

doi: 103969/j. issn. 0490-6756. 2016. 03. 012

基于滚动时域的遗传-免疫算法优化航班着陆调度

陈文平, 梁文快, 李毅

(四川大学计算机学院, 成都 610065)

摘要: 航班调度问题一直是空中交通管制(ATC)中的一个复杂而具有重要意义的任务,而航班着陆问题(ALS)是其中的核心问题。航班着陆调度是NP-hard问题,具有规模大、约束条件多的特点。因此,为了有效合理地解决航班着陆问题,本文提出了基于滚动时域的遗传-免疫算法(RHC HGIA)的航班着陆调度算法。RHC HGIA主要从两个方面解决航班着陆问题,一方面根据设定的滚动时域长度与大小选择需要进行优化的待降落航班;另一方面对选择的待降落航班使用遗传-免疫算法进行优化并确定其实际着陆时间。经过优化后的航班组成新的航班降落序列,从该序列中选择实际着陆时间在给定时域范围内的航班进行着陆,重新设置滚动时域长度,选择待降落航班进行优化,直到所有待着陆航班都已着陆为止。本文仿真实验以某机场一天内的20架待着陆航班数据为基础,并在机场管制仿真系统中进行模拟仿真。仿真实验表明,与传统航班着陆调度算法(FCFS)相比,经过RHC_HGIA算法优化后的待着陆航班的额外成本有明显的降低。

关键词: 航班着陆排序; 滚动时域; 遗传算法; 免疫算法; 条件约束

中图分类号: TP319 **文献标识码:** A **文章编号:** 0490-6756(2016)02-0311-08

Genetic-immune algorithm based on receding horizon for aircraft landing scheduling problems

CHEN Wen-Ping, LIANG Wen-Kuai, LI Yi

(College of Computer Science, Sichuan University, Chengdu 610065, China)

Abstract: Flight scheduling problem has been a complex and key task for the air traffic control (ATC), and aircraft landing scheduling (ALS) problem is one of the core issues. ALS is a NP-hard problem with a large scale and multi-constraints characteristics. Thus, in order to solve the flight landing problem effectively and rationally, a flight landing scheduling algorithm based on receding horizon and genetic-immune algorithm (RHC_HGIA) is proposed. RHC_HGIA solves the problem of flight landing by two aspects mainly, one is that selecting the flights that are waiting to land and need to be optimized based on the receding horizon length and size which have been set; on the other hand, optimizing The selected flights which are waiting to land by using genetic-immune algorithm and determining actual landing time of them. the flights that have been optimized form a new flight landing sequence, selecting the flights from the sequence that the actual landing time of them in the field within a given time range to land. Then resetting receding horizon length and re-selecting the flights to be optimized until all pending landings have landed so far. In this paper, simulation is conducted in the airport control simulation system

收稿日期: 2015-03-12

基金项目: 国家高技术研究发展计划(863计划)(2013AA013902)

作者简介: 陈文平(1990-),男,四川资阳人,硕士研究生,研究方向为空中交通管理自动化管理。

通讯作者: 李毅. E-mail: liyi_ws@scu.edu.cn

on the base of an airport of 20 flights to be landing of one day. Simulation results show that, RHC_HGIA algorithm can solve ALS problem preferably, and comparing with traditional flights landing scheduling algorithm(FCFS), the extra costs of flight is reduced much more.

Key words: Aircraft Landing Scheduling; Receding Horizon Control; Genetic algorithm; Immune Algorithm; Constraints

1 引言

终端区航班着陆排序与调度 ASS (Area Flight Landing Scheduling And Scheduling) 是空中交通管理 ATC(Air Traffic Control)中最为重要的问题,在中国,随着民用航空的发展,航班延误问题日益严重,而航班延误问题一直是 ATC 中的一个难题.降低航班延误率的主要途径是通过对一系列待着陆航班确定其航班的实际着陆时间,以确定航班着陆序列 ALS(Flight Landing Sequence).ALS 还需要满足一系列的约束条件,两个最基本的约束条件是着陆时间窗以及相邻航班之间尾流间隔标准.由于 ALS 问题是多约束的 NP-hard 问题,因此,在解决该问题时主要有两个难题:一是很难找到全局最优的着陆序列;二是高计算复杂度.

目前,国内外大多数机场的航班调度策略主要是先来先服务 FCFS(First Come First Serve).FCFS 调度简单,但是却存在航班调度效率低,航班延误严重的问题.文献[1]针对单跑道的航班着陆问题,提出运用蚁群算法对航班着陆调度进行优化;文献[2]针对多跑道起降航班调度问题,采用带最大移动位置数 MPS(Maximum Position Shifting)和相对移动位置数 RPS(Relative Position Shifting)约束的蚁群算法对航班调度进行优化;文献[3]针对航班着陆排序问题提出了带约束处理的多目标演化算法;文献[4]在传统遗传算法的基础上加入了启发式的插入空闲时间的方法;文献[5]针对航班的离场排序问题建立了相应的离场排序优化模型,在原始遗传算法的基础上设计了求解模型的双码自适应遗传算法,在交叉概率和变异概率方面提出了自适应交叉概率和变异概率求解模型;文献[6]针对单跑道提出了一种混合整数 0-1 规划,并将此方法应用到多跑道,通过引入额外约束以此强化这些问题的线性规划,并且提出了基于树搜索的线性规划优化航班调度;文献[7]和文献[8]提出了滚动时域与启发式算法相结合的思想;文献[9]首次将元胞自动机 CA(Cellular Automaton)应用于航班着陆优化问题.

机场管制的实际操作中,航班着陆实时性很强,要求管制人员快速反应,选择最合适的航班编排策略,在目前的管制操作中,管制人员主要是靠管制经验进行航班编排,遵循先去先服务原则,虽然解决了航班调度问题,但是其编排策略并非最佳.为了解决航班最佳调度问题,本文提出了基于滚动时域的遗传-免疫算法(RHC_HGIA),本算法在真实的管制系统中进行仿真实验,实验结果表明 RHC_HGIA 算法在解决航班着陆调度问题上要优于 FCFS 算法,算法收敛速度较快,基本上能够满足机场调度实时性的要求,较好的解决了机场航班调度问题,可以为管制人员提供良好辅助决策支持.

2 航班着陆问题描述

2.1 航班进场过程

机场航班进场是一个相对复杂的过程,进场航班脱离航路后必须按规定高度进入终端区,对于机型不同的航班一般为其分配不同的高度层,进场航班按照标准仪表进场程序飞至起始进近定位点后开始仪表进近飞行,航班自起始进近定位点开始,按照一定的下降梯度调整高度和速度,直至以规定的速度、高度以及飞行姿态对准最后进近航段并开始最后进近直至降落跑道,下图反映了终端区内航班进场过程.

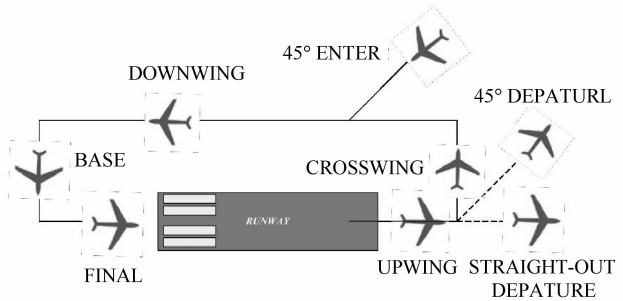


图 1 航班进离场过程

Fig. 1 Enter and departure process of aircraft

航班着陆排序遵循三边排序原则,在进入进近区时,采用 Y 型汇聚,将航班排列成一条线,并通

过调速和改变航向等管制手段, 保证航班安全到达机场降落区. 下图展示了飞机进场过程中常用的管制手段.

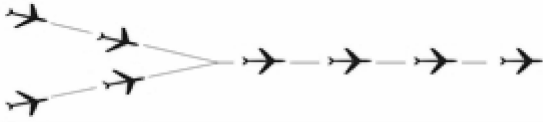


图 2 “Y”型汇聚排序
Fig. 2 “Y” gathering sorting



图 3 航班调速
Fig. 3 Aircraft speed adjusting

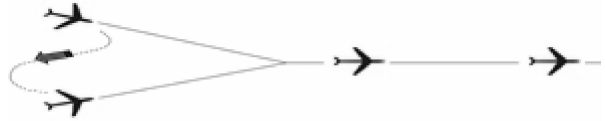


图 4 改变航向
Fig. 4 Track changing

2.2 航班着陆约束

在排序过程中, 航班队列进行调整时要满足一些约束, 其中最关键的一个约束就是尾流间隔标准, 即前后相邻两个航班之间必须保持一个最小的距离间隔(MLSI), 不同的飞机之间这种间隔是不一样的. 如表 1 所示, 文献[6]给出的最小安全间隔标准(转换成以时间为单位: s).

表 1 最小安全间隔标准
Tab. 1 Minimum safety standards

后机	前机			
	H	M	L	J
H	96	200	181	228
M	72	80	70	110
L	72	100	70	130
J	72	80	70	90

最小降落距离间隔是可变的; 可以通过改变不同类型的飞机的相互次序来降低飞机延迟.

2.3 降落时间窗约束

E_i : 航班 i 的最早着陆时间, $0 < E_i < 180$ (单位: s), 即限制飞机比其 T_i 时间最多提前 180 s;
 V_i : 航班 i 的实际着陆时间;

L_i : 航班 i 的最晚着陆时间, $0 < L_i < 300$ (单位: s), 即限制飞机比其 V_i 时间最多延迟 300 s;

实际着陆时间必须满足:

$$E_i \leq V_i \leq L_i (i = 1, \dots, M) \quad (1)$$

假如航班对 (i, j) , 航班 i 先于航班 j 着陆, 那么:

$$V_j - V_i \geq S_{ij} \quad (i = 1, \dots, M; j = 1, \dots, M; i \neq j) \quad (2)$$

这保证了航班 i 和航班 j 的最小安全间隔.

2.4 航班着陆调度排序数学模型

航班排序研究中, 优化目标一般有两种, 一种是总延误时间 T_{delay} 最小, 另一种是在单位时间内机场跑道降落的航空器数量 Ca 最多, 一般情况下, 两种优化结果效果是大体一致的. 本文采取了总延误时间最小作为评判标准. 以下给出航班着陆问题中需要定义的一些变量和模型.

M : 待降落航班数;

T_i : 航班 i 的预计着陆时间;

W_i : 航班 i 的尾流类型;

S_{ij} : 航班 i 和航班 j 的最小安全间隔时间, 航班 i 在航班 j 之前;

E_{xpen} : 表示表示所有待着陆航班的额外成本;

r : 跑道编号, $r \in [1, R]$;

g_i : 表示航班 i 先于预计着陆时间 T_i 着陆的单位成本系数, 取值在 $[0, 1]$ 之间;

h_i : 表示航班 i 后于预计着陆时间 T_i 着陆的单位成本系数, 取值在 $[0, 1]$ 之间;

航班 i 的时间窗为 $[E_i, L_i]$, 且 $E_i \leq T_i \leq L_i$. 航班 i 的多跑道航班排序需要优化的时间延误目标函数为

$$\text{Min}_{s(1), s(2), \dots, s(M)} E_{xpen} = \min \sum_{i=1}^M Z_{ir} \quad (3)$$

其中, Z_{ir} 表示航班 i 降落到跑道 r 的额外成本, Z_i 与航班 i 的实际着陆时间 V_i 与预计着陆时间 T_i 之差的平方成正比. 式(2)~式(4)表示如何计算航班 i 的额外成本, 参考文献[6].

$$Z_i = \begin{cases} g_i | T_i - V_i |, & \text{if } V_i \leq T_i \\ h_i | V_i - T_i |, & \text{if } V_i > T_i \end{cases} \quad (4)$$

3 基于滚动时域的遗传-免疫算法求解

3.1 滚动时域(RHC)

图 5 给出了滚动时域(Receding horizon sketch)算法的示意图. 其中, T_m 表示一个操作间

隔的大小(滚动时域的大小), N 表示滚动时域的长度, k 表示 k -th 滚动时域.

3.2 遗传-免疫算法(HGIA)

3.2.1 遗传算法(GA) 遗传算法(Genetic Algo-

rithm)是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型. 借助于自然遗传学的遗传算子进行组合交叉和变异, 产生出代表新的解集种群, 直到收敛得到近似最优解.

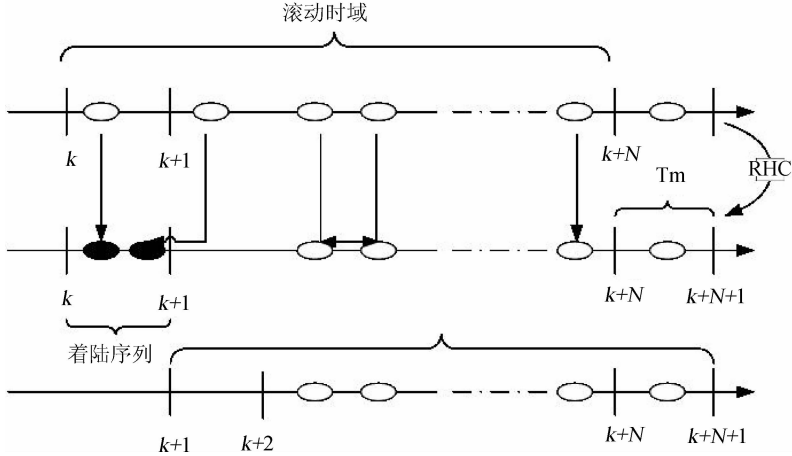


图 5 滚动时域示意图

Fig. 5 Receding horizon sketch

1) 目标函数

本文使用飞机序列延误时间最长作为目标函数

$$\max_{s(1), s(2), \dots, s(Na)} T_{\text{delay}} = \min \left\{ \sum_{i=1}^{Na} (STAi_k - ETAi_k)^2 \right\} \quad (5)$$

2) 遗传操作

选择: 采用最优保存策略.

交叉与变异: 采用单点交叉和单点突变. 并使用自适应方法调节交叉率和变异率, 避免算法的早熟现象

$$p_c = \frac{1}{1 + e^{-k_1 \lambda}}, p_m = 1 - \frac{1}{1 + e^{-k_2 \lambda}}; \quad (6)$$

$$k_1 > 0, k_2 > 0$$

其中, p_c 和 p_m 分别表示交叉概率和变异概率; $\lambda = F_{\text{max}} - \bar{F}$ 表示早熟程度; F_{max} 表示在一次遗传算法中个体的最大适应度, \bar{F} 为一次迭代中所有个体的平均适应度.

3.2.2 免疫算法(IA) 免疫算法是模仿生物免疫学和基因进化原理, 用人工方式构造的一类优化搜索算法, 是一种使用数学仿真方法对生物免疫过程的模拟技术, 是免疫计算的一种最重要的形式.

多样性: 通过识别抗体与抗原、抗体与抗体之间的独特性, 免疫系统产生的抗体具有多样性. 假设免疫系统含有 N 个抗体, 且每个抗体含有 M 个

基因, 等位基因数为 m , 如图 6 所示.

本文假设 $M = m$. j -th 基因的信息熵可表示为

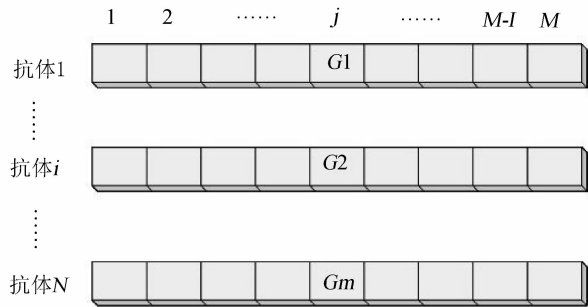


图 6 抗体中基因的信息熵的定义

Fig. 6 Definition of gene's entropy in antibody

亲和度: 抗体与抗原、抗体与抗体之间的结合强度被称为亲和度. 抗体 v 和抗体 w 的亲和度定义为

$$af_{v,w} = \frac{1}{1 + H(2)} \quad (7)$$

其中, $H(2)$ 表示抗体 v 和抗体 w 的信息熵; $af_{v,w}$ 的取值在 $[0, 1]$ 之间.

另一方面, 抗原与抗体 v 的亲和度 af_v 可表示为

$$af_v = \frac{1}{1 + opt_v} \quad (8)$$

其中, opt_v 表示抗原与抗体 v 的结合强度; af_v 的取值在 $[0, 1]$ 之间.

3.3 滚动时域遗传-免疫算法(RHC HGIA)

3.3.1 算法概述 上文中提到 ALS 问题是 NP 难问题, 在解决该问题时主要有两个难题: 一是很难找到全局最优的着陆序列; 二是高计算复杂度. 在机场管制的实际操作中, 航班着陆实时性很强, 它要求管制人员快速反应, 并选择最合适的航班编排策略, 而本文提出的滚动时域遗传-免疫算法在真实的管制系统中进行仿真实验, 实验结果表明该算法能很好的解决这个问题.

首先, RHC HGIA 算法分为两个阶段: 遗传算法阶段和免疫算法阶段. 遗传算法阶段的主要工作是产生抗原, 设定目标函数找出航班延误最大的飞机序列, 加入到抗原集合中; 免疫算法阶段是从抗原集合中随机选择一个抗原, 利用免疫算法产生抗体, 即使用免疫算法优化最大时间延误的飞机序列, 最终得到一个时间延误最小的飞机序列. 遗传-免疫算法中采用了可变的交叉率和变异率, 以记忆库(优秀抗体库)记录每次算法迭代产生的抗原和优秀抗体, 信息熵来保持抗体的多样性, 避免了陷入局部最优解. 其次, 引入 RHC, 将时间分割成时间段, 使得在每一次算法迭代计算过程中, 参与排序的飞机数量明显减少, 有效的减少算法的计算负担, 这也是对实时计算一种较合理的方法.

机序列, 加入到抗原集合中; 免疫算法阶段是从抗原集合中随机选择一个抗原, 利用免疫算法产生抗体, 即使用免疫算法优化最大时间延误的飞机序列, 最终得到一个时间延误最小的飞机序列. 遗传-免疫算法中采用了可变的交叉率和变异率, 以记忆库(优秀抗体库)记录每次算法迭代产生的抗原和优秀抗体, 信息熵来保持抗体的多样性, 避免了陷入局部最优解. 其次, 引入 RHC, 将时间分割成时间段, 使得在每一次算法迭代计算过程中, 参与排序的飞机数量明显减少, 有效的减少算法的计算负担, 这也是对实时计算一种较合理的方法.

3.3.2 算法流程图 根据 RHC 以及 HGIA 算法的特点, RHC_HGIA 算法的流程图, 如图 7 所示:

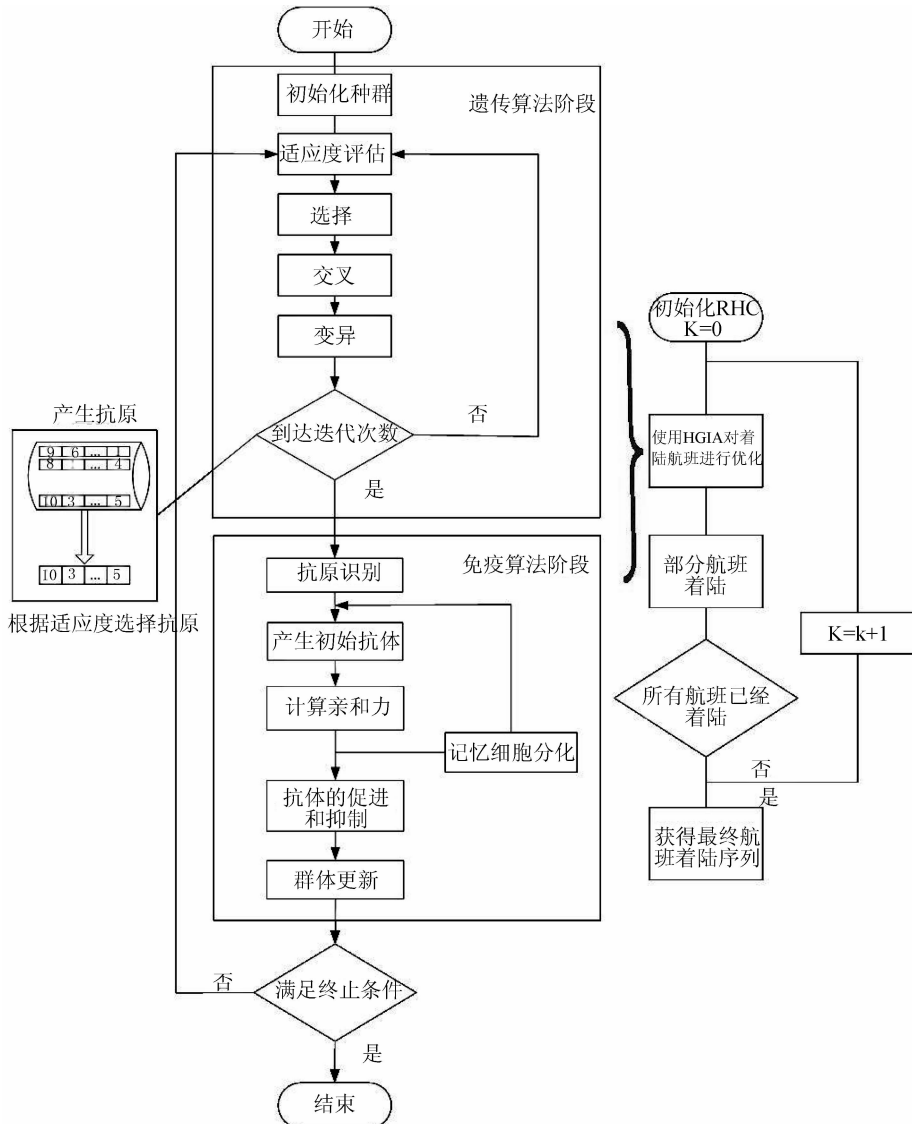


图 7 RHC HGIA 流程图
Fig.7 RHC HGIA sketch

3.3.3 编码策略 抗体编码(抗原)一般包括二进制编码和十进制编码(整数编码),本文选择十进制编码策略. 每个抗体(抗原)对应一个可能的航班着陆序列,每个抗体包含若干个基因,每架航班对应一个基因,

对于一个给定的航班着陆时间序列 (T_1, \dots, T_M) , 相应的航班着陆序列为 (b_1, \dots, b_M) , 它是航班序列 $(1, \dots, M)$ 的一个排列, 相邻航班之间的着陆时间差为 $q_i = V_{b_i} - V_{b_{i-1}}$, 且 q_i 必须满足:

$$q_i \geq S_{b_i, b_{i-1}} \quad (i = 1, \dots, M) \quad (9)$$

即两架航班之间的时间间隔需要满足最小安全间隔. 抗体的编码可表示为: (b_1, \dots, b_M) , 且其约束条件是:

$$\begin{cases} V_{b_1} = V_{b_1} \quad (i = 1) \\ V_{b_j} = q_i - S_{b_i, b_{i-1}} + \max(V_{b_j} + S_{b_j, b_i}) \\ (j = 1, \dots, i - 1, i > 1) \end{cases} \quad (10)$$

抗原的适应度函数为

$$\begin{cases} f_i = (V_{ir} - T_{ir})^2 |V_{ir} - T_{ir}| < L_i \\ f_i = (V_{ir} - T_{ir})^4 |V_{ir} - T_{ir}| < L_i \end{cases} \quad (11)$$

$$F = \sum_{i=1}^M f_i \quad (12)$$

$$A = \text{Max}(Fk), k \in [1, v] \quad (13)$$

4 实验结果与分析

本文使用某机场一天内的 20 架待着陆航班进行仿真实验, 仿真实验平台是 Visual Studio2008, Matlab R2011b, 两台 3.4GHz CPU 的 PC 机以及 CDM 空管仿真系统. 本文用 RHC_HGIA 与 FCFS、IA、基于滚动时域和克隆选择算法(RHC_CSA)以及克隆选择算法(CSA)进行对比实验. 实验结果如表 2 (E_i, T_i, L_i 单位为 s; V_i 单位为元):

由表 2 可知, 使用 RHC_HGIA 算法优化后的航班着陆额外成本要明显小于其余四个算法得到的航班着陆额外成本, 且与传统调度算法 FCFS 相比, 航班着陆额外成本减少了 28.79%, 由此可验证 RHC_HGIA 算法在优化航班着陆调度序列问题上要优于 FCFS、IA、RHC_CSA 以及 CSA 算法.

图 8 表示 5 种算法对 20 架待着陆航班优化后的结果图, 图 9 表示 5 种算法与最终的航班着陆额外成本关系图.

表 2 5 种算法的实验对比结果

Tab.2 Results of 5 different algorithm experiments

No.	待着陆航班					W_i	FCFS	CSA	RHCCSA	IA	RHCHGIA
	E_i	T_i	L_i	g_i	h_i		V_i	V_i	V_i	V_i	V_i
1	0	17	517	0.5	0.5	J	17	17	17	0	15
2	0	100	600	0.2	0.3	M	100	97	100	80	95
3	0	217	617	0.2	0.7	H	217	277	217	152	163
4	0	279	979	0.5	0.6	L	340	343	279	333	295
5	143	343	743	0.3	0.6	J	427	349	350	463	383
6	324	424	1424	0.6	0.2	H	521	445	439	535	464
7	217	517	717	0.9	0.8	L	681	626	533	716	521
8	210	610	1010	0.2	0.5	H	753	698	610	788	565
9	488	688	1588	0.3	0.1	L	871	850	715	969	614
10	470	770	2270	0.5	0.4	J	1064	879	799	1099	887
11	50	850	1850	0.7	0.7	J	1154	1009	900	1189	913
12	205	905	2005	0.7	0.3	M	1234	1179	946	1269	1025
13	1026	1126	1926	0.4	0.2	M	1314	1259	1226	1349	1058
14	741	1541	2241	0.7	0.2	H	1541	1331.49	1541	1421	1450
15	1254	1554	1854	0.3	0.8	J	1669	1554	1677	1649	1691
16	990	1790	2590	0.1	0.7	L	1800	1629	1722	1719	1727
17	1048	1848	2748	0.5	0.8	J	1848	1759	1741	1849	1817
18	1451	2051	2651	0.5	0.8	H	2051	2031	1813	1921	1889
19	1669	2069	2469	0.6	0.5	M	2171	2101	1848	2121	2089
20	1872	2172	3000	0.9	0.6	L	22331	2103	1918	2191	2159
额外成本(元)							886.2	897.857	722.5	1981.1	631.1

在图 8 中,横坐标表示算法优化后的航班的序号,纵坐标表示对应航班的实际着陆时间;在图 9 中,横坐标表示算法编号,从 1~5 分别对应 FCFS、CSA、RHC_CSA、IA、RHC_HGIA,纵坐标对应航班着陆额外成本。

本算法中除了需要输入数据(待着陆航班)和一些约束条件之外,还需要确定变异概率 p_m ,交叉概率 p_c ,种群大小 N_P ,记忆库大小 N_M ,算法的迭代次数 IterationNum,滚动时域大小 T_m 以及滚动时域长度 N . 本文设定 $p_m = 0.05$, $p_c = 0.8$, $N_P = 250$, $N_M = 100$. IterationNum、 T_m 以及 N 不仅影响算法的效率,而且还影响最终的航班着陆额外成本,因此这三个参数必须由实验确定其值。

图 7 表示 RHC_HGIA、IA、RHC_CSA 与 CSA 算法所得的航班着陆额外成本与算法的迭代次数 IterationNum 的对应关系. 从图可知,当迭代次数为 100 时,四种算法所得航班额外成本达到最小. 因此,本文选择算法的迭代次数 IterationNum = 100;图 11、12 分别表示 RHC_HGIA、RHC_CSA 算法中航班额外成本与滚动时域大小以及滚动时域长度的关系图. 从图 11 和图 12 可以看出,在 $T = 100$, $N = 3$ 或 4 时,航班着陆额外成本相对最小,因此,本文选择滚动时域大小 $T_m = 100$,滚动时域长度 $N = 3$.

5 结束语

本文使用 RHC_HGIA 算法与 FCFS、IA、RHC_CSA 以及 CSA 四种算法进行了实验对比,并且将 RHC_HGIA 算法应用于空管仿真系统进

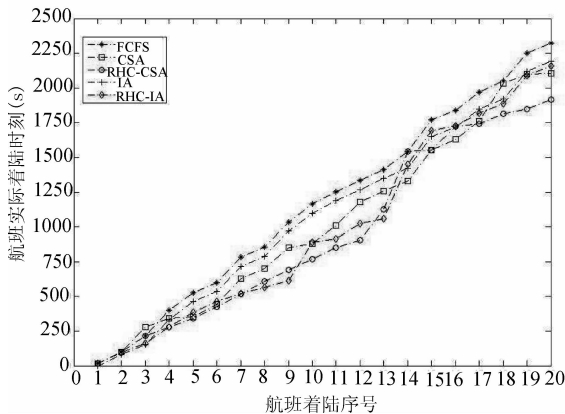


图 8 5 种算法的对比图

Fig. 8 Comparison diagram of 5 algorithms

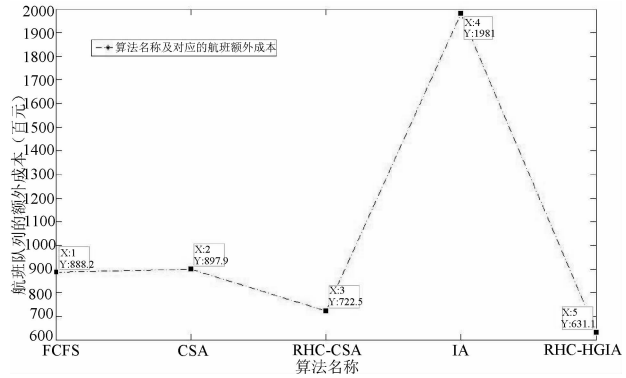


图 9 5 种算法与航班额外成本的对应关系图

Fig. 9 Relationship diagram between 5 algorithms and Extra cost of aircraft

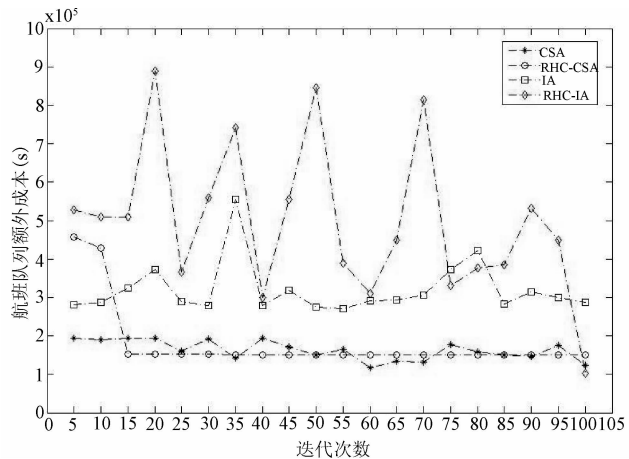


图 10 四种算法所得的航班着陆额外成本与算法的迭代次数关系图

Fig. 10 Iterations and extra cost of four different algorithms

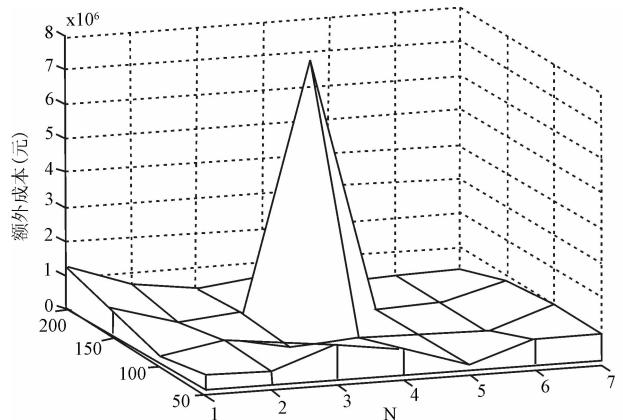


图 11 RHC HGIA 算法中航班额外成本与滚动时域大小以及滚动时域长度的关系图

Fig. 11 Relationship between scale and length of receding horizon and extra cost in RHC HGIA

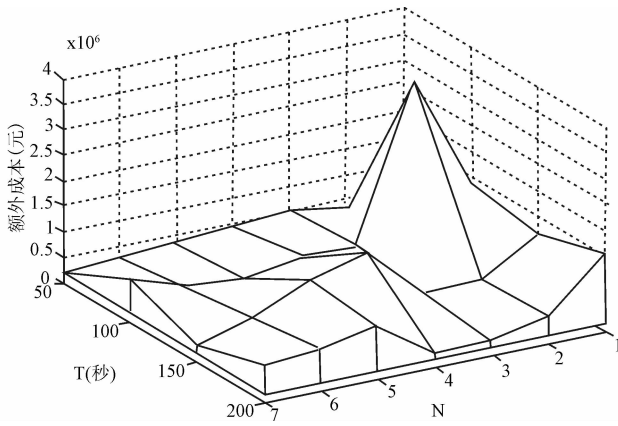


图 12 RHC CSA 算法中航班额外成本与滚动时域大小以及滚动时域长度的关系图

Fig. 12 Relationship between scale and length of receding horizon and extra cost in RHC CSA

行模拟仿真实验,实验结果验证了 RHC_HGIA 算法在解决航班着陆调度问题上要优于其他四种算法,且相对 FCFS 算法航班额外成本降低了 28.79%。因此,针对 ALS 问题,RHC_HGIA 算法是一种有效的算法。接下来的工作一方面是将 RHC_HGIA 算法应用在现实机场的管制系统中,以进一步验证其实际可行性;另一方面是从算法的执行效率着手,通过改进 IA 算法以提高算法的执行效率。

参考文献:

[1] 李志荣,张兆宁. 基于蚁群算法的航班着陆排序[J]. 交通运输工程与信息学报, 2006, 4(2): 66.

- [2] 刘丹,韩松臣,舒旒. 多跑道起降航班排序模型和算法研究[J]. 武汉理工大学学报:信息与管理工程版, 2011, 33(1): 27.
- [3] 杨秋辉,游志胜,冯子亮,等. 一种改进的基于遗传算法的多跑道达到飞机调度[J]. 四川大学学报:工程科学版, 2006, 38(2): 141.
- [4] 王来军,始忠科. 航班离场排序问题的遗传算法设计[J]. 系统工程理论与实践, 2005, 25(9): 119.
- [5] Beasley J E, Krishnamoorthy M, Sharaiha Y M, *et al.* Scheduling aircraft landings-the static case[J]. *Trans Sci*, 2000, 34(2): 180.
- [6] Hu X B, Chen W H. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling [J]. *Eng Appl Artif Intel*, 2005, 18(5): 633.
- [7] Yu S P, Cao X B, Zhang J. A real-time schedule method for aircraft landing scheduling problem based on cellular automation[J]. *ASC*, 2011, 11(4): 3485.
- [8] John E. Fuzzy reasoning based sequencing of arrival aircraft in the terminal area [C]//*Proceedings of AI A A Guidance, Navigation and Control Conference*. New Orleans, LA: American Institute of Aeronautics and Astronautics, Inc., 1997.
- [9] Jiao L C, Wang L. A Novel Genetic Algorithm Based on Immunity[J]. *IEEE SMC*, 2009, 30(5): 552.
- [10] Chun J S, Kim M K, Jung H K. Shape optimization of electromagnetic devices using immune algorithm [J]. *IEEE Trans on Mag*, 1997, 33(2): 1876.