计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2017,26(12):110−115 [doi: 10.15888/j.cnki.csa.006070]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

双交叉算子遗传算法在终端区飞机排序中的应用①

赵 俨[1], 乐 俊[2], 刘 丹[1]

[1](电子科技大学 电子科学技术研究院, 成都 611731)
[2](西南电子电信技术研究所, 成都 610041)

摘 要: 当今, 普遍的航班延误现象不仅增加了巨额飞行成本, 还影响乘客体验. 对终端区待降飞机队列进行合理调整, 可以提高跑道利用率, 减少航班延误, 达到降低延误代价的效果. 针对终端区飞机排序问题, 提出一种包含双交叉算子的遗传算法, 针对不同适应度染色体采取不同的交叉操作, 使得在交叉过程中既能保护优质染色体, 也能使其它染色体继续进化. 同时引入重排算子对变异后的子代进行优化, 共同加快遗传算法收敛速度, 使其更加符合实际使用需求. 实验结果表明, 算法收敛速度得到改进, 能在可接受时间内得到可行解.

关键词: 空中交通管制; 终端区; 飞机排序; 遗传算法; 交叉算子

引用格式: 赵俨,乐俊,刘丹.双交叉算子遗传算法在终端区飞机排序中的应用.计算机系统应用,2017,26(12):110–115. http://www.c-s-a.org.cn/1003-3254/6070.html

Application of Double Crossover Operator Genetic Algorithm to Aircraft Sequencing in Terminal Area

ZHAO Yan[1], YUE Jun[2], LIU Dan[1]

[1](Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China)
[2](Research Institute of Southwest Electronic Telecommunication Technology, Chengdu 610041, China)

Abstract: Nowadays, the widespread phenomenon of flight delays does not only increase massive flight costs, but also impacts the experience of passengers. Reasonably adjusting the aircraft queue in terminal areas can raise the utilization ratio of the runway and reduce flight delays. Finally, the cost of delay can be cut down. To resolve the problem of aircraft scheduling in terminal areas, this article puts forward an genetic algorithm including double crossing operator. Different crossover operations are carried out for chromosomes with different fitness so that the quality of chromosomes can be protected and the others continue to evolve. At the same time, reordering operator is introduced to optimize the descendants after mutation to improve convergence rate of genetic algorithm and makes it more suitable for practical use. The experimental results show that the convergence rate of algorithm is improved and the feasible solution can be obtained within acceptable time.

Key words: air traffic control; terminal area; aircraft sequencing; genetic algorithm; crossover operator

随着国际民航运输的高速发展, 我国对民航运输的需求日益提高, 飞行器的数量也随之增加. 伴随空中交通流量的快速增长, 机场、空域和航线普遍出现了拥堵现象. 2016 年我国民航平均航班正常率为 76.76%, 较前两年虽有回升, 但与美、日等国比较, 仍有较大差距. 终端区拥堵不仅造成航班延误, 还会存在安全隐患, 引发安全事故. 通过对终端区飞机队列的合理排序, 可有效的提高跑道容量, 缓和拥堵, 进而减少飞机延误[1].

① 收稿时间: 2017-03-06; 修改时间: 2017-03-23; 采用时间: 2017-03-27

目前, 国内常用的飞机排序方式是先到先服务算法, 即仅根据飞机的预计到场时间进行排序. 该算法的优势在于实现简便、易操作, 但会产生较大的延误代价. 此外, 较常见的排序算法还有滑动窗优化算法[2]、约束位置交换算法[3]、时间提前量算法[4]和模糊模式识别算法[5]等确定性优化算法. 但随着飞机数量的增加, 确定性优化算法难以在可接受的时间内得出高复杂度排序问题的结果. 终端区飞机排序问题已被归为 NP 难问题, 使用常规算法求最优解已不切实际.

1975 年由 Michigan 大学的 John Holand 教授与其同事提出的遗传算法 (Genetic algorithm, GA)[6]具有良好的全局搜索能力、潜在并行性、可扩展性等优点, 适合用于解决此类问题. 许多科研人员通过对传统遗传算法进行改进并融入其他算法或约束条件, 提出了许多针对飞机排序的算法. 常会贤、曲世茹利用矩阵编码为飞机队列在多跑道降落的问题提出解决方案[7]; 白重阳等人利用遗传算法对飞机的速度进行调整, 使得飞机在终端区航路冲突极小, 从而减少飞机延误[8]. 陈亮等人对飞机延误代价系数的计算进行了讨论, 指出代价系数会根据飞机类型和等待时间等因素的影响而动态改变. 根据该系数得出的飞机延误总代价模型更具有实际意义, 而不再是片面的考虑如何将总延误时间降至最低[9].

在飞机排序的实际应用场景中, 要求在可接受的时间范围内得到可以减少飞机延误的飞机排序队列. 这不仅对排序质量有要求, 排序的效率也不可忽视. 本文主要受文献[10]中提出的“交叉掩码”的概念和使用方式的启发, 采取针对不同适应度的染色体采用双交叉算子分别处理的方式, 达到既能保证优质染色体平稳进化, 又不会影响其他染色体的进化速度的目的, 使种群进化方向更加明确, 从而加快算法收敛速度.

1 建立算法模型

1.1 终端区环境

由于机场附近的航线非常复杂, 飞机在此空域活动频繁, 因此需要设立终端管制区进行合理管制. 飞机的起飞和降落都将在该区域接受管制员的调度, 安全、有序的完成离场和进场. 终端区结构如图 1 所示.

终端区的定义和作用使其成为了极其复杂的区域, 也是造成空中交通拥堵的关键因素. 仅仅依靠新建跑道或扩大终端区等改善硬件条件的方法已经不足以解决终端区空中交通压力过载的问题. 采取更有效的利用已有的资源、选择合理的飞机降落次序、提高跑道容量等策略, 可以大大缓解终端区和机场拥堵带来的交通压力.

本文将针对飞机进场排序问题, 并基于已有的算法, 提出一些改进和创新, 力求得到一个更加高效的问题解决方案.

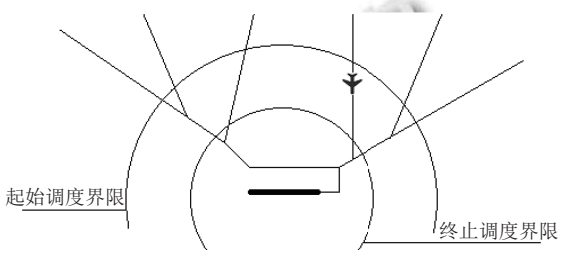

图 1 终端区结构示意图

1.2 算法模型

本文以多班次待降航班在单跑道降落的场景建模. 假设终端区待降队列中有 n 架飞机, 用集合 A $=\{A_1,A_2,\ldots A_n\}$来表示. E_i 表示飞机 i 的预计到达时间, S_i 表示飞机的实际到达时间. 飞机的类型有: 重型机 (H)、中型机 (L)、轻型机 (S) 三类. 相邻的第 i-1 架与第 i 架飞机的最小时间间隔用$I_{i-1,i}$表示.

由国际民航组织规定的不同机型之间的最小尾流时间间隔标准 (单位: 秒) 如表 1 所示.

表 1 尾流间隔标准

尾流间隔		后机机型		
		H	L	S
前机机型	H	94	114	167
	L	74	74	138
	S	74	74	98

模型约束条件:

$$S_i \geq E_i, \quad i \in \{1,2,3\ldots,n\}$$
$$S_i - S_{i-1} \geq I_{i-1,i}, \quad i \in \{2,3\ldots,n\}$$

前者表示每架飞机实际降落时间不得早于预计降落时间; 后者表示相邻飞机实际降落时间间隔不得小于尾流时间间隔要求. 根据上述约束条件, 第 i 架飞机的实际降落时间应为:

$$S_i = \max\{E_i, S_{i-1} + I_{i-1,i}\} \tag{1}$$

第 i 架飞机的延误时长为:

$$d_i = S_i - E_i \tag{2}$$

对于单架飞机来说, 延误代价主要是由飞机类型、等待时长和航班优先级这些因素决定. 参考文献[8,9]并加以改进得代价系数公式:

$$K_i = \begin{cases} 0, & d_i = 0 \\ P_i + \alpha * \beta^{\frac{d_i}{\gamma}}, & d_i > 0 \end{cases} \tag{3}$$

公式 (3) 参数说明: P_i 表示飞机 i 的优先级, 用于调整特定航班降落优先级或应对紧急处理. α, β, γ 均为正整数, 其中, α 为与机型相关的线性参数, β 为与机型相关的指数底数, γ 为时间延误基数. 代价系数的含义: 即飞机在延误时长 d_i 达到 γ 后, 该飞机的延误代价系数会根据不同机型决定的 α、β 以及 d_i 的大小呈指数增长, 若 d_i=0, 则 K_i=0, 延误代价也等于 0.

根据上述条件, 每架飞机的延误代价可如下计算:

$$D_i = K_i * d_i, \quad (n \geq 1) \tag{4}$$

为避免出现延误代价集中在个别飞机上, 采用各架飞机延误代价平方和建立模型:

$$D_{target} = \sum_1^n D_i^2 = \sum_1^n (K_i * d_i)^2, \quad (n \geq 1) \tag{5}$$

那么, 问题转化为求得飞机使 D_{target} 的值最小的飞机队列. 个体适应度函数:

$$f = \frac{1}{D_{target} + 1} \tag{6}$$

2 改进的遗传算法

2.1 改进遗传算法概述

为加快遗传算法的收敛速度, 文献[10]中提出了交叉掩码的概念和使用, 以此来保证交叉、变异得到的后代染色体会继承父代的优质基因, 提高个体适应度, 从而加快收敛. 然而, 交叉掩码的使用可以保护高适应度染色体, 但也会阻碍低适应度染色体的进化, 延缓整个种群的进化速度. 尤其是在搜索初期, 如果较优质染色体仅占很小的比例, 那么搜索周期可能会被大幅延长.

为了既能利用交叉掩码带来的优势, 又不阻碍低适应度染色体的进化, 可以尝试针对不同适应度的染色体采用不同的交叉算子和变异算子做区别处理. 对高适应度的染色体, 通过使用交叉掩码提供保护; 对低适应度染色体, 采用部分映射杂交算子 (Partially mapped crossover, PMX) 完成交叉操作. 通过这种双交叉算子的遗传算法来提高种群收敛速度, 从而更快的得出可行解.

2.2 编码

为方便后文中将介绍的重排算子的使用, 飞机 i 的编码采用“飞机型号”+“该飞机在同类型飞机中预计到达跑道先后的序号”, 不同类型飞机编码互不影响. 编码结果举例: S1、L1、S2、H1、S3、S4、H2、S5、L2、S6. 该编码方式可以保证在交叉、变异或重排后, 每架飞机编号依然唯一, 符合实际意义.

2.3 自适应算法

自适应遗传算法是针对每个染色体的适应度来计算适合该染色体的 P_c 和 P_m. 对高适应度的染色体降低 P_c 和 P_m, 使其得到保护; 反之, 对适应度较低的染色体提高 P_c 和 P_m, 使其在下一代中被淘汰. 目的是避免发散和陷入局部最小[11]. 自适应遗传算法的作用与本文的目的一致. 因此, 算法将采用以下公式计算 P_c 和 P_m[12]:

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{\max} - f_{avg}}, & (f' \geq f_{avg}) \\ P_{c1}, & (f' < f_{avg}) \end{cases} \tag{7}$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f - f_{avg})}{f_{\max} - f_{avg}}, & (f \geq f_{avg}) \\ P_{m1}, & (f < f_{avg}) \end{cases} \tag{8}$$

式中, f 为适应度值, f' 取双亲染色体的适应度中的较大值, $f_{\max}$ 表示当代种群中的适应度最大值, f_{avg} 表示当代适应度平均值. 一般 P_{c1} 和 P_{c2} 分别取 0.9 和 0.6, P_{m1} 和 P_{m2} 分别取 0.1 和 0.001.

2.4 双交叉算子

2.4.1 交叉掩码

交叉掩码的计算[10]: 交叉掩码是根据近两代的适应度最高的个体的相似基因来确定的. 取近两代适应度最高个体的染色体进行比较, 对应基因座的基因相同, 则交叉掩码对应位置记为 1, 否则记为 0, 如此就产生了交叉掩码, 并可以利用它将优质基因传递下去.

具体的, 在本算法中计算示例如下.

已知 i-2、i-1 代种群中适应度最高的染色体 C_{i-2}、C_{i-1}, 则第 i 代交叉掩码的计算如下:

C_{i-2}:	S1	L1	S3	H1	S5
C_{i-1}:	S1	H1	S3	L1	S5
M_i:	1	0	1	0	1

在染色体使用交叉掩码进行交叉操作时, 交叉掩码基因为 1 的基因座直接沿用父代基因, 剩余基因按照另一条父代染色体中相应基因的顺序进行排序. 在本算法中示例如下:

$$\begin{array}{lccccc} M_i: & 1 & 0 & 1 & 0 & 1 \\ P_1: & S1 & L1 & S3 & H1 & S5 \\ P_2: & H1 & S1 & S5 & L1 & S3 \end{array}$$

那么得到的孩子染色体如下 (假设两条染色体适应度均符合用交叉掩码进行交叉操作):

$$\begin{array}{l} C_1: \ S1\ H1\ S3\ L1\ S5 \\ C_2: \ H1\ S1\ S5\ L1\ S3 \end{array}$$

使用交叉掩码进行交叉操作的优势在于父代的优质基因完全得以保留, 使高适应度个体平稳进化, 避免发散.

2.4.2 部分映射杂交算子[13]

第一步. 在范围[1, n)(n 为飞机数量) 随机产生两个整数 r_1、r_2, 将待交叉染色体均分为 3 段, 并交换中间部分. 例如: r_1=2, r_2=5.

交叉双亲初始状态:

$$\begin{array}{l} P_1: \ L1\ H1|S3\ S1\ S5|H2 \\ P_2: \ H1\ S1|H2\ S5\ L1|S3 \end{array}$$

交换中间部分得:

$$\begin{array}{l} C_1: \ L1\ H1|H2\ S5\ L1|H2 \\ C_2: \ H1\ S1|S3\ S1\ S5|S3 \end{array}$$

第二步. 染色体两端部分有与中间部分重复的基因用*代替, 表示冲突基因. 即:

$$\begin{array}{l} C_1: \ *\ H1|H2\ S5\ L1|* \\ C_2: \ H1\ *|S3\ S1\ S5|* \end{array}$$

通过中间部分对应关系消除冲突, 例如: $L1 \rightarrow S5$, 但 C_1 中也已经含有 $S5$, 故继续找到映射关系$S5 \rightarrow S1$, C_1 中没有 $S1$, 因此, 用 $S1$ 填补原来 $L1$ 的位置. 按照此方式依次消除冲突后得交叉结果:

$$\begin{array}{l} C_1: \ S1\ H1|H2\ S5\ L1|S3 \\ C_2: \ H1\ L1|S3\ S1\ S5|H2 \end{array}$$

该算子的优势是交叉过后, 后代的染色体仍具有每个基因不会重复的特点, 符合飞机队列的实际意义. 并且, 后代将部分保留父代的飞机序列, 也达到了进化的目的.

2.4.3 双交叉算子

如上所述, 交叉掩码具有保护优质染色体的能力, 但也有阻碍劣质染色体进化的弊端. 因此, 考虑使用双交叉算子来针对不同适应度染色体进行不同的交叉操作.

当个体适应度值满足: $f_i \geq k$(其中$f_i = \dfrac{1}{D_{target}+1}$, k=最后一代最大适应度*0.8), 认为该个体是优质染色体, 配合使用交叉掩码进行交叉操作; 反之, 劣质染色体采用部分映射杂交算子进行交叉操作.

当选取的两条父染色体, 一条为优质染色体, 另一条为劣质染色体时, 优质染色体使用交叉掩码进行交叉操作; 同时对两条父染色体进行部分映射杂交操作, 产生的两个后代染色体选取适应度较高者保留.

2.5 变异算子

根据排序的实际意义, 要求排序结果中包含所有飞机编码且每架飞机编码仅出现一次. 因此, 变异算子采用染色体内部变异的方式, 以满足上述要求. 并且考虑到重排算子的存在, 则应随机选择型号不同 (即编号字母不同) 的两个基因进行交换.

变异操作示例如下:

$$C: \ S5\ H1\ S1\ L1\ S3$$

随机选取产生变异基因为: $S1$, $L1$. 则变异操作后结果如下:

$$C': \ S5\ H1\ L1\ S1\ S3$$

2.6 重排算子

重排算子参考了文献[14]中命题 1 提出的一条理论: 同类型的飞机, 应按其预计到达跑道的时间先后依次着陆. 但该理论没有考虑航班优先级对降落次序产生的影响, 需要在使用时做相应处理. 重排操作是在变异操作之后, 其目的是使新生的飞机排序结果的适应度更高.

本文中的重排操作是指将变异操作后得到的飞机序列中的各类型 (H, L, S) 飞机分别按飞机的预计到达时间分别排序. 具有高优先级的飞机不参与这次排序, 防止高优先级飞机被延后.

具体的在本算法中使用重排算子的示例如下:

$$C: \ S5\ H1\ S1\ L1\ S3$$

对 C 进行重排操作后得:

$$C': \ S1\ H1\ S3\ L1\ S5$$

2.7 改进遗传算法流程

第一步. 获取待排序航班信息, 为飞机编码;

第二步. 随机产生初始种群 (染色体数量根据飞机数量决定, 可取与飞机数量相等的值);

第三步. 计算个体适应度值, 判断是否符合要求. 符合要求, 则输出结果; 否则进入下一步;

第四步. 根据前两代最优适应度染色体计算交叉

掩码 (从第三代开始);

第五步. 采用赌轮盘选择法从当代种群中选出双亲;

第六步. 采用自适应算法公式 (7) 计算 P_c, 根据双亲的适应度选择对应交叉方式进行交叉操作;

第七步. 采用自适应算法公式 (8) 计算 P_m, 进行变异操作;

第八步. 对染色体进行重排操作, 产生新个体;

第九步. 新生代是否达到种群数量要求, 未达要求返回第五步继续产生后代; 否则, 返回第三步循环寻优.

3 实验仿真及分析

本实验采用 C++编写算法仿真程序, 对 10 架次飞机在单跑道降落的情况进行模拟, 分别使用先到先服务算法和双交叉算子遗传算法对飞机待降队列进行排序, 计算两种排序结果的总延误代价及个飞机延误代价平方和用作排序结果对比; 记录遗传算法种群适应度变化情况, 检验收敛速度是否达到预期目标.

算法参数说明: 已知飞机的预计到达时间、飞机类型; 飞机类型为 H、S、L 的飞机, α 分别取 20、15、10, β=2, γ=600, 即飞机在延误 10 分钟以上, 延误代价将呈指数增长, 且重型飞机延误代价系数底数是轻型飞机的 2 倍, 中型飞机延误代价系数底数是轻型飞机的 1.5 倍; 种群初始数量取 20, 最大遗传代数取 100, 初始两代交叉掩码无法计算, 均取 1; 自适应算法中 P_{c1}、P_{c2} 分别取 0.9 和 0.6, P_{m1}、P_{m2} 分别取 0.1 和 0.001. 使用以上参数进行仿真, 得到的结果如表 2、表 3 所示.

表 2 FCFS 排序结果

降落序号	飞机编号	E_i(s)	S_i(s)	d_i(s)	K_i	D_i	D_i^2
1	L1	20	20	0	0.0000	0.00	0.00
2	L2	74	118	44	10.5214	462.94	214316.90
3	S1	137	192	55	15.9840	879.12	772852.70
4	L3	149	330	181	12.3257	2230.95	4977125.37
5	H1	322	404	82	21.9872	1802.95	3250642.64
6	H2	401	498	97	22.3716	2170.04	4709084.59
7	S2	585	612	27	15.4752	417.83	174583.32
8	L4	600	750	150	11.8921	1783.81	3181980.52
9	H3	712	824	112	22.7626	2549.41	6499509.28
10	S3	766	938	172	18.2973	3147.13	9904440.40
合计						15444.20	33684535.72

表 3 改进遗传算法排序结果

降落序号	飞机编号	E_i(s)	S_i(s)	d_i(s)	K_i	D_i	D_i^2
1	L1	20	20	0	0.0000	0.00	0.00
2	L2	74	118	44	10.5214	462.94	214316.90
3	L3	149	216	67	10.8048	723.92	524058.34
4	S1	137	290	153	17.9000	2738.71	7500507.98
5	H1	322	364	42	20.9943	881.76	777504.25
6	H2	401	458	57	21.3613	1217.59	1482536.53
7	L4	600	625	25	10.2930	257.33	66216.44
8	S2	585	699	114	17.1115	1950.71	3805253.72
9	S3	766	773	7	15.1218	105.85	11204.76
10	H3	712	847	135	23.3755	3155.70	9958433.47
合计						11494.51	24340032.39

根据表 2、表 3, 对比 S_i 项可知改进遗传算法将飞机队列降落耗时减少 91 秒, 且相邻飞机降落间隔满足尾流间隔标准要求; 对比 D_i 项可知改进遗传算法降低飞机延误总代价 3949.69, 相比减少了 25.57%; 比较各飞机D_i^2的大小, 可知改进遗传算法排序后的飞机延误分配更加均匀, 极少出现由个别航班承担重大延误代价的情况.

各飞机延误代价平方和变化曲线如图 2 所示. 种群在前 10 代收敛速度很快, 在第 10-50 代进化更加平稳, 在第 50 代左右达到稳定, 与文献[10]中的收敛曲线对比, 收敛曲线相似, 但平均提前 5 代达到稳定, 基本达到算法改进预期目的.

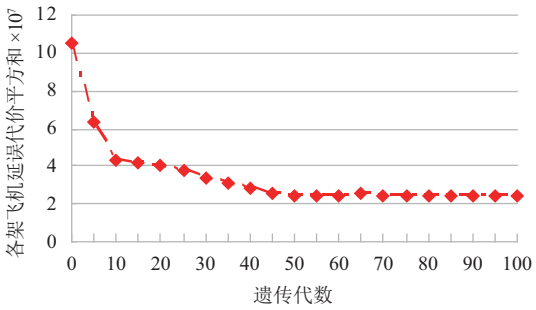

图 2 延误代价平方和曲线图

4 结束语

本文针对终端区飞机排序问题, 结合文献[10]提到的交叉掩码的使用方式, 再融入使用双交叉算子针对不同适应度染色体进行不同交叉操作的思想, 提出了一种改进的遗传算法. 根据实验数据, 证明该算法的收敛速度相比文献[10]中的算法有进一步的改进, 得到的排序结果也达到了降低航班延误总代价的效果, 且延误代价分配更加均匀.

参考文献

1 Venkatakrishnan CS, Barnett A, Odoni AR. Landings at Logan airport: Describing and increasing airport capacity. Transportation Science, 1993, 27(3): 211–227. [doi: 10.1287/trsc.27.3.211]

2 王莉莉, 史忠科, 张兆宁. 机场着陆排序的一种滑动窗优化算法. 中国民航学院学报, 2004, 22(6): 18–21.

3 Neuman F, Erzberger H. Analysis of sequencing and scheduling methods for arrival traffic. NASA-TM-102795, 1990.

4 Erzberger H, Nedell W. Design of automated system for management of arrival traffic. NASA TM 102201, 1989.

5 Neuman F, Erzberger H. Analysis of delay reducing and fuel saving sequencing and spacing algorithms for arrival traffic. NASA TM 103880, 1991.

6 Holland JH. Adaptation in Natural and Artificial Systems. Ann ArborMI: The University of Michigan Press, 1975.

7 常会贤, 曲仕茹. 终端区航班排序的遗传算法. 测控技术, 2010, 29(7): 91–93, 101.

8 白重阳, 张学军, 管祥民. 基于改进遗传算法的终端区排序研究. 航空计算技术, 2011, 41(5): 42–44, 48.

9 陈亮, 邹赟波, 吴值民. 改进遗传算法在终端区飞机排序中的应用. 军事交通学院学报, 2013, 15(1): 29–33.

10 张维杰, 龙华, 胡婷, 等. 改进的遗传算法在延误飞机进场排序中的应用. 信息技术, 2016, (7): 78–83.

11 陈长征, 王楠. 遗传算法中交叉和变异概率选择的自适应方法及作用机理. 控制理论与应用, 2002, 19(1): 41–43.

12 焦潇冰, 费向东, 谢泽辉. 基于改进的遗传算法航班进港排序模型研究. 计算机技术与发展, 2014, 24(2): 246–249.

13 张泳, 赵建民, 朱信忠, 等. 基于 SLP 方法的遗传算法在多目标布置设计中的应用. 计算机时代, 2010, (5): 1–3, 7.

14 孟祥伟, 张平, 李春锦. 到场飞机排序及调度问题的 Memetic 算法. 西南交通大学学报, 2011, 46(3): 488–493.