量数据上计算精度与速度较优,但其同样存在大规模数据集的计算效率较低的问题;王莉莉等<sup>[18]</sup>基于点融合系统分别建立了可对冲不确定性影响的静态和动态排序模型,引入了额外缓冲区来应对不确定因素的影响;王建忠等<sup>[19]</sup>提出基于多目标时间索引模型的进离场航班排序方法,通过对目标函数设置不同的权重,同时以进离场航班的总体延误时间最小以及管制员工作负荷增加最小为目标,采用遗传算法对模型进行求解;Park等<sup>[20]</sup>设计扩展先到先服务调度程序,基于优先级的计划,优先满足高优先级航班的约束,解决航班调度问题;杜卓铭等<sup>[21]</sup>考虑管制员工作经验,将进场排序问题转化为机器学习领域的分类问题,构建随机森林分类器预测着陆次序;周鼎凯<sup>[22]</sup>运用强化学习算法求解航班排序问题,从离场、进场、进离场3个方面构建终端区航班排序静态模型,在进离场排序模型中引入了“航班满意度”,增强了机场和航空公司使用资源的公平性。

虽然以上方法在求解航班着陆顺序时有不错的效果,但算法复杂度较高,对于大运算量的排序问题,需要花费大量的运算时间,难以快速调整应对实时性的需求。点融合技术解决了航班在飞行过程中到着陆点时间的不确定性,对此问题也有较好的解,但点融合技术受空域结构影响较大,在中国现行的空域规划模式下应用较为困难。因此,如何提高算法效率、减少计算时间是当前需要解决的问题。

针对上述问题,本文提出了一种限界优化的动态规划方法(Dynamic programming approach to limit optimization, DPALO)求解固定顺序航班着陆问题。将航班着陆调度问题离散化,依据动态规划思想建立固定顺序下航班着陆调度问题的递推公式。运用数学归纳的方法,限界优化每一步的求解范围,并证明了优化后的求解范围不影响问题最优值的求解。采用精英遗传算法、粒子群算法(Particle swarm optimization, PSO)对航班顺序进行搜索,求得问题的较优解。所提算法是一种确定性算法,具有较强鲁棒性。主要贡献如下:

(1)将航班着陆调度问题离散化,基于动态规划的思想,建立固定顺序下航班着陆调度问题的求解模型,并推导出求解问题的递推公式;

(2)针对固定顺序求解模型搜索范围大,时间复杂度高,采用数学归纳方法对每步搜索范围进行限界优化,并证明了优化后的搜索范围不影响最优值的求解。

## 1 航班着陆问题数学模型及求解

### 1.1 航班着陆调度的数学模型

本文沿用文献[7]的数学模型,采用离散化,限界优化的方法对其进行优化。

时间窗约束的多航班优化模型的目标是对 $N$ 架航班搜寻一个最优的调度顺序 $\pi^*$ ,并计算 $\pi^*$ 顺序下每个航班的最优着陆时间 $t_{\pi}^i$ ,使得 $N$ 架航班的总代价 $C_{T\pi}^N$ 最小。

$$C_{T\pi}^N = \arg \min_{\pi \in \Pi} c_{T\pi} \quad (1)$$

式中: $\pi$ 指当前航班着陆顺序; $\Pi$ 为 $N$ 架航班着陆顺序排列的集合; $c_{T\pi}$ 指 $\pi$ 顺序下航班最小着陆总代价。

$$c_{T\pi} = \min \sum_{i=1}^N f(t_{\pi}^i) \quad (2)$$

式中 $f(t_{\pi}^i)$ 为 $\pi$ 顺序下第 $i$ 架航班在 $t_{\pi}^i$ 时刻降落时的代价,即

$$f(t_{\pi}^i) = \begin{cases} g_{\pi}^i \times (T_{\pi}^i - t_{\pi}^i) & t_{\pi}^i \leq T_{\pi}^i \\ h_{\pi}^i \times (t_{\pi}^i - T_{\pi}^i) & t_{\pi}^i > T_{\pi}^i \end{cases} \quad (3)$$

式中: $T_{\pi}^i$ 表示 $\pi$ 顺序下 $i$ 航班的最优着陆时间; $g_{\pi}^i$ 、 $h_{\pi}^i$ 分别表示 $\pi$ 顺序下 $i$ 航班相较于其最优着陆时间 $T_{\pi}^i$ 提前或延误的单位代价系数,模型同时满足尾流时间间隔约束、最早着陆时间约束和最晚着陆时间约束。

$$t_{\pi}^j - t_{\pi}^i \geq S_{\pi(i,j)} \quad \forall j > i; i \leq N; j \leq N \quad (4)$$

$$t_{\pi}^i \geq E_{\pi}^i \quad \forall i \leq N \quad (5)$$

$$t_{\pi}^i \leq L_{\pi}^i \quad \forall i \leq N \quad (6)$$

式中: $t_{\pi}^i$ 、 $t_{\pi}^j$ 、 $S_{\pi(i,j)}$ 分别为 $\pi$ 顺序下航班 $i$ 、航班 $j$ 可以



降落时间及二者之间的最小尾流时间间隔; $E_{\pi}^i, L_{\pi}^i$ 分别为 $\pi$ 顺序下航班 $i$ 的最早着陆时间和最晚着陆时间。

## 1.2 离散化

一个固定航班顺序的着陆调度问题可以转化为每架航班着陆时间的整数规划问题,用动态规划算法(Dynamic programming algorithm, DPA)的方法求解。

设 $F(t_{\pi}^i)$ 表示 $\pi$ 顺序下第 $i$ 个航班在 $t_{\pi}^i$ 时刻着陆时前 $i$ 架航班累计产生的最小总代价。考虑航班序列中第1架航班有 $F(t_{\pi}^1)=f(t_{\pi}^1)$ ,则第 $i$ 架航班有 $F(t_{\pi}^i)=f(t_{\pi}^i)+\min(F(t_{\pi}^{i-1}))$ 。建立 $F(t_{\pi}^i)$ 用时间 $t_{\pi}^i$ 表示的递推函数为

$$F(t_{\pi}^i)=\begin{cases} f(t_{\pi}^i) & i=1 \\ f(t_{\pi}^i)+\min_{E_{\pi}^{i-1}\leq t_{\pi}^{i-1}\leq \min(t_{\pi}^i-S_{\pi(i-1,i)}, L_{\pi}^{i-1})} (F(t_{\pi}^{i-1})) & 1<i\leq n \end{cases} \quad (7)$$

式(2)可以进一步简化为

$$c_{1\pi}=\min(F(t_{\pi}^i)) \quad \pi\in\Pi \quad (8)$$

根据式(5,6),可以得到函数 $f(t_{\pi}^i)$ 的因变量 $t_{\pi}^i$ 的取值范围为 $[E_{\pi}^i, L_{\pi}^i]$ ,考虑式(4)和目标函数 $c_{1\pi}$ 的最佳值,因变量 $t_{\pi}^i$ 的取值范围可以进一步缩小。1.3节将进一步分析因变量 $t_{\pi}^i$ 的取值范围。图2为离散化后 $\pi$ 序列下航班着陆代价的计算步骤。其

中每个圆形的上半部分表示航班在该时刻着陆所花费的代价,如: $f(E_{\pi}^i)$ 为 $\pi$ 序列下第 $i$ 架航班在最早着陆时间着陆所花费的代价,圆形的下半部分表示航班在该时刻着陆,与所有前序航班累积产生的最小总代价;如: $F(E_{\pi}^i)$ 为第 $i$ 架航班在最早着陆时间着陆时与前序 $i-1$ 架航班累积产生最小总代价。航班降落时间需要同时满足约束(4,5,6),当 $E_{\pi}^{i+1}<E_{\pi}^i+S_{\pi(i,i+1)}$ 时,第 $i+1$ 架航班不能在 $[E_{\pi}^{i+1}, E_{\pi}^i+S_{\pi(i,i+1)})$ 时间降落,如图2左下部分虚框所示;当 $L_{\pi}^i>L_{\pi}^{i+1}-S_{\pi(i,i+1)}$ 时,在计算前 $i+1$ 架航班着陆的总代价时,第 $i$ 架航班在时间窗内的 $(L_{\pi}^{i+1}-S_{\pi(i,i+1)}, L_{\pi}^i]$ 不予计算,如图2右上部分虚框所示。第 $i+1$ 架航班在 $E_{\pi}^i+S_{\pi(i,i+1)}$ 时刻着陆时,第 $i$ 架航班只能在 $E_{\pi}^i$ 时刻着陆,此时前 $i+1$ 架航班的着陆代价为 $F(E_{\pi}^i+S_{\pi(i,i+1)})=f(E_{\pi}^i+S_{\pi(i,i+1)})+F(E_{\pi}^i)$ 。若第 $i+1$ 架航班在 $L_{\pi}^{i+1}$ 时刻着陆,第 $i$ 架航班则可以在 $[E_{\pi}^i, L_{\pi}^{i+1}-S_{\pi(i,i+1)}]$ 其中任意时刻着陆而不影响第 $i+1$ 架航班的着陆,因此计算 $F(L_{\pi}^{i+1})$ 时需要找出前序航班累积产生的最小总代价,即 $F(L_{\pi}^{i+1})=f(L_{\pi}^{i+1})+\min(F(E_{\pi}^i), \dots, F(L_{\pi}^{i+1}-S_{\pi(i,i+1)}))$ 。根据式(8),那么 $\pi$ 顺序下前 $i+1$ 架航班的最小着陆总代价为 $c_{1\pi}^{i+1}=\min(F(E_{\pi}^i+S_{\pi(i,i+1)}), \dots, F(L_{\pi}^{i+1}))$ 。

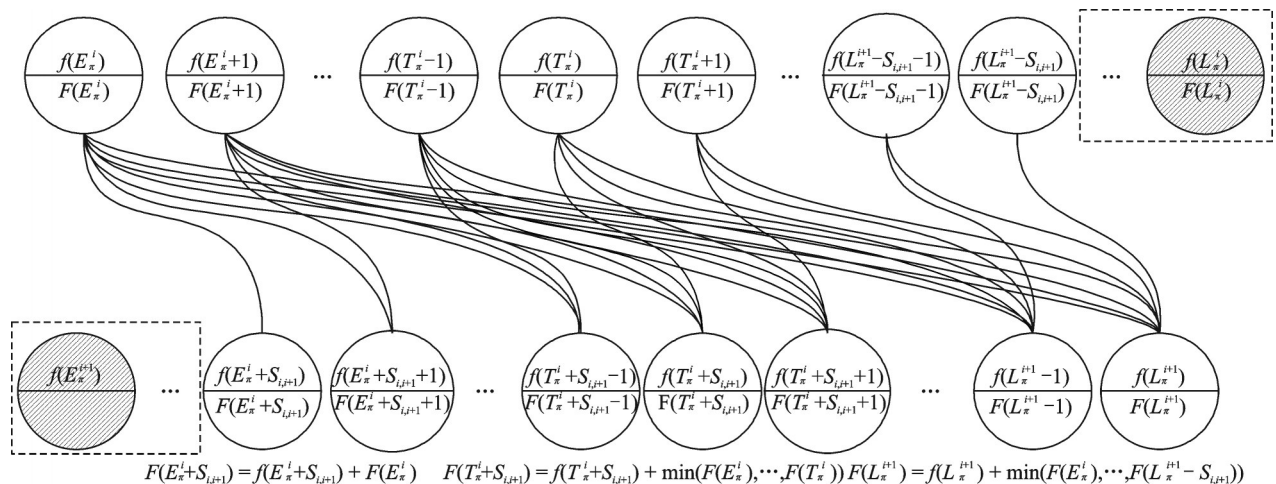


图2 离散化后的搜索范围

Fig.2 Search range after discretization

## 1.3 限界优化调度时间窗

**命题1** 设 $\pi$ 是一个固定的航班降落顺序,在满足约束(3,4,5)的情况下,序列 $t=\{t_{\pi}^i|i=1,2,\dots,n\}$ 是该顺序下使得 $c_{1\pi}$ 取得最小值的各航班着陆时间,即 $c_{1\pi}=\sum_{i=1}^n f(t_{\pi}^i)=\min(F(t_{\pi}^n))=$

$F(t_{\pi}^{n*})$ ,则有: $t_{\pi}^{1*}\leq T_{\pi}^1$ 。

采用反证法证明,具体过程如下:首先假设 $\pi$ 顺序下,存在序列 $t'=\{t_{\pi}^{i'}|i=1,2,\dots,n\}$ 使得 $c_{1\pi}$ 获得最小值, $c_{1\pi}=\sum_{i=1}^n f(t_{\pi}^{i'})$ ,且有 $t_{\pi}^{1'}>T_{\pi}^1$ 。然后,将第1架航班的着陆时间由 $t_{\pi}^{1'}$ 提前到 $T_{\pi}^1$ 。由于第

1架航班前移,并不影响后续航班的安全间隔和着陆时间,令后续航班的降落时间保持不变,即 $\pi$ 序列按照 $t''=\{T_{\pi}^1, t_{\pi}^{1'}, t_{\pi}^{2'}, \dots, t_{\pi}^{n'}\}$ 确定的时间顺序着陆,此时,  $c_{1\pi}=f(T_{\pi}^1)+\sum_{i=2}^n f(t_{\pi}^{i'})=0+\sum_{i=2}^n f(t_{\pi}^{i'})<h_{\pi}^1\times(t_{\pi}^{1'}-T_{\pi}^1)+\sum_{i=2}^n f(t_{\pi}^{i'})$ ,及序列 $t''$ 可以获得更小的代价。因此,假设与命题1产生矛盾, $t_{\pi}^{1'}\leq T_{\pi}^1$ 得证。第1架航班的时间窗可以限界优化为 $[E_{\pi}^1, T_{\pi}^1]$ 。

**推论1** 同理采用反证法可得,最后1架航班 $n$ 的最优着陆时间不会早于其最早着陆时间,因此,第 $n$ 架航班的时间窗也可以限界优化为 $[T_{\pi}^n, L_{\pi}^n]$ 。

**推论2** 根据命题1、推论1和约束(3,4),可以

推导出式(8,9)。

$$e_{\pi}^i = \begin{cases} T_{\pi}^i & i = n \\ \max(E_{\pi}^i, \min(T_{\pi}^i, e_{\pi}^{i+1} - S_{\pi(i+1,i)})) & i < n \end{cases} \quad (8)$$

$$l_{\pi}^i = \begin{cases} T_{\pi}^i & i = 1 \\ \min(L_{\pi}^i, \max(T_{\pi}^i, l_{\pi}^{i-1} + S_{\pi(i-1,i)})) & i > 1 \end{cases} \quad (9)$$

推论2给出了限界优化后航班着陆的时间窗 $[e_{\pi}^i, l_{\pi}^i]$ 。图3描述了限界优化后的搜索范围,图中虚框为相较于DPA优化省去的搜索范围,虚线为省略的搜索步骤,剩余部分的搜索策略不变。前 $i+1$ 架航班的最小着陆代价所需要搜索的范围也对应减小, $c_{1\pi}^{i+1}=\min(F(e_{\pi}^{i+1}), F(e_{\pi}^{i+1}+1), \dots, F(l_{\pi}^{i+1}))$ 。相比DPA大幅减少了计算步骤,简化了算法复杂度。

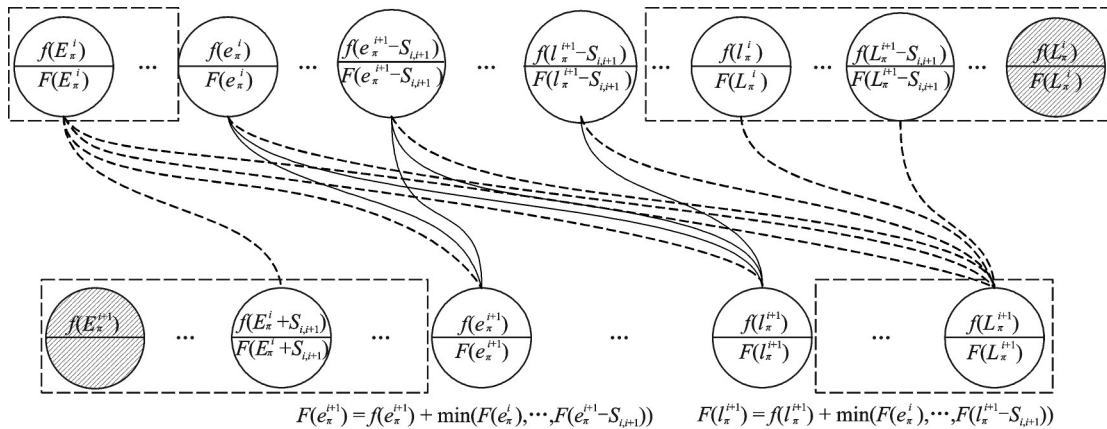


图3 限界优化后的搜索范围

Fig.3 Search range after limit optimization

图4为OR-Library数据集Airland1中10架航班的初始时间窗 $[E_{\pi}^i, L_{\pi}^i]$ 与限界优化后的时间窗 $[e_{\pi}^i, l_{\pi}^i]$ ,其中虚线为DPA的搜索范围,实线为DPA LO的搜索范围,可以看出限界优化可以大幅缩小算法的计算范围,降低算法复杂度。

综上,对搜索范围限界后对此序列进行动态规划求解,具体算法如下。

函数:计算着陆成本

输入:planes\_ticks(飞机的着陆时间信息数组)  
planes\_costs(飞机的成本信息数组)  
sep\_data(飞机之间的间隔时间数组)

输出:最终成本,每架飞机的最终着陆时间

**步骤1** 根据式(9,10)计算限界优化后每架飞机着陆时间的上下界, $e_{\pi}^i, l_{\pi}^i$ 。

**步骤2** 初始化pre\_cost,每架飞机的着陆成本;初始化time\_list,存储每架飞机的着陆时间;初始化record\_list,记录每架飞机的成本选择。

**步骤3** 正序遍历每架飞机,对第 $i$ 架航班

(1)对于每架飞机,初始化cur\_cost数组,计算其可能的着陆时间的成本;

(2)对于在 $[e_{\pi}^i, l_{\pi}^i]$ 内的每个时刻 $t$ ,计算该时刻 $t$ ,第 $i$ 架航班的代价 $f(t_{\pi}^i)$ ;

(3)计算该时刻,前 $i$ 架航班的最小总代价 solution\_a: $F(t_{\pi}^i)$ ;

(4)如果是第1个时间点,或者 solution\_a 小于之前记录的成本 solution\_b,则更新 cur\_cost 和 record\_list;

(5)更新 pre\_cost 和记录着陆时间和 time\_list,记录当前飞机的着陆时间。

**步骤4** 当 $i < N$ , GO to 步骤3,否则下一步。

**步骤5** 逆序遍历 record\_list,根据记录更新每架飞机的最终成本和着陆时间调整。

**步骤6** 输出总成本和每架飞机的最终着陆时间。

#### 1.4 优化策略

为了搜索最优的航班着陆顺序 $\pi^*$ ,本文基于动态规划理论的思想,考虑最大位置偏移(Maxi-

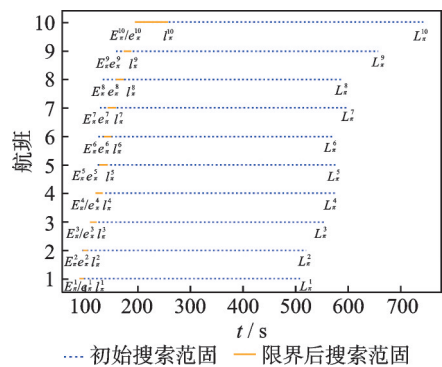


图4 算法搜索范围

Fig.4 Algorithm search range

mize position shifting, MPS),采用精英遗传算法(Genetic algorithm,GA)与PSO,对1.1所建的问题模型进行求解,并将两种算法计算结果与以往的算法进行比较。

(1)适应度函数设计

适应度函数可以评价个体优劣,其选取直接影响到遗传算法的收敛速度以及能否找到最优解。适应度函数为

$$I(x)=\min(F(t_{\pi}^i)) \quad \pi \in \Pi$$

(2)GA算子设计

①编码:使用整数编码方法来编码染色体。染色体的长度(即每条染色体中的基因数)等于等待排序的航班数,每条染色体对应于航班着陆的最优调度问题的解,染色体中每个基因代表一个航班索引。

②选择:根据计算结果,然后保留成本较低的数个精英个体作为下一代的父代。

③交叉:随机选择2个父代个体进行交叉。首先随机选择1个区域进行交换,然后处理交换区域内的冲突,确保每个元素在解决方案中只出现1次。

④变异:随机选择2个位置并交换这2个位置上的元素,模拟基因突变。

(3)PSO算子设计

在PSO中,一个粒子代表着一种航班排序方式。粒子通过在搜索空间中移动来寻找函数的最小值。在每次迭代中,粒子会根据个体经验,自己找到的最好位置来调整其速度和位置。粒子还会考虑群体经验,让整个群体找到的最好位置,以帮助引导搜索方向。以期找到使目标函数值最小的参数组合。

2 实验及结果分析

2.1 实验数据

实验数据选取OR-Library中的航空器着陆问

题的数据集。数据集中包含13组航班序列,每组航班序列中包含不同数量的航班,第1组到第8组,为小样本量数据集,分别包含10、15、20、20、20、30、44、50架航班。后5组为大样本量数据集。分别包含100、150、200、250、500架航班。数据集中每架航班具有7个属性,分别表示航班在管制雷达出现时间、可以降落的最早时间、最优降落时间、可以降落的最晚时间、相对于最优时间提前降落单位代价、相对于最优时间延后降落单位代价,最后一个数据项表示与其他航班的尾流时间间隔,是一个 $n \times n$ 的矩阵, $n$ 表示航班的数量, $n_{ij}$ 表示第 $j$ 架航班尾流第 $i$ 架航班所需要的尾流间隔,其数值取决于两架航班的机型。此数据集是由Beasley等<sup>[9]</sup>最早提出,此前许多学者的研究基于此数据集展开的<sup>[6-11,13,23]</sup>。Girish<sup>[6]</sup>、Beasley等<sup>[8]</sup>使用此数据集研究多跑道的航班着陆排序;冯小荣等<sup>[7]</sup>、Pinol等<sup>[10]</sup>、王满超<sup>[13]</sup>使用此数据集研究了单跑道着陆排序。本文适用运行场景为双跑道机场,跑道运行模式为隔离平行运行模式,即一条跑道用于离场,一条跑道用于进近,因此进离场航班互不影响。

2.2 实验结果及分析

实验环境为AMD 7950X处理器、128 GB内存,编程软件为Python语言。

对于固定顺序下航班着陆代价的求解,如表2所示,DPA的运算效率要优于CSA,DPA更适合解决此问题。在对其进行限界优化后,运算效率得到了进一步的提升。对于样本量小于50架航班(前7组数据集),DPALO的运算速度为0.012 ms,算法效率较高。并且在大样本量计算时的运算速度同样达到毫秒级。500架航班(第13组数据集)

表2 算法运算时间对比

Table 2 Comparison of algorithm operation time ms

组别	DPA	CSA	DPALO
1	0.02	1	0.01
2	0.04	2	0.01
3	0.06	5	0.01
4	0.05	6	0.01
5	0.06	6	0.01
6	0.89	6	0.01
7	0.11	15	0.01
8	0.38	21	0.04
9	4.74	65	0.06
10	5.58	11	0.13
11	7.57	27	0.12
12	9.87	29	0.16
13	18.85	64	0.35



固定顺序下着陆代价的计算仅用时 0.35 ms,远快于 CSA 的 64 ms,可见限界优化对算法复杂度与运算效率有大幅提升。

在航班着陆顺序优化前,首先确定初始序列,对基于 FCFS 的航班排序与基于目标时间顺序(Target time sorting, TTS)的航班排序,同时采用 DPALO 对航班着陆总代价进行求解。FCFS 是以航班在雷达上出现的先后顺序进行排序,TTS 是以航班最佳降落时间的先后顺序进行排序。不同的初始序列对其进行优化所需迭代的次数也有所不同,若初始序列本身就有较好的结果,那么需要优化迭代的次数就会减少。选择效果更好的初始序列能够减少大量的计算过程,提高算法效率,更容易得到最优解,结果见表 3。其中 TTS 的运算结果更接近已知最优解(Best known values, BKV)<sup>[24]</sup>,对其进行优化可以更快速的收敛,得到较优解。

表 3 初始序列选取对比

Table 3 Comparison of initial sequence selection

组别	BKV	DPALO	
		FCFS	TTS
1	700	1 280	700
2	1 480	1 790	1 500
3	820	1 790	1 730
4	2 520	4 890	2 520
5	3 100	6 470	5 420
6	24 442	24 442	24 442
7	1 550	1 550	1 550
8	1 950	18 870	2 510
9	5 611	18 937.16	8 019.42
10	12 329	27 660	20 806.94
11	12 418	35 532.91	18 513.27
12	16 209	46 471.72	22 937.88
13	44 832	99 776.58	49 207.07

为了更直观看到 FCFS 与 TTS 航班顺序的差异,在数据集 Airland1 中,以 FCFS 排序为航班编号,得到航班 1~10。如图 5 和 6 所示,在 FCFS 排序下航班 8 的实际降落时间离最佳降落时间有较远的距离,会产生较大的代价,其总代价为 1 280。而在 TTS 排序下,航班 8 的实际降落时间与最佳降落时间更接近,且航班 7 的实际降落时间也在最佳降落时间附近,其总代价降至 700。因此,选取 TTS 作为后续优化的初始序列。

以 TTS 作为初始序列,基于最大位置偏移的 GA 进行优化排序,变异率 60%,群体总数 100,遗传保留 30,遗传 500 次。实验结果见表 4,其中当 MPS=1 时,那么序列中的航班至多只能与其相邻

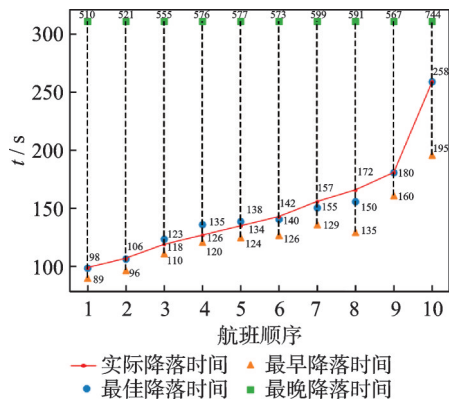


图 5 基于 FCFS 的航班排序

Fig.5 Sorting of flights based on FCFS

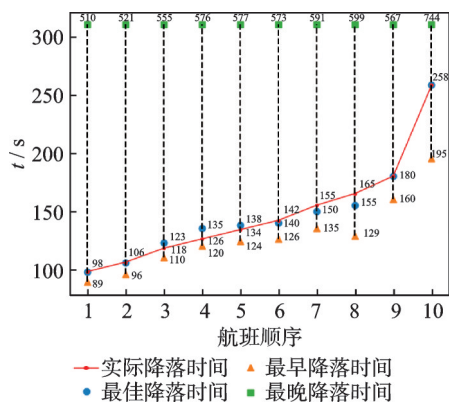


图 6 基于 TTS 的航班排序

Fig.6 Sorting of flights based on TTS

一位的航班进行交换遗传的操作。当 MPS=2 时,航班至多可以与相邻两位的航班进行交换遗传,以此类推。由表 4 可知,4 组算法平均结果均优于 BKV,平均运算时间均能达到 3.2 s 左右,运算效率有较大的提高。且 MPS=3 时的效果最好,平均差值达到 -255.36。前 7 组的运算结果均达到 BKV,运算时间达到毫秒级别,第 8 组与第 13 组的运算结果优于 BKV。4 组数据运算均采用 DPALO+GA+MPS 进行求解,其平均运算时间相差较小,认为遗传算法更适用于此问题的求解。因此选取 MPS=3 的算法结果,进行后续比较研究,结果如表 4 所示。

分析基于 MPS-GA 遗传算法的执行效率,以 Airland8(50 架航班)与 Airland13(500 架航班)为例,观察算法迭代次数与总代价下降的关系。图 7 为 50 架航班分别设 MPS=1、2、3 和 4 的迭代优化过程,迭代次数为 30 次。可以看出,在小样本量的条件下总代价随迭代次数下降非常迅速,迭代不超过 20 次就能给出最优解,并且耗时 0.06 s。表明此算法在中小样本量条件下,计算精度高,运算时间达到毫秒级别。

图 8 为 500 架航班分别设 MPS=1、2、3 和 4 的

表 4 DPALO+GA 的运算结果

Table 4 Results of DPALO+GA operations

组别	BKV	GA							
		MPS=1		MPS=2		MPS=3		MPS=4	
		$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$
1	700	700	0.004	700	0.004	700	0.004	700	0.004
2	1 480	1 480	0.03	1 480	0.037	1 480	0.028	1 500	0.031
3	820	820	0.028	820	0.027	820	0.025	820	0.027
4	2 520	2 520	0.004	2 520	0.004	2 520	0.004	2 520	0.004
5	3 100	3 100	0.043	3 100	0.043	3 100	0.043	3 100	0.048
6	24 442	24 442	0.005	24 442	0.004	24 442	0.004	24 442	0.004
7	1 550	1 550	0.005	1 550	0.005	1 550	0.005	1 550	0.005
8	1 950	1 864.5	0.061	1 862.25	0.052	1 863.75	0.056	1 865.5	0.069
9	5 611	5 735.06	1.874	5 761	2.244	5 694.52	2.541	5 731.42	2.391
10	12 329	13 972.36	5.445	14 242.29	6.014	13 826.42	5.727	13 746.13	5.579
11	12 418	12 664.96	5.383	12 714.03	4.226	12 671.36	4.706	12 712.92	4.315
12	16 209	17 188.38	6.604	17 190.12	7.877	17 249.11	6.72	17 209.45	6.777
13	44 832	38 777.33	21.45	38 836.6	21.203	38 724.21	21.91	38 870.42	21.496
平均差值	—	−242.03	—	−210.98	—	−255.36	—	−245.628	—
平均时间	—	—	3.149	—	3.211	—	3.214	—	3.135

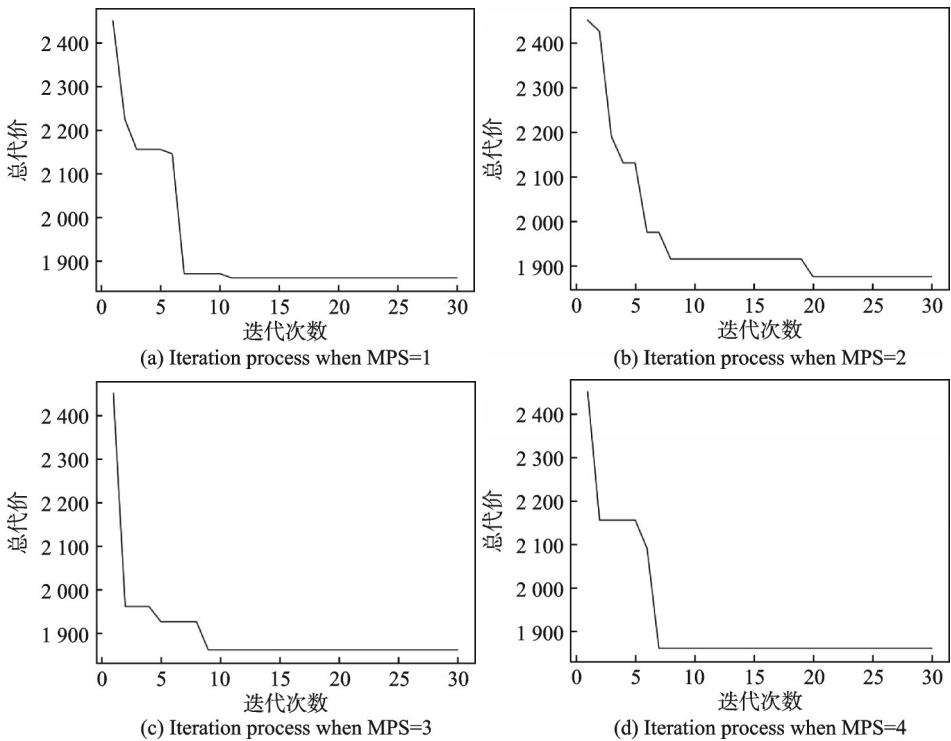


图 7 第 8 组 GA 迭代过程

Fig.7 Iteration process of GA for group 8

迭代优化过程,迭代次数 500 次。可以看出,对于大样本量而言,平均迭代 450 次左右可以得出较优解,运算时间在 20 s 左右,说明该算法在大样本量计算时也能有较强的运算效率,对于样本量大且要求实时性的运算有一定意义。

以总代价变化较大的数据集 Airland3 为例,将其具体着陆顺序及着陆时间可视化,可以更直观看

到优化前后航班着落顺序的变化与航班着陆时间的变化,如图 9 所示。

若按照 TTS 进行着陆,那么航班 9、航班 10、航班 11 均不能在最佳着陆时间降落。对其进行适当的排序后,航班 11 与航班 9 可以在最优着陆时间降落。所花费的代价也从 1 730 减少到 820。

为了尝试不同算法与 DPALO 结合的效果,本



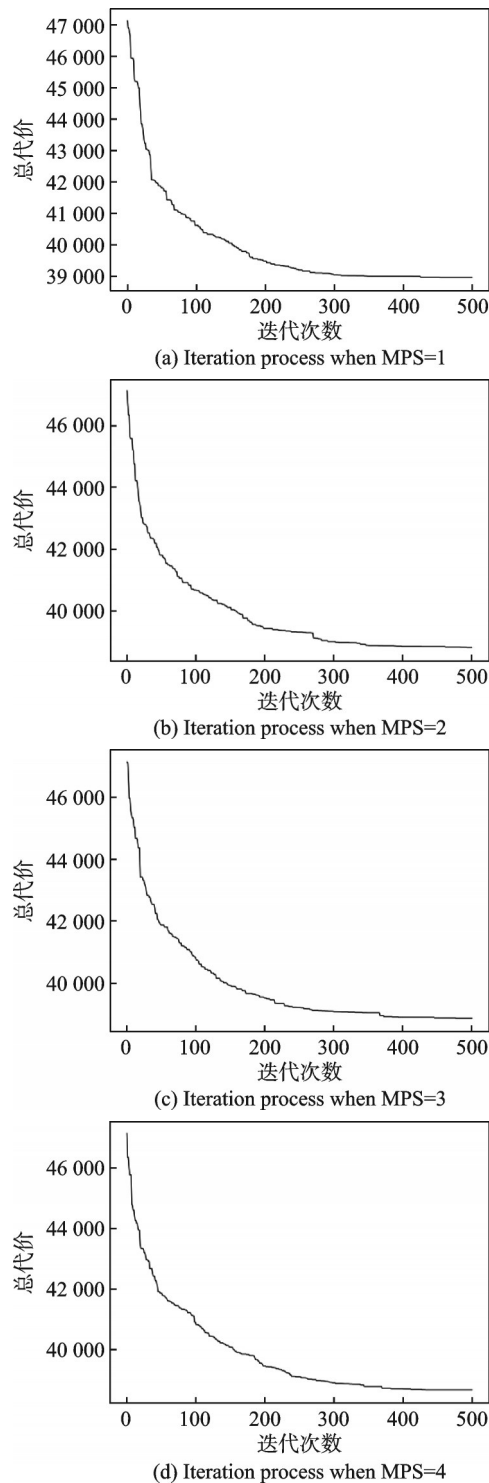


图8 第13组GA迭代过程

Fig.8 Iteration process of GA for group 13

文使用最大位置偏移的粒子群算法对航班着陆顺序进行优化,粒子总数为128,如表6所示,分别为DPALO+PSO在MPS=1、2、3和4时的计算结果,其中当MPS=3时PSO的平均差值最小,为77.457。算法平均运算时间不超过2.8 s,对于小运算量(前8组)运算速度快,且运算精度基本可以达到BKV,但对于大运算量(后5组),效果略有下降,计算精度有待提升,500架航班的情况下(13

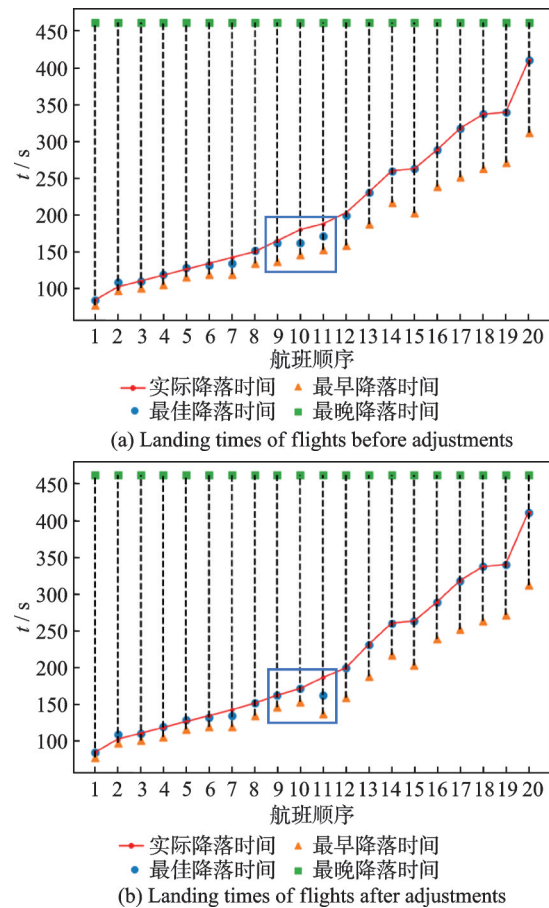


图9 航班顺序调整前后对比

Fig.9 Comparison of flight orders before and after adjustments

组),运算时间在10 s左右。DPALO+PSO运算结果与DPALO+GA的运算结果相比,认为遗传算法更适用于此问题的求解。粒子群算法虽在运算时间上更快,但整体效果不如遗传算法,后续仍使用DPALO+GA,MPS=3的结果进行后续比较,结果如表5所示。

### 2.3 对比实验

本文选取滚动时域-混合粒子群优化-局部搜索算法(Rolling horizon framework hybrid particle swarm optimization local search algorithm, RH-HPSO-LS)<sup>[6]</sup>、CSA+FEB<sup>[7]</sup>、DALP<sup>[9]</sup>、BA<sup>[10]</sup>、CAO<sup>[11]</sup>5个方法与DPALO+GA进行比较,比较结果见表6。

可以看出,DPALO+GA相较于现有的算法均有不同程度的提升,尤其是在运算时间方面提升巨大。DPALO+GA的平均差值相较于BA和DALP分别提升了127.54%和108.44%,运算速度方面,DPALO+GA的平均运算时间相较于CSA+FEB、BA、CAO、DALP和RH-HPSO-LS分别提升了76.88%、99.65%、89.11%、87.83%和78.28%,计算效率最高。

表 5 DPALO+PSO 运算结果

Table 5 Results of DPALO+PSO operations

组别	BKV	PSA								GA	
		MPS=1		MPS=2		MPS=3		MPS=4		MPS=3	
		$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$
1	700	700	0.004	700	0.006	700	0.006	700	0.004	700	0.004
2	1 480	1 484	0.04	1 482	0.054	1 482	0.046	1 484	0.045	1 480	0.028
3	820	820	0.58	820	0.078	820	0.058	820	0.063	820	0.025
4	2 520	2 520	0.004	2 520	0.006	2 520	0.006	2 520	0.004	2 520	0.004
5	3 100	3 166.5	0.119	3 166.5	0.155	3 244	0.159	3 143.5	0.126	3 100	0.043
6	24 442	24 442	0.004	24 442	0.006	24 442	0.006	24 442	0.005	24 442	0.004
7	1 550	1 550	0.005	1 550	0.007	1 550	0.007	1 550	0.005	1 550	0.005
8	1 950	1 863.75	0.157	1 865	0.159	1 863.25	0.185	1 862	0.145	1 863.75	0.056
9	5 611	5 708.37	3.067	5 710.09	3.323	5 675.01	3.223	5 723.88	3.052	5 694.52	2.541
10	12 329	14 810.64	4.511	15 479.85	4.709	14 799.94	4.615	14 886.66	4.511	13 826.42	5.727
11	12 418	12 935.45	5.457	12 944.06	6.216	12 922.42	5.506	12 943.65	5.414	12 671.36	4.706
12	16 209	17 659.06	6.824	17 720.9	7.877	17 626.09	6.849	17 636.86	6.908	17 249.11	6.72
13	44 832	41 361.61	12.985	41 335.88	13.784	41 323.23	12.052	41 347.15	11.938	38 724.21	21.91
平均 差值	—	81.568	—	136.56	—	77.457	—	84.515	—	−255.36	—
平均 时间	—	—	2.597	—	2.798	—	2.517	—	2.478	—	3.214

表 6 各算法结果对比

Table 6 Results of comparison of each algorithm

组别	BKV	DPALO+GA		CSA+FEB		BA		CAO		DALP		RH-HPSO-LS	
		$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$	$C_T$	$t/s$
1	700	700	0.004	700	0.001	700	60	700	0.002 5	740	0.1	700	0.01
2	1 480	1 480	0.028	1 480	0.052	1 480	90	1 480	0.003 4	1 870	0.5	1 480	0.03
3	820	820	0.025	820	0.005	820	99	820	0.004 2	1 440	0.5	820	0.07
4	2 520	2 520	0.004	2 520	0.006	2 520	95	2 520	0.004 3	2 670	0.9	2 520	0.1
5	3 100	3 100	0.043	3 100	0.006	3 100	100	3 100	0.194 5	6 130	1.3	3 100	0.13
6	24 442	24 442	0.004	24 442	0.006	24 442	274	24 442	0.052 1	24 442	0.6	24 442	0.22
7	1 550	1 550	0.005	1 550	0.015	1 550	79	1 550	0.085 7	3 974	1.5	1 550	1
8	1 950	1 863.75	0.056	1 920	0.622	2 655	287	1 995	1.711	2 915	3.1	1 950	0.76
9	5 611	5 694.52	2.541	5 618.6	2.301	6 425.9	554	5 611	6.272	7 800.9	5.7	5 611.7	5.8
10	12 329	13 826.42	5.727	12 539.7	8.362	16 508.9	925	12 457	29.314	17 726.1	13.5	12 292.2	18.8
11	12 418	12 671.36	4.706	12 544.7	15.771	14 488.4	1 417	12 439	31.717	14 152.4	22.6	12 418.3	15.6
12	16 209	17 249.11	6.72	16 428.3	24.344	20 032	20 111	16 262	47.328	25 049.2	43.2	16 122.2	35.1
13	44 832	38 724.21	21.91	37 625.7	128.76	45 294.1	5 825	38 573	267.125	58 392.6	249.5	37 064.1	115.1
平均 差值	—	−255.36	—	−513.2	—	927.3	—	−462.5	—	3 026.26	—	−606.9	—
平均 时间	—	—	3.214	—	13.9	—	911	—	29.5	—	26.4	—	14.8
前 8 组 平均时 间	—	—	0.021	—	0.089	—	135.5	—	0.257	—	1.063	—	0.29
后 5 组 平均时 间	—	—	8.322	—	35.908	—	5 766.4	—	76.351	—	66.9	—	38.08

对于小运算量而言,当样本量小于 50 时(前 8 组数据),DPALO 可以直接给出最优结果。在前 6 组数据中所有算法均达到了 BKV 值。第 8 组数据,DPALO+GA 的计算结果为 1 863.75,优于 BKV 的 1 950,且优于以往的其他算法。对于前 8 组数据 DPALO+GA 的运算平均时间仅为 0.037 s,大幅快于 BA 经典算法,快于 DALP 和 RH-HPSO-LS,与 CSA+FEB 和 CAO 的运算时间

相近。

对于高峰期大运算量的情况,当样本量大于 100 时(后 5 组数据),DPALO+GA 在保证运算精度的同时,相比于以往的算法运算效率大幅提升。DPALO+GA 后 5 组数据的平均运算时间为 8.322 s,相较于 CSA+FEB、BA、CAO、DALP 和 RH-HPSO-LS 分别提升了 76.82%、99.86%、89.1%、87.56% 和 78.15%。在大运算量的情况下

DPALO+GA算法的优势更能凸显出来。当样本量达到500时(第13组数据),DPALO的运算时间在20 s左右,而以往的其他算法所花费的运算时间均超过100 s在保证较高运算效率的同时也可以保证其计算精度 DPALO+GA 的计算结果优于BKV、BA 和 DALP,与 CAO、CSA+FEB 和 RH-HPSO-LS 的结果相近。

### 3 结 论

本文针对机场终端区航班着陆调度问题,建立了时间窗约束的动态规划优化模型。基于动态规划提出了确定性的DPALO算法求解固定顺序航班着陆问题。通过数学归纳优化了大量的运算过程,减少模型所计算的范围,节省了大量的运算时间,使算法运算时间达到毫秒级。通过与基于最大位置偏移的遗传算法结合,求解航班着陆总代价较小的着陆顺序,平均运算时间相较于CSA+FEB、BA、CAO、DALP 和 RH-HPSO-LS 分别提升了76.88%,99.65%,89.11%,87.83%和78.28%。平均精度相较于BA 和 DALP 分别提升了127.54%和108.44%,与现有模型最优解相差较小。

DPALO针对有时间窗约束的问题提出了一种新的求解策略,尤其对于样本量较大的排序问题时,DPALO的运算速度优势更加明显,具有一定的借鉴意义。下一步将对多跑道降落情形进行研究,如双跑道平行仪表进近,两条跑道降落,一条跑道起飞,并采用西部某枢纽机场高峰时段实际运行数据,研究进离港协同排序调度。

#### 参考文献:

- [1] 中国民用航空局.2023年民航行业发展统计公报[EB/OL].(2024-05-31).[http://www.caac.gov.cn/XXGK/XXGK/TJSJ/202405/t20240531\\_224333.html](http://www.caac.gov.cn/XXGK/XXGK/TJSJ/202405/t20240531_224333.html).
- [2] 李雨吟.基于大数据挖掘的机场延误分析及预测研究[D].南京:南京航空航天大学,2019.  
LI Yuyin. Research on airport delay analysis and prediction based on big data mining[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2019.
- [3] 张同荣,郭海鹏,魏志强.航空器尾流重新分类的实施效果量化评价方法[J].指挥信息系统与技术,2023,14(5):37-42,93.  
ZHANG Tongrong, GUO Haipeng, WEI Zhiqiang. Quantitative evaluation method for implementation effect of aircraft wake recategorization[J]. Command Information System and Technology, 2023, 14(5): 37-42, 93.
- [4] BALAKRISHNAN H, CHANDRAN B G. Algorithms for scheduling runway operations under constrained position shifting[J]. Operations Research, 2010, 58(6): 1650-1665.
- [5] XU B. An efficient ant colony algorithm based on wake-vortex modeling method for aircraft scheduling problem[J]. Journal of Computational and Applied Mathematics, 2017, 317: 157-170.
- [6] GIRISH B S. An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem[J]. Applied Soft Computing, 2016, 44: 200-221.
- [7] 冯小荣,高正达,王进,等.基于紧致子序列的航班着陆调度问题研究[J].北京航空航天大学学报,2024,50(8):2421-2431.  
FENG Xiaorong, GAO Zhengda, WANG Jin, et al. Research on aircraft landing scheduling problem based on compact subsequence[J]. Journal of Beijing University of Aeronautics and Astronautics, 2024, 50(8): 2421-2431.
- [8] BEASLEY J E, KRISHNAMOORTHY M, SHARAIHA Y M, et al. Scheduling aircraft landings—The static case[J]. Transportation Science, 2000, 34(2): 180-197.
- [9] BEASLEY J E, KRISHNAMOORTHY M, SHARAIHA Y M, et al. Displacement problem and dynamically scheduling aircraft landings[J]. Journal of the Operational Research Society, 2004, 55(1): 54-64.
- [10] PINOL H, BEASLEY J E. Scatter search and biometric algorithms for the aircraft landing problem[J]. European Journal of Operational Research, 2006, 171(2): 439-462.
- [11] YU S P, CAO X B, ZHANG J. A real-time schedule method for aircraft landing scheduling problem based on cellular automation[J]. Applied Soft Computing, 2011, 11(4): 3485-3493.
- [12] 许军海,何长青,王昭.机场终端区飞机排序的遗传算法[J].信息系统工程,2013(10):132-134.  
XU Junhai, HE Changqing, WANG Zhao. Genetic algorithm for aircraft sequencing in airport terminal area[J]. China CIO News, 2013(10): 132-134.
- [13] 王满超.动态飞机着陆调度及其经验粒子群算法研究[D].天津:中国民航大学,2015.  
WANG Manchao. Research on dynamic aircraft landing scheduling and empirical particle swarm optimization algorithm[D]. Tianjin: Civil Aviation University of China, 2015.
- [14] CHEN X D, YU H, CAO K, et al. Uncertainty-aware flight scheduling for airport throughput and flight delay optimization[J]. IEEE Transactions on Aerospace and Electronic Systems, 2020, 56(2): 853-862.



- [15] RODRÍGUEZ-DÍAZ A, ADENSO-DÍAZ B, GONZÁLEZ-TORRE P L. Improving aircraft approach operations taking into account noise and fuel consumption[J]. *Journal of Air Transport Management*, 2019, 77: 46-56.
- [16] 张军峰, 游录宝, 周铭, 等. 基于点融合系统的多目标进场排序与调度[J]. *北京航空航天大学学报*, 2023, 49(1): 66-73.  
ZHANG Junfeng, YOU Lubao, ZHOU Ming, et al. Multi-objective arrival sequencing and scheduling based on point merge system[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2023, 49(1): 66-73.
- [17] JIANG H, LIU J X, ZHOU W S. Bi-level programming model for joint scheduling of arrival and departure flights based on traffic scenario[J]. *Transactions of Nanjing University of Aeronautics and Astronautics*, 2021, 38(4): 671-684.
- [18] 王莉莉, 熊子睿. 不确定条件下点融合系统进场排序研究[J]. *重庆交通大学学报(自然科学版)*, 2023, 42(6): 138-145, 154.  
WANG Lili, XIONG Zirui. Approach sequencing of PMS under uncertain conditions[J]. *Journal of Chongqing Jiaotong University (Natural Science)*, 2023, 42(6): 138-145, 154.
- [19] 王建忠, 丁小芹, 王树伟. 考虑潜在冲突的进离场航班协同排序[J]. *交通运输系统工程与信息*, 2023, 23(5): 312-320.  
WANG Jianzhong, DING Xiaoqin, WANG Shuwei. Collaborative sequencing of arrival and departure aircraft considering potential conflicts[J]. *Journal of Transportation Systems Engineering and Information Technology*, 2023, 23(5): 312-320.
- [20] PARK B S, LEE H T. Airport surface arrival and departure scheduling using extended first-come, first-served scheduler[J]. *Aerospace*, 2023, 11(1): 24.
- [21] 杜卓铭, 张军峰, 杨春苇. 基于分类与优化的进场航空器调度方法[J]. *南京航空航天大学学报*, 2023, 55(6): 1065-1071.  
DU Zhuoming, ZHANG Junfeng, YANG Chunwei. Arrival scheduling based on classification and optimization[J]. *Journal of Nanjing University of Aeronautics & Astronautics*, 2023, 55(6): 1065-1071.
- [22] 周鼎凯. 基于强化学习的终端区进离场航班排序[D]. 广汉: 中国民用航空飞行学院, 2024.  
ZHOU Dingkai. Scheduling of arrival and departure flights in terminal area based on reinforcement learning[D]. Guanghan: Civil Aviation Flight University of China, 2024.
- [23] MOSER I, HENDTLASS T. Solving dynamic single-runway aircraft landing problems with extremal optimisation[C]//*Proceedings of 2007 IEEE Symposium on Computational Intelligence in Scheduling*. Honolulu, USA: IEEE, 2007: 206-211.
- [24] SABAR N R, KENDALL G. An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem[J]. *Omega*, 2015, 56: 88-98.

(编辑:张蓓)