

基于 RHC 的航班进港排序多局部搜索 GA 优化算法

张玉州¹, 陈文莉¹, 江克勤^{1,2}, 王一宾¹

(1. 安庆师范学院计算机与信息学院, 安徽安庆 246133; 2. 华中科技大学控制科学与工程系, 湖北武汉 430074)

摘要:为解决航班进港排序难以应对动态环境下航班信息变化的问题, 首先建立了一种基于滚动时域控制的机场进港航班动态排序优化模型, 并将存储时域上的航班进港优化序列作为后继时域航班排序的启发信息; 然后提出了一种基于滚动时域控制的多局部搜索遗传算法对问题求解, 给出了一种以存储的航班优化信息为基础的种群初始化策略; 针对遗传算法易陷入局部最优的不足以及单一局部搜索难以取得很好收敛效果与满意解的现状, 提出的多局部搜索遗传算法, 在不同时期采用不同的局部搜索策略, 其中定向局部搜索根据个体、基准个体的基因结构和适应性调整最大搜索速度; 最后, 大量仿真实验结果证明了所提模型与算法的有效性以及算法的稳定性, 并由实验结果得出了航班排序问题的相关特性结论。

关键词:进港排序; RHC; 多局部搜索遗传算法; 航班优化序列

中图分类号: V355, TP18 **文献标识码:** A doi:10.3969/j.issn.0253-2778.2015.01.012

引用格式: Zhang Yuzhou, Chen Wenli, Jiang Keqin, et al. Genetic algorithm with multiple local searches based on receding horizon control for aircraft arrival sequencing and scheduling[J]. Journal of University of Science and Technology of China, 2015, 45(1): 75-86.

张玉州, 陈文莉, 江克勤, 等. 基于 RHC 的航班进港排序多局部搜索 GA 优化算法[J]. 中国科学技术大学学报, 2015, 45(1): 75-86.

Genetic algorithm with multiple local searches based on receding horizon control for aircraft arrival sequencing and scheduling

ZHANG Yuzhou¹, CHEN Wenli¹, JIANG Keqin^{1,2}, WANG Yibin¹

(1. School of Computer and Information, Anqing Normal University, Anqing 246133, China;

2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In order to solve the problem that aircraft arrival scheduling and sequencing (ASS) has difficulty when meeting changes of the aircraft messages in dynamic environments, an optimization model based on receding horizon control (RHC) was proposed for the dynamic ASS problems, and optimized sequence of the aircraft in a horizon was saved as a heuristic message for ASS in the next horizon. Then an RHC-based genetic algorithm (GA) with multiple local searches (RHC-MLSGA) was designed to solve the model, and an initialization strategy for population was given on the basis of the saved optimization message. Due to the fact that existing GA may easily fall into local peak and that GA with single local search can not obtain remarkable performance in convergence and satisfactory solution, different local searches were

收稿日期: 2014-05-16; 修回日期: 2014-10-10

基金项目: 国家自然科学基金 (61272071), 安徽省自然科学基金 (1408085MF131), 安徽省高校自然科学基金重点项目 (KJ2013A177, KJ2014A140) 资助.

作者简介: 张玉州 (通讯作者), 男, 1976 生, 硕士/副教授. 研究方向: 智能交通和进化算法研究. E-mail: yzhzhang@mail.ustc.edu.cn

employed at different stages in the proposed RHC-MLSGA, among which directed local search adjusts the individual maximum searching speed according to gene structures and fitness of the individual and benchmark individual. A large number of experiments show the validity of the proposed model and algorithm and the stability of the algorithm when solving ASS problems in dynamic environments. Several conclusions about the characteristics of ASS problem's have been drawn from results of the experiments as well.

Key words: arrival sequencing and scheduling; receding horizon control; genetic algorithm with multiple local searches; aircraft optimization sequence

0 引言

随着航空运输业的高速发展,空中交通流量激增,航班延误日趋严重,机场成为空中交通管制的主要瓶颈.目前,机场航班进港排序(arrival sequencing and scheduling, ASS)是解决这一问题的主要方法,其目标是提高机场吞吐量、减少航班延误^[1].

FCFS (first come first serve)按航班到达顺序分配着陆时间,该策略忽略了诸如航班着陆时间间隔(landing time interval, LTI)非对称性等重要因素^[2],从而不能有效地降低航班的延误.所以确定合理的航班着陆次序和着陆时间,是 ASS 的研究重点.国内外学者对 ASS 进行了深入研究,提出了相应的模型及求解算法^[3-7],这些工作都是基于静态环境开展的,即所有航班都按计划到达.然而,现实航空运营过程中的航班信息往往动态变化,如航班的取消、到达时间的更改以及未知航班的到达,所以需要进行实时的航班在线优化排序.学者们建立了动态 ASS 问题模型,并设计了问题的求解算法^[8-10],但这些研究增加了问题的规模,影响算法的求解效果,且忽略了随着时间推移而发生的航班信息变化^[8],在线优化排序算法的频繁启动^[10],相关参数对算法性能的影响未作讨论分析^[8-9].滚动时域控制(receding horizon control, RHC)作为一种动态环境下在线优化有效控制技术, Hu 较早将其引入了动态 ASS 问题模型^[11],并设计基于二进制编码的在线优化遗传算法(genetic algorithm, GA)^[12],使用 Monte Carlo 方法对多种动态情况及参数变化情况进行了仿真,验证了基于 RHC 的动态航班排序算法的鲁棒性^[11-12];其他学者也分别设计了基于 RHC 的算法对动态 ASS 问题进行求解^[13-15]. RHC 策略中相邻时域中的公共航班经在线优化排序后,通常具有相同或相近的次序,而目前基于 RHC 的 ASS 问题研究,各时域中的航班优化排序独立进

行,无信息共享^[11-15],势必影响航班着陆次序的优化效果.加强时域间的信息交流,可进一步提高基于 RHC 的动态 ASS 问题的优化效果.

ASS 问题是典型的 NP-C 组合优化问题,目前存在的确定性算法如线性与非线性规划等难以满足航空运营的实时性需要. GA 以其良好的全局搜索能力而被用来解决 ASS 问题^[8-9, 12, 15],但较弱的局部搜索特性容易导致 GA 陷入局部最优^[16].有学者从参数以及遗传算子等方面改进了 GA^[17-19],然而 GA 自身的改进无法完全克服局部早熟的缺陷,引入局部搜索机制,构成一种混合 GA,是提高算法求解性能的一种有效手段^[20].目前混合 GA 通常采用单一的局部搜索策略^[21-22],并且个体局部搜索的步长是固定的,使用一种局部搜索策略具有一定的局限性,对于复杂问题,单一局部搜索机制难以使得混合算法在有限代数内取得较好的解^[23].

综合上述各因素,本文以降低进港航班总延误时间为目标,建立了基于 RHC 的动态 ASS 优化模型.针对动态 ASS 的特点和复杂性,设计了一种基于 RHC 的多局部搜索 GA(RHC-based GA with multiple local searches, RHC_MLSGA)对问题模型进行求解,包括时域间航班优化序列信息的传递、随机局部搜索以及定向局部搜索的设计.最后通过 Monte Carlo 实验对问题模型以及所提算法进行了仿真,实验结果表明了模型和 RHC_MLSGA 的有效性.

1 动态 ASS 优化模型描述

1.1 RHC 策略

RHC 策略是一种具有良好鲁棒性的在线优化控制方法^[24],该策略向前展望 N_{RHC} 步,并由 N_{RHC} 个时间间隔组成当前时域,在线优化处理的对象处于该时域中,并且结果中只有落在第一个时间间隔中的对象被实际执行,其余对象留至下一时域.更新对象信息,继续进行新时域的在线优化.基于 RHC 框架下的动态 ASS 过程如图 1 所示,令时域 k 的起

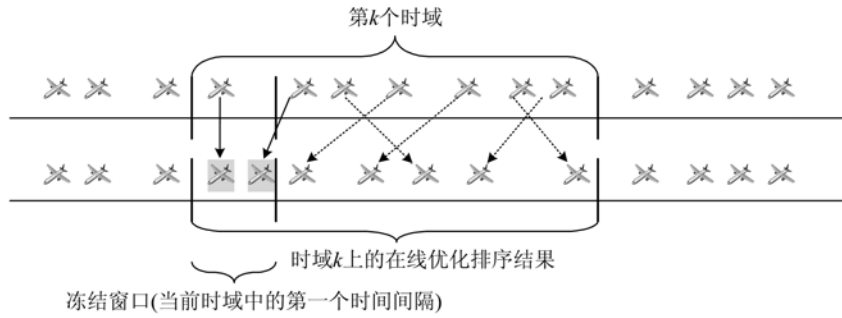


图 1 基于 RHC 的 ASS 过程

Fig. 1 Process of ASS based on RHC

始时间为 $t(k)$, 则该时域构成时间窗 $[t(k), t(k) + N_{RHC} T_{OI}]$, 其中 T_{OI} 为时间间隔长度. 计划着陆时间落在 $[t(k), t(k) + N_{RHC} T_{OI}]$ 中的航班被优化排序, 优化序列中只有分配着陆时间落在 $[t(k), t(k) + T_{OI}]$ 中的航班被安排着陆, 其余航班等待下一时域的排序, 所以该时间段又称为冻结窗口. 更新剩余航班信息, 执行时域 $k+1$ 上的在线优化排序.

1.2 动态 ASS 优化模型

1.2.1 变量定义

N_{AC} : 时间段 $[0, T]$ 内目标机场 Z 进港的飞机数; $PLT(i)$: 第 i 架飞机的计划达到和着陆时间; $ALT(i)$: 第 i 架飞机的分配着陆时间; $ADT(i)$: 第 i 架飞机的空中等待时间; $LTI(f_1, f_2)$: 相继着陆的两架飞机最小安全尾流间隔时间, 参数 (f_1, f_2) 表示 f_2 在 f_1 之后着陆; $U(i, t)$: t 时刻第 i 架飞机的跑道服务时间; k : 当前时域的序号; $t(k)$: 时域 k 的起始时间; T_{OI} : 时间间隔的长度; N_{RHC} : 时域上时间间隔总数; $WS(k)$: 时域 k 上待排序的航班集合; $(\cdot | k)$: 时域 k 中所涉及的变量和计算; $N_{AC}(\cdot | k)$: 计划着陆时间落在第 k 个时域内的航班数; $(k+j | k)$: 时域 k 上第 j 个时间间隔, $j=1, 2, \dots, N_{RHC}$; $left_queue(k)$: 时域 k 上优化后没有着陆的航班序列.

1.2.2 模型的建立

ASS 问题的任务是优化所有进港航班的着陆次序, 分配着陆时间, 使总延误时间最小化. 第 i 架飞机的空中延误时间为:

$$ADT(i) = ALT(i) - PLT(i) \quad (1)$$

从而 ASS 问题的目标函数为:

$$\min \sum_{i=1}^{N_{AC}} ADT(i) \quad (2)$$

下面分时域对航班排序过程进行讨论, 因为每

个时域上的航班在线排序是重复进行的, 所以仅讨论时域 k 上的排序过程. 更新当前需要着陆航班信息, 按计划着陆时间构造时域 k 上的待排序航班集合 $WS(k)$. 若 $k > 1$, 则需合并上一时域的剩余航班, 即 $WS(k) = WS(k) \cup left_queue(k-1)$. 调整后的第一架飞机, 若 $k=1$, 则跑道空闲, 可直接降落, 所以有 $ALT(1 | 1) = PLT(1 | 1)$; 当 $k > 1$ 时, 若跑道为空, 即时域 $k-1$ 上最后着陆航班已脱离跑道, 记该航班为 $lf | k-1$, 并且它们满足最小安全尾流间隔, 即 $PLT(1 | k) \geq ALT(lf | k-1) + \max(LTI(lf | k-1, 1 | k), U(lf | k-1, t))$, 则时域 k 上的第一架飞机按计划着陆时间降落, 此时有 $ALT(1 | k) = PLT(1 | k)$; 否则该飞机产生空中延误, 使得 $ALT(1 | k) = ALT(lf | k-1) + \max(LTI(lf | k-1, 1 | k), U(lf | k-1, t))$, 所以时域 k 上的第一架飞机分配着陆时间为:

$$ALT(1 | k) = \begin{cases} PLT(1 | 1), & k = 1 \\ \max(PLT(1 | k), ALT(lf | k-1) + \max(LTI(lf | k-1, 1 | k), U(lf | k-1, t))), & k > 1 \end{cases} \quad (3)$$

时域 k 上的第 $i (i > 1)$ 架飞机可依据上述讨论进行类似分析, 其分配着陆时间为:

$$ALT(i | k) = \max(PLT(i | k), ALT(i-1 | k) + \max(LTI(i-1 | k, i | k), U(i-1 | k, t))) \quad (4)$$

所以经过调整后时域 k 上的航班 j 的空中延误时间为:

$$ADT(j | k) = ALT(j | k) - PLT(j | k) \quad (5)$$

式中, $j=1, 2, \dots, N_{AC}(\cdot | k)$. 传统模型时域 k 上的目标函数为该时域上航班总延误时间的最小化, 即

$$\min \sum_{j=1}^{N_{AC}(k)} ADT(j|k) \quad (6)$$

由于现实的航班信息会动态变化,计划到达时间靠后的航班信息可信度降低,其延误时间对以式(6)为目标的贡献将会弱化,所以对航班设置权重,从而式(6)更替为:

$$\min \sum_{j=1}^{N_{AC}(k)} c(j|k) ADT(j|k) \quad (7)$$

式中, $c(j|k)$ 为航班 j 的权重系数,在时域 k 上随着计划到达时间的推移,该航班的权重系数逐渐减小,本文给出的该系数为:

$$c(j|k) = N_{RHC}/l \quad (8)$$

式中, l 为航班 j 所在的时间间隔, $l=1, 2, \dots, N_{RHC}$, $t(k) + (l-1)T_{OI} < PLT(j|k) \leq t(k) + lT_{OI}$, 并且对于 $left_queue(k-1)$ 中的航班按 $l=1$ 计算。

动态环境下的进港航班排序,主要解决的问题是如何消除不确定因素的干扰,使得航班的总延误时间达到最小化. 上述模型采用 RHC 策略,将整个待排序过程划分成若干个时域进行,并以式(8)作为航班权重,体现出排序时主要考虑位置较近的航班,以式(7)作为时域的优化目标,从而能有效消除航班信息动态变化的干扰,确保以式(2)为总目标的实现。

2 基于 RHC 的多局部搜索 GA

针对模型的特点,本文设计了多局部搜索 GA (GA with multiple local searches, MLSGA), 算法前期,采用随机局部搜索扩大搜索空间;算法后期,为了避免随机搜索的任意性,进行定向局部搜索. 将 RHC 与 MLSGA 进行结合,作为本文问题模型的在线优化算法。

2.1 基本思路

根据当前时域上待排序航班集合 WS,生成初始化群体,基因代表航班,个体表示航班着陆次序. 按一定的策略选择个体进入交配池,并进行交叉、变异进化操作. 对于新生成的个体进行局部搜索,算法前期,随机选择一位置,交换该位置相邻的航班基因,在邻域中寻求更好的个体;随机局部搜索在一定程度上使得 GA 在更大空间里寻找最优解,然而这样往往会导致算法无法收敛,所以后阶段算法中的个体朝着既定的好的方向进行局部搜索,本文以群体中最优个体为基准,进行定向搜索,改善个体基因结构. 算法迭代若干代后,其最优解即为当前时域上满意的航班着陆次序,且对该次序中处于冻结窗口

的航班进行实际着陆,其余航班按优化次序存储到 $left_queue$ 队列中,以备后续时域的优化排序。

2.2 个体编码和适应度函数

根据 ASS 的特征,个体依据航班编号进行整数编码. 如当前时域上需要降落的航班有(2, 3, 5, 6, 7, 9, 10),图 2 所示个体代表该时域上的一种航班着陆顺序。

3	2	5	9	7	6	10
---	---	---	---	---	---	----

图 2 基于航班号的整数编码

Fig. 2 Coding with integer based on flight number

针对本文的极小化优化模型,以式(7)为时域上的目标函数,定义 x 的适应度函数:

$$F(x) = f_{\max} - f(x) + E \quad (9)$$

式中, $f(x)$ 和 f_{\max} 分别表示个体目标函数以及当前群体中个体最大目标函数值,常量 E 使得最大目标函数值个体在进化过程中有机会被选中,增加群体的多样性。

2.3 群体的初始化

为了促进时域间的信息交流,提高优化效果,本文给出一种群体初始化策略. 个体基因串实际上是当前时域上航班集合 WS 的一种排列, $left_queue$ 队列按优化次序存储了前一时域上的剩余航班,初始化群体由三部分组成:①构造一个航班排列, $left_queue$ 作为该排列的前缀, WS 中的其余航班按计划到达时间先后顺序排列,部分个体按此排列生成;②另一部分个体按 WS 中的航班计划到达时间的先后顺序或逆序生成;③其余个体随机生成. 如前所述,相邻时域上的公共航班经排序后,通常具有相同或相近的着陆顺序,引入策略①,可以在上一时域的排序基础上继续寻优,从而可以提高寻优效果. 图 3 所示为集合 WS 中的航班号 No 及计划到达时间 PLT,图 4 为剩余航班队列 $left_queue$. 图 5 中(a)~(c)所示分别为上述策略①—③对应的三种个体。

No.	2	3	5	6	7	8	9	10	11
PLT(s)	15	60	85	135	200	320	420	490	550

图 3 航班集合 WS

Fig. 3 Flight set WS

3	5	2	7	6
---	---	---	---	---

图 4 航班队列 $left_queue$

Fig. 4 Flight queue $left_queue$



图 5 三种个体基因串

Fig. 5 Three kinds of individuals

2.4 局部搜索策略

针对 GA 进化过程的特点,本文设计了随机局部搜索和定向局部搜索,个体进行 N_{li} 次局部搜索.

2.4.1 随机局部搜索

任意选取个体 x 基因串中的某一位置,并交换此位置处的相邻航班基因,生成新个体 y ,若 y 对应进港航班着陆次序优于 x ,则以新个体 y 取代 x .

2.4.2 定向局部搜索

为了避免随机搜索的盲目性,令 g_{best} 为种群到目前为止找到的最优个体,以其为基准,确定其余个体的局部搜索方向.本文基于离散粒子群算法思想^[25],设计了一种定向局部搜索策略,个体最大搜索速度依据其与基准个体 g_{best} 的差异而定,以改变现有恒速搜索难以取得满意解的现状,这里搜索速度是指个体基因结构的修改幅度.具体包括因素:①两者目标函数值的差值;②两者基因结构上的差异程度.本文设计的个体定向局部搜索如下:

Step 1 计算个体 x 的 $f(x) - f(g_{best})$,记为 $df(x)$.

Step 2 构造集合 $D_s, D_s = \{g_i | g_i \text{ 为个体 } x \text{ 基因串中的第 } i \text{ 个基因,且与 } g_{best} \text{ 基因串中该位置的基因不同}\}$,令 $dl(x) = |D_s|$.

Step 3 设置 x 局部搜索的最大速度阈值 $\max v(x); \max v(x) = dl(x) \exp^{-1/df(x)}$

Step 4 随机选取 x 的当前搜索速度 $Curv$,使得 $Curv = \text{rand}(1, \max v(x))$,其中 $\text{rand}(1, \max v(x))$ 表示 $[1, \max v(x)]$ 中的任意一个整数.在集合 D_s 中任选 $Curv$ 个元素构成子集 Sub_D_s .

Step 5 将 x 中所有非 D_s 中的基因按原位序复制到 y 中,将集合 Sub_D_s 中的基因按 g_{best} 中的位序复制到 y 中.

Step 6 构造集合 $D_s' = D_s - Sub_D_s$,将 D_s' 中所有基因按 x 中位序复制到 y 中,若 y 中该位置已有基因存在,则从左至右扫描空白位置插入,从而生

成新个体 y .

Step 7 若 y 优于 x ,则接受新个体 y .

2.5 交叉、变异算子

(I) 交叉算子

航班相对安排次序是 ASS 中的重要因素,合理的航班着陆次序是个体中的优良基因组织,在进化过程中尽量维持这样的基因组织.本文采用部分匹配交叉策略,将参与交叉的两个父个体的公共基因组织分别按对方的位序进行排列,从而生成两个子个体.设参与交叉的父个体为 P_1, P_2 ,选取交叉区间 $[\text{pos}_1, \text{pos}_2]$,令该区间上 P_1, P_2 公共基因构成集合 g_{set} ,将 P_1 中不属于 g_{set} 的基因按原序复制到 C_1 中, g_{set} 中的基因按在 P_2 中的顺序复制到 C_1 中;同理可得到子个体 C_2 .

(II) 变异算子

交叉算子中子个体主要是继承双亲的基因结构,难以扩展解的搜索空间,本文针对问题特点,设计了一种变异算子辅以寻找最优解.随机选取两个航班基因,交换它们的位置.

2.6 选择算子及相关参数

选择算子决定了哪些个体能够进入交配池,从而参与生成下一代的遗传操作.遵循“自然选择,适者生存”的生物进化原理,本文采用适应度比例选择策略复制个体,高适应度个体以较高概率进行复制.个体参与交叉、变异的概率采用固定形式.

2.7 基于 RHC 的多局部搜索 GA

以多局部搜索 GA 作为时域上航班的在线优化排序算法,从而基于 RHC 的多局部搜索 GA 具体步骤描述如下:

Step 1 假设已知 N_{AC} 架进港航班计划到达时间,设置时间间隔长度 T_{OI} 与时域上时间间隔总数 N_{RHC} ,初始化时域序号 $k=1$,并置 $\text{left_queue}(0)$ 为空.

Step 2 根据当前航班信息,由所有计划到达时间 PLT 位于 $[t(k), t(k) + N_{RHC} T_{OI}]$ 的航班以及 $\text{left_queue}(k-1)$ 构成当前时域上的待排序航班集合 $WS(k)$.

Step 3 使用多局部搜索 GA 对 $WS(k)$ 中的航班进行着陆优化排序.

Step 4 根据优化序列,给 $WS(k)$ 中的航班分配着陆时间,将处于冻结窗口中的航班进行着陆,其余航班存储到队列 $\text{left_queue}(k)$ 中,若 $k > 1$,则先清空队列 left_queue .

Step 5 若所有航班均已着陆,则结束算法.

Step 6 更新未着陆航班的相关信息, $k = k + 1$, 转 Step 2 继续执行.

其中 Step 3 中的多局部搜索 GA 描述如下:

Step 1 按 2.3 所述策略生成初始群体 Pop, 并计算种群中个体的适应值, 种群规模为 N_p ; 设置最大世代数 N_g 、交叉概率 P_c 、变异概率 P_m 以及两种局部搜索转换代数 C_{lg} 及局部搜索次数 N_{ls} ; 令代数 $g = 1$.

Step 2 使用 2.6 中的适应度比例选择策略从 Pop 中选取个体进入交配池, 进行 2.5 中的交叉、变异操作, 生成新个体, 并组成临时过渡种群 temp_Pop.

Step 3 对临时种群 temp_Pop 中的所有个体 x 执行 N_{ls} 次局部搜索, 步骤如下:

Step 3.1 令 $tl = 1$;

Step 3.2 若当前代数 $g \leq C_{lg}$, 则个体 x 进行 2.4.1 中的随机局部搜索; 否则进行 2.4.2 所述定向局部搜索;

Step 3.3 $tl = tl + 1$;

Step 3.4 若 $tl > N_{ls}$, 则停止个体 x 的局部搜索; 否则转 Step 3.2 继续执行.

Step 4 令 $Pop = temp_Pop$, 得到新一代种群.

Step 5 $g = g + 1$.

Step 6 如果 $g > N_g$, 结束算法, 否则转 Step 2 继续执行.

3 仿真与结果分析

为了验证模型与算法的有效性, 本文基于文献[12]的仿真方法, 分别对静态(static)和动态(dynamic)环境下的 ASS 问题进行实验仿真, 并使用 FCFS、多局部搜索遗传算法 MLSGA、基于 RHC 的基本遗传算法(RHC based basic GA, RHC_BGA)以及基于 RHC 的多局部搜索遗传算法 RHC_MLSGA 对算例进行计算. 这里, RHC_BGA 无局部搜索且初始种群随机产生; 另外, 为了验证多局部搜索的意义, 去除 RHC_MLSGA 中的局部搜索, 记此算法为 RHC_GA, 采用 RHC_GA 与上述算法同时对各算例进行计算, 并对比各算法的仿真结果. 本文以 VC6.0 为工具实现算法, 运行机器的处理器为双核(主频均为 2.93 GHz), 内存为 2 GB.

3.1 参数设置及仿真算例

(I) 参数设置

LTI(f_1, f_2)采用文献[2]中的最小安全尾流间隔, 如表 1 所示. 其中, TP 代表航班类型. 各种 GA 的种群规模 N_p 及最大世代数 N_g 均为待排航班数的 4 倍, 即 $N_p = N_g = 4 |WS|$, 而局部搜索次数 $N_{ls} = |WS|/2$, 局部搜索转换代数 $C_{lg} = N_g/2$; 交叉概率 $P_c = 0.9$, 变异概率 $P_m = 0.25$, 适应度函数常量 $E = 80$; 基于文献[11]的讨论, RHC 策略中的参数 $N_{RHC} = 4$, $T_{O1} = 300$ s.

表 1 连续着陆飞机最小间隔时间 LTI(s)

Tab. 1 Minimum landing time intervals(s)

	后着陆飞机				
	TP	1	2	3	4
首先	1	96	200	181	228
着陆	2	72	80	70	110
飞机	3	72	100	70	130
	4	72	80	70	90

(II) 仿真算例说明

首先依据文献[11]中某机场的航班信息, 选取 30 架进港航班作为排序对象, 组成单个仿真算例, 其中航班的计划到达时间 PLT 随机产生, 且无航班信息的变化情况. 航班信息如表 2 前三列所示, 包括航班号 No、航班类型 TP 以及 PLT. 为了较全面讨论算法的性能, 对文献[11-12]中的两组仿真算例采用本文所提算法进行计算.

为进一步验证模型及算法的有效性, 需减弱具体仿真算例数据对算法优化结果的影响, 分静态和动态两种环境进行 Monte Carlo 仿真实验, 算例中的航班数均为 50 架, 航班类型与 PLT 均随机产生. 每种环境又分为三种状态, 分别为非拥挤(uncongested)、正常(normal)和拥挤(congested)状态, 对应的航班计划到达时段 $[0, T]$ 分别为 $[0, 2\ 500\ s]$ 、 $[0, 5\ 000\ s]$ 和 $[0, 7\ 500\ s]$, 每种状态随机生成 20 份航班信息数据, 每份航班信息对应一个算例. 对于动态环境, 本文只考虑航班计划到达时间 PLT 的更改, 所以为动态环境中的每份航班信息附上一份随机生成的 PLT 变动数据, 且航班的 PLT 变化范围约束为 $[-300\ s, 300\ s]$, 每份信息中有 25% 的航班存在 PLT 变动. 在进行测试时, 每份航班信息数据, 除 FCFS 运行一次外, 其余算法均运行 50 次.

(III) 算法评价指标

实验中主要采用两种评价指标: ① 航班空中总延误时间, 这是 ASS 问题的核心目标, 对于同一算例的多次实验, 比较其平均值; ② 标准偏差

(standard deviation, Sdev)的平均值,标准偏差常作为 Monte Carlo 实验中衡量算法稳定性的评价标准,本文对各状态对应的多个算例仿真结果的标准偏差均值进行比较,其定义为:

$$M_{Sdev} = \frac{1}{N_{example}} \sum_{i=1}^{N_{example}} \sqrt{\frac{1}{N_{test,i}} \sum_{j=1}^{N_{test,i}} (TADT(j) - MDT(i))^2} \quad (10)$$

式中, $N_{example}$ 、 $N_{test,i}$ 分别表示仿真算例总数、仿真算例 i 的实验次数, $TADT(j)$ 、 $MDT(i)$ 分别代表算例 i 的第 j 个实验总延误时间以及该算例所有实验延误时间均值. 另外对算法的运行时间作简单分析.

3.2 仿真结果及分析

(I) 单个算例实验结果

表 2 所示为单个仿真算例及各算法运行结果,其中 TADT, RUNT 分别表示算法计算结果对应的总延误时间及运行时间,单位均为 s. 在总延误时间方面,相对于 FCFS 算法,MLSGA、RHC_BGA 及 RHC_GA 下降率分别为 44.07%、39.87% 和 42.62%,而 RHC_MLSGA 的降幅最大,为 53.07%. 显然,具有时域间优化排序等启发信息的 RHC_GA 下降程度大于 RHC_BGA,结合了 RHC 策略的 RHC_MLSGA 的优化结果好于 MLSGA,并且优于无局部搜索的 RHC_GA.

表 2 单个仿真算例及仿真结果

Tab. 2 Simulation results of a single example

Flight data			FCFS			MLSGA			RHC_BGA			RHC_GA			RHC_MLSGA		
No	TP	PLT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT
1	4	2 139	3	135	0	5	193	0	5	193	0	5	193	0	5	193	0
2	2	606	5	363	170	3	265	130	3	265	130	3	265	130	3	265	130
3	1	135	30	584	0	2	606	0	30	584	0	30	584	0	2	606	0
4	3	2 044	2	784	178	27	693	0	6	680	40	6	680	40	27	693	0
5	4	193	6	856	216	22	763	15	27	880	187	22	861	113	22	763	15
6	1	640	27	1 056	363	6	835	195	19	960	31	2	961	355	30	835	251
7	3	2 761	22	1 126	378	30	931	347	22	1 030	282	27	1 041	348	6	931	291
8	1	987	17	1 256	380	8	1 027	40	17	1 160	284	17	1 151	275	8	1 027	40
9	3	1 325	19	1 336	407	29	1 255	315	20	1 230	27	29	1 241	301	20	1 208	5
10	4	1 478	29	1 446	506	17	1 345	469	9	1 325	0	19	1 321	392	16	1 278	3
11	4	1 870	8	1 518	531	19	1 425	496	16	1 395	120	20	1 391	188	9	1 348	23
12	2	2 366	20	1 699	496	20	1 495	292	2	1 495	889	16	1 461	186	19	1 448	519
13	2	1 612	16	1 769	494	9	1 565	240	10	1 605	127	9	1 531	206	17	1 558	682
14	4	2 349	9	1 839	514	16	1 635	360	13	1 685	73	8	1 603	616	29	1 648	708
15	4	1 871	10	1 969	491	18	1 707	125	29	1 795	855	18	1 699	117	10	1 738	260
16	3	1 275	18	2 041	459	13	1 907	295	15	1 885	14	24	1 899	67	13	1 818	206
17	4	876	13	2 241	629	24	1 987	155	25	1 975	124	13	1 979	367	24	1 898	66
18	1	1 582	24	2 321	489	10	2 097	619	11	2 065	195	4	2 049	5	25	2 008	157
19	2	929	25	2 431	580	25	2 187	336	1	2 155	16	25	2 179	328	11	2 098	228
20	3	1 203	11	2 521	651	11	2 277	407	4	2 225	181	15	2 269	398	15	2 188	317
21	1	2 882	15	2 611	740	15	2 367	496	24	2 325	493	10	2 359	881	1	2 278	139
22	3	748	4	2 681	637	1	2 457	318	14	2 435	86	11	2 449	579	14	2 368	19
23	4	2 514	1	2 811	672	14	2 547	198	23	2 525	11	14	2 539	190	12	2 448	82
24	2	1 832	14	2 901	552	12	2 627	261	18	2 597	1 015	1	2 629	490	23	2 558	44
25	4	1 851	12	2 981	615	4	2 697	653	8	2 693	1 706	23	2 719	205	4	2 628	584
26	1	2 682	23	3 091	577	7	2 767	6	26	2 789	107	12	2 799	433	18	2 700	1 118
27	2	693	26	3 163	481	23	2 897	383	21	2 885	3	7	2 869	108	26	2 796	114
28	4	2 946	7	3 344	583	28	2 987	41	7	3 066	305	28	2 999	53	21	2 892	10
29	4	940	21	3 416	534	21	3 059	177	12	3 166	800	21	3 071	189	7	3 073	312
30	1	584	28	3 644	698	26	3 155	473	28	3 276	330	26	3 167	485	28	3 203	257
TADT			14 021			7 842(RUNT,0.312)			8 431			8 045			6 580(RUNT,0.296)		

算法运行时间方面,因为 RHC 策略将整个航班排序过程划分成若干个阶段进行,降低了问题的规模,所以 RHC_MLSGA 的运行时间要少于 MLSGA. RHC 框架下的算法运行耗时降低情况在文献[11-12,14]中已作了详细阐述,限于篇幅,本文后续内容不再讨论.为了与现有代表性的算法进行比较,本文以文献[11-12]中的两组初始数据为算例,采用 RHC_MLSGA 进行计算,并与文献中相关算法的运行结果进行比较,运行结果如表 3、表 4 所示.

表 3 和表 4 所示算例是用于测试 HU 提出的基于 RHC 的航班排序算法(这里记为 RHC_HU)以

及 BRGA2 的性能,文献[11-12]给出了其他算法的运算结果,并作比较,如 DOA、DTSPM 与 PRGA. 表 3 所示结果表明,本文所提算法 RHC_MLSGA 对应的航班延误时间要少于 DOA,与 RHC_HU 对应的延误时间相同;表 4 中 RHC_MLSGA 对应的延误时间要好于 DTSPM 与 PRGA,亦优于 BRGA2. 表 2~4 三组算例计算结果显示了 RHC_MLSGA 具有良好的航班排序能力,为了充分验证算法的性能,下面对众多算例的 Monte Carlo 实验结果进行统计与分析.

(II) Monte Carlo 实验结果

表 5、表 6 分别统计了静态和动态环境下的

表 3 文献[11]仿真算例及仿真结果

Tab. 3 Simulation results of the example from Ref. [11]

Flight data			DOA			RHC_HU			RHC_MLSGA		
No	TP	PLT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT
1	4	17	1	17	0	1	17	0	1	17	0
2	2	50	2	97	47	2	97	47	2	97	47
3	1	117	4	167	38	4	167	38	4	167	38
4	3	129	7	237	20	7	237	20	7	237	20
5	4	143	9	307	19	9	307	19	9	307	19
6	1	174	3	379	262	6	379	205	6	379	205
7	3	217	12	579	74	8	475	215	3	475	358
8	1	260	13	659	133	3	571	454	8	571	311
9	3	288	5	769	626	16	752	162	16	752	162
10	4	340	16	839	249	20	822	131	22	822	60
11	4	350	20	909	218	22	892	130	20	892	201
12	2	505	22	979	217	12	992	487	12	992	487
13	2	526	8	1 051	791	13	1 072	546	19	1 072	403
14	4	541	18	1 147	496	19	1 152	483	24	1 152	360
15	4	554	21	1 243	536	24	1 232	440	27	1 232	168
16	3	590	26	1 339	346	27	1 312	248	13	1 312	786
17	4	648	6	1 435	1 261	10	1 422	1 082	5	1 422	1 279
18	1	651	30	1 531	130	11	1 512	1 162	29	1 512	305
19	2	669	19	1 731	1 062	14	1 602	1 061	17	1 602	954
20	3	691	24	1 811	1 019	15	1 692	1 138	14	1 692	1 151
21	1	707	27	1 891	827	17	1 782	1 134	11	1 782	1 432
22	3	762	11	2 001	1 651	23	1 872	1 103	15	1 872	1 318
23	4	769	14	2 091	1 550	25	1 962	1 103	28	1 962	885
24	2	792	15	2 181	1 627	28	2 052	975	23	2 052	1 283
25	4	859	17	2 271	1 623	29	2 142	935	10	2 142	1 802
26	1	993	23	2 361	1 592	5	2 232	2 089	25	2 232	1 373
27	2	1 064	25	2 451	1 592	18	2 304	1 653	18	2 304	1 653
28	4	1 077	28	2 541	1 464	21	2 400	1 693	21	2 400	1 693
29	4	1 207	29	2 631	1 424	30	2 496	1 095	26	2 496	1 503
30	1	1 401	10	2 721	2 381	26	2 592	1 599	30	2 592	1 191
TADT			23 275			21 447			21 447		

表 4 文献[12]仿真算例及仿真结果

Tab. 4 Simulation results of the example from Ref. [12]

Flight data			DTSPM			PRGA			BRGA2			RHC_MLSGA		
No	TP	PLT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT	No	ALT	DAT
1	1	1 935	9	35	0	9	35	0	9	35	0	9	35	0
2	3	400	5	142	0	5	142	0	5	142	0	5	142	0
3	4	879	10	307	0	10	307	0	10	307	0	10	307	0
4	1	328	4	403	75	4	403	75	4	403	75	4	403	75
5	3	142	19	499	5	19	499	5	19	499	5	19	499	5
6	2	1 980	17	595	30	17	595	30	17	595	30	17	595	30
7	2	915	2	776	376	2	776	376	12	795	433	2	776	376
8	2	1 814	18	846	181	18	846	181	18	865	200	18	846	181
9	4	35	12	946	584	12	946	584	2	935	535	12	946	584
10	1	307	3	1 056	177	3	1 056	177	7	1 035	120	7	1 026	111
11	3	1 414	15	1 146	193	15	1 146	193	3	1 145	266	3	1 136	257
12	2	362	7	1 226	311	7	1 226	311	15	1 235	282	15	1 226	273
13	4	1 279	14	1 298	318	13	1 336	57	13	1 325	46	13	1 316	37
14	1	980	13	1 526	247	20	1 416	8	20	1 408	0	20	1 408	0
15	4	953	20	1 606	198	14	1 488	508	11	1 478	64	11	1 478	64
16	3	1 726	11	1 676	262	11	1 669	255	14	1 550	570	14	1 550	570
17	1	565	16	1 746	20	16	1 739	13	16	1 731	5	16	1 731	5
18	3	665	8	1 846	32	8	1 839	25	8	1 831	17	8	1 831	17
19	1	494	6	1 980	0	6	1 980	0	6	1 980	0	6	1 980	0
20	2	1 408	1	2 052	117	1	2 052	117	1	2 052	117	1	2 052	117
TADT			3 126			2 915			2 765			2 702		

表 5 静态环境下的仿真结果

Tab. 5 Simulation results in static environments

算法	uncongested		normal		congested	
	MTADT	Dec_rate	MTADT	Dec_rate	MTADT	Dec_rate
FCFS	7 488.90		25 213.35		74 233.05	
MLSGA	4 736.41	36.75%	12 636.54	49.88%	48 216.85	35.05%
RHC_BGA	5 881.13	21.47%	16 391.88	34.99%	55 409.09	25.36%
RHC_GA	4 786.60	36.08%	13 143.49	47.87%	51 815.65	30.20%
RHC_MLSGA	4 676.15	37.56%	12 221.54	51.53%	48 169.71	35.11%

表 6 动态环境下的仿真结果

Tab. 6 Simulation results in dynamic environments

算法	uncongested		normal		congested	
	MTADT	Dec_rate	MTADT	Dec_rate	MTADT	Dec_rate
FCFS	7 971.90		24 538.70		73 250.50	
MLSGA	6 645.13	16.64%	15 933.05	35.07%	50 702.78	30.78%
RHC_BGA	7 484.78	6.11%	18 983.34	22.64%	56 682.53	22.62%
RHC_GA	6 549.17	17.85%	15 906.37	35.18%	52 848.58	27.85%
RHC_MLSGA	6 563.01	17.67%	15 818.46	35.54%	50 588.52	30.94%

Monte Carlo 实验结果,其中 MTADT 表示每种状态下所有仿真实验对应航班总延误时间的平均值,Dec_rate 表示算法相对于 FCFS 的 MTADT 下

降率.

由表 5、表 6 统计结果知:①各算法均取得较 FCFS 好的优化效果,其中 RHC_GA 的总延误时间

均值少于 RHC_BGA 对应的结果,这是因为 RHC_GA 对初始种群的产生引入了启发信息,其中前一时域上优化结果信息的传递使得当前时域上的排序在一定的已有排序基础上进行,其效果等价于延长了 RHC_GA 进化代数. 为了避免完全受优化启发信息的约束,RHC_GA 初始种群的部分个体按计划到达时间顺序或逆序产生,其余则随机生成,而 RHC_BGA 初始种群则全部随机生成. ②RHC_MLSGA 相对于 FCFS,在各环境对应状态下总延误时间均值的下降率分别为 37.56%、51.53%、35.11%、17.67%、35.54% 和 30.94%,均高于对应的 MLSGA 下降率. 由于结合了 RHC 策略,将整个时间段内的航班排序划分成若干个阶段进行,从而降低了问题优化复杂性,提高了 RHC_MLSGA 的优化效果. ③RHC_MLSGA 的 6 种总延误时间的均值中仅在动态环境的 uncongested 状态下稍劣于 RHC_GA,其余 5 种情况均好于后者,并且航班密度越大,两者的优化效果差距越明显. 由于 RHC_MLSGA 引入了局部搜索机制,使个体能够在其邻域中挖掘更好的个体,因而弥补了 GA 弱局部搜索的特性. 此外,多局部搜索机制使得算法在前期能够在较大空间中寻找优秀个体;后期的定向局部搜索能够牵引个体向好的方向搜索,并且结合个体的具体情况调整其局部搜索速度,从而能够提高群体整体解的品质. 结合表 2 结果,RHC_MLSGA 对于问题的求解性能优于 RHC_GA. ④RHC_MLSGA 的延误时间下降率都高于基本遗传算法 RHC_BGA. 为进一步分析算法的性能,表 7 给出了各状态下算法计算结果的标准偏差平均值 MSdev.

表 7 中算法各状态下的标准偏差均值按降序排列依次为 RHC_BGA、RHC_GA、MLSGA 和 RHC_MLSGA. 显然,具有启发信息的 RHC_GA 的稳定性好于 RHC_BGA,但是劣于引入局部搜索的 RHC_MLSGA 和 MLSGA,充分体现了多局部搜索机制能有效提高算法的稳定性. 另外,RHC_MLSGA 较

没有结合 RHC 策略的 MLSGA 稳定.

综上所述,本文提出的基于 RHC 策略的 ASS 问题模型及其求解算法 RHC_MLSGA 能够有效降低航班总延误时间.

3.3 ASS 问题的特性总结及分析

ASS 问题的性质与环境变化以及航班密度等因素相关,这里总结几点特性并作分析.

(I) 相对于 FCFS,正常状态 normal 下各算法的优化效果通常达到最佳. 这是因为在非拥挤状态 uncongested 下,航班密度稀疏,于是航班无需太多的位移即可降低总延误时间,所以各算法的优化性能在此状态下不能较好地体现;另一方面,在拥挤状态 congested 下,由于大量航班在短时间内到达而形成的高压,为了实现总延误时间最小化的目标,需要进行航班位移次数激增,而为了提高算法的计算效率,所使用算法只能进行一定量的位移操作,从而约束了该状态下各算法的问题优化性能.

(II) 静态环境 static 下各算法的总延误时间均值下降率要高于对应的动态环境 dynamic. 这是因为在静态环境中,航班的信息无变化,从而算法可以依照原始信息进行问题优化,而动态情境中航班信息的随时更新,增加了问题的复杂性,从而降低了算法的优化性能,同时亦影响了算法求解问题的稳定性(见表 7 中各算法的标准偏差均值).

(III) 相对于静态环境,动态环境下各算法优化性能的降低程度随着航班拥挤程度的增加而逐渐变小. 如 RHC_MLSGA,表 5 和表 6 中静态和动态环境对应三种状态下的延误时间均值下降率分别为 (uncongested: 37.56%, 17.67%)、(normal: 51.53%, 35.54%) 和 (congested: 35.11%, 30.94%),同一状态下降率比值分别为 2.13、1.45 和 1.13. 这是因为随着航班密度的减小,对到达时间的更改会变得敏感起来,因为在密度稀疏的情况下,大多数航班能够按照其计划到达时间着陆,或进行较小的空中延误,若此时更改这些航班的到达时

表 7 仿真结果标准偏差平均值

Tab. 7 Mean value of standard deviation of simulation results

算法	Static			Dynamic		
	uncongested	normal	congested	uncongested	normal	congested
MLSGA	81.842 5	509.880 5	1 289.630 0	227.834 5	1 091.291 0	2 547.856 0
RHC_BGA	620.498 5	1 506.183 0	2 462.660 0	1 006.157 0	2 411.440 0	3 880.499 0
RHC_GA	108.306 0	554.255 0	1 374.040 0	314.147 0	1 394.671 0	2 170.654 0
RHC_MLSGA	25.129 0	239.088 0	888.196 5	121.199 5	679.567 0	2 103.300 0

间,会导致更多的延误时间;相反密度较高航班队列受 PLT 变化影响较小.

4 结论

针对现有基于 RHC 策略的 ASS 问题模型不足,本文提出了一种时域间优化排序信息传递方法,并将其融入问题模型的求解算法 RHC_MLSGA. 鉴于 GA 较易陷入局部最优的缺陷,在 RHC_MLSGA 中引入了多局部搜索机制,通过随机和定向两种局部搜索来提高算法的问题求解性能. 大量的仿真实验结果显示了本文所提模型及算法的有效性和算法的稳定性. 本文针对单跑道机场中航班进港排序问题进行了建模和算法设计,任何机场都存在航班的进港和离港过程,并且随着机场跑道的增建,如何在多跑道机场中协调进离港航班之间的关系,进一步提高机场吞吐量,是我们的下一步工作.

参考文献(References)

- [1] Pélegrin M. Towards Global Optimization for Air Traffic Management[R]. AGARDAGraph-321, 1994.
- [2] Bianco L, Dell'Olmo P, Giordani S. Scheduling models and algorithms for TMA traffic management[A]// Modelling and Simulation in Air Traffic Management. Berlin, Germany: Springer, 1997: 139-167.
- [3] Beasley J E, Krishnamoorthy M, Sharaiha Y M, et al. Scheduling aircraft landings - the static case [J]. Transportation Science, 2000, 34(2): 180-197.
- [4] Hansen J V. Genetic search methods in air traffic control[J]. Computers & Operations Research, 2004, 31(3): 445-459.
- [5] Guo Y P, Cao X B, Zhang J. Multiobjective evolutionary algorithm with constraint handling for aircraft landing scheduling [C]// Proceedings of the IEEE Congress on Evolutionary Computation. Hong Kong, China: IEEE Press, 2008: 3 657-3 662.
- [6] Zhang Honghai, Hu Minghua. Modeling and simulation of capacity utilization and flow assignment in multi-airport terminal area [J]. Journal of System Simulation, 2009, 21(18): 5 639-5 644.
张洪海, 胡明华. 多机场终端区容流利用和分配建模与仿真[J]. 系统仿真学报, 2009, 21(18): 5 639-5 644.
- [7] Ying Shenggang, Sun Fuchun, Hu Laihong, et al. Multi-objective dynamic programming algorithm for aircraft arrival sequencing and runway scheduling[J]. Control Theory & Applications, 2010, 27(7): 827-835.
- [8] CaprÀ S, Ignaccolo M. Genetic algorithms for solving the aircraft-sequencing problem: The introduction of departures into the dynamic model[J]. Journal of Air Transport Management, 2004, 10(5): 345-351.
- [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] Malaek S M B, Naderi E. A new scheduling strategy for aircraft landings under dynamic position shifting [C]// The 2008 Aerospace Conference. Montana, USA: IEEE Press, 2008: 1-8.
- [11] Hu X B, Chen W H. Receding horizon control for aircraft arrival sequencing and scheduling [J]. IEEE Transactions on Intelligent Transportation Systems, 2005, 6(2): 189-197.
- [12] Hu X B, Paolo E D. Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling [J]. IEEE Transactions on Intelligent Transportation Systems, 2008, 9(2): 301-310.
- [13] Jia X L, Cao X B, Guo Y P, et al. Scheduling aircraft landing based on clonal selection algorithm and receding horizon control [C]//The 11th International IEEE Conference on Intelligent Transportation Systems. Beijing, China: IEEE Press, 2008: 357-362.
- [14] Zhan Z H, Zhang J, Li Y, et al. An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem[J]. IEEE Transactions on Intelligent Transportation Systems, 2010, 11(2): 399-412.
- [15] Zhang Qiqian, Hu Minghua, Shi Saifeng. Multi-object optimization algorithm for aircraft landing based on receding horizon control strategy [J]. Journal of Nanjing University of Aeronautics & Astronautics, 2012, 44(3): 393-398.
张启钱, 胡明华, 施赛锋. 基于 RHC 的航班着陆调度多目标优化算法[J]. 南京航空航天大学学报, 2012, 44(3): 393-398.
- [16] Beasley D, Martin R R, Bull D R. An overview of genetic algorithms: Part 1. Fundamentals [J]. University Computing, 1993, 15(2): 58-69.
- [17] Srinivas M, Patnaik L M. Adaptive probabilities of crossover and mutation in genetic algorithms[J]. IEEE Transactions on Systems, Man and Cybernetics, 1994, 24(4): 656-667.
- [18] Jun J H, Lee D W, Sim K B. Realization of cooperative strategies and swarm behavior in

- distributed autonomous robotic systems using artificial immune system[C]//Proceedings of the International Conference on Systems, Man and Cybernetics. Tokyo, Japan: IEEE Press, 1999, 6: 614-619.
- [19] Gao F, Shen Y P, Li L X. The optimal design of piezoelectric actuators for plate vibroacoustic control using genetic algorithms with immune diversity[J]. *Smart Materials and Structures*, 2000, 9(4): 485-491.
- [20] Orvosh D, Davis L. Using a genetic algorithm to optimize problems with feasibility constraints[C]//The 1st IEEE Conference on Evolutionary Computation. Orlando, USA: IEEE Press, 1994: 548-553.
- [21] Munawar A, Wahib M, Munetomo M, et al. Hybrid of genetic algorithm and local search to solve max-sat problem using nvidia cuda framework [J]. *Genetic Programming and Evolvable Machines*, 2009, 10(4): 391-415.
- [22] Li S Z, Chen Y H, Du H F, et al. A genetic algorithm with local search strategy for improved detection of community structure[J]. *Complexity*, 2010, 15(4): 53-60.
- [23] Duan Xianzhong, Li Zhihuan, Li Yin hong. Multimeme memetic algorithm based on multiple local search strategies for reactive power optimization [J]. *Proceedings of the CSEE*, 2008, 28(34): 59-65.
段献忠, 李智欢, 李银红. 采用多局部搜索策略的无功优化多模因算法[J]. *中国电机工程学报*, 2006, 28(34): 59-65.
- [24] Primbs J A, Sung C H. Stochastic receding horizon control of constrained linear systems with state and control multiplicative noise[J]. *IEEE Transactions on Automatic Control*, 2009, 54(2): 221-230.
- [25] Zhong Yiwen, Cai Rongying. Discrete particle swarm optimization algorithm for QAP[J]. *Acta Automation Sinica*, 2007, 33(8): 871-874.
钟一文, 蔡荣英. 求解二次分配问题的离散粒子群优化算法[J]. *自动化学报*, 2007, 33(8): 871-874.

(上接第 74 页)

参考文献(References)

- [1] 向伟昌. 流媒体服务器性能测评系统的设计与实现[D]. 国防科学技术大学, 2004.
- [2] 程超. 基于排队论的视频点播系统性能分析[D]. 华中科技大学, 2009.
- [3] Xiang Weichang, Peng Yuxing. Design and implementation of a performance evaluation system for stream media servers[J]. *Computer Engineering & Science*, 2006, 28(5): 45-47.
向伟昌, 彭宇行. SSMU 流媒体服务器性能测评系统的设计与实现[J]. *计算机工程与科学*, 2006, 28(5): 45-47.
- [4] Coffman Jr E. G., Muntz R. R., Trotter H. Waiting time distributions for processor-sharing systems [J]. *Journal of the ACM (JACM)*, 1970, 17(1): 123-130.
- [5] Kleinrock L. Time-shared systems: A theoretical treatment [J]. *Journal of the ACM (JACM)*, 1967, 14(2): 242-261.
- [6] Morrison J. A., Response-time distribution for a processor-sharing system [J]. *SIAM Journal on Applied Mathematics*, 1985, 45(1): 152-167.
- [7] Guillemin F, Boyer J. Analysis of the M/M/1 queue with processor sharing via spectral theory [J]. *Queueing Systems*, 2001, 39(4): 377-397.
- [8] Akar N. Moments of Conditional Sojourn Times in Finite Capacity M/M/1/N-PS Processor Sharing Queues [J]. *IEEE Communications Letters*, 2012, 16(4): 533-535.
- [9] Lee W Y, Wang C L. Conditional sojourn times of processor-sharing queues [J]. *Probability in the Engineering and Informational Sciences*, 2013, 27(01): 99-114.
- [10] Knessl C. On finite capacity processor-shared queues [J]. *SIAM Journal on Applied Mathematics*, 1990, 50(1): 264-287.
- [11] Knessl C. On the sojourn time distribution in a finite capacity processor shared queue [J]. *Journal of the ACM (JACM)*, 1993, 40(5): 1 238-1 301.