

改进乌鸦搜索算法在邮件集散中心航空运力调度中的应用^①



卢卓桓^{1,2}, 仰燕兰^{1,2}, 叶桦^{1,2}

¹(东南大学 自动化学院, 南京 210096)

²(复杂工程系统测量与控制教育部重点实验室, 南京 210096)

通信作者: 仰燕兰, E-mail: yyl@seu.edu.cn

摘要: 邮件集散中心航空运力调度涉及固定运力和备选运力两种调度对象, 本文在航空运力资源充足的前提下, 建立了以最小化运输成本为目标的优化模型, 研究了一种改进的乌鸦搜索算法求解方法. 首先根据问题的数学模型, 引入惩罚函数法将部分约束转化为惩罚项, 与目标函数共同构成适应度函数; 然后引入 Logistic 混沌映射提高初始种群的多样性; 根据问题的特点, 提出了基于个体最优追随机制和正余弦算法的位置更新策略, 并引入交叉变异机制以丰富搜索过程中种群的多样性. 通过大量算例实验分析, 证明了该算法的有效性与优越性.

关键词: 运力调度; 乌鸦搜索算法; 惩罚函数法; Logistic 混沌映射; 正余弦算法; 交叉变异

引用格式: 卢卓桓, 仰燕兰, 叶桦. 改进乌鸦搜索算法在邮件集散中心航空运力调度中的应用. 计算机系统应用, 2022, 31(9): 324-332. <http://www.c-s-a.org.cn/1003-3254/8710.html>

Application of Improved Crow Search Algorithm in Air Transport Capacity Scheduling of Mail Distribution Center

LU Zhuo-Huan^{1,2}, YANG Yan-Lan^{1,2}, YE Hua^{1,2}

¹(School of Automation, Southeast University, Nanjing 210096, China)

²(Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing 210096, China)

Abstract: The scheduling of air transport capacity by a mail distribution center involves fixed capacity and alternative capacity. This study establishes an optimization model to minimize transportation costs under the premise of sufficient air transport capacity and proposes an improved crow search algorithm. First, depending on the mathematical model of the problem, the penalty function method is employed to convert some constraints into penalty terms which then form a fitness function together with the objective function. Second, the Logistic chaotic map is utilized to improve the diversity of the initial population. In addition, considering the characteristics of the problem, the study proposes a location update strategy based on the individual optimal follow mechanism and the sine-cosine algorithm. Finally, a cross-mutation mechanism is introduced to enrich population diversity in the search process. The effectiveness and superiority of the algorithm are proved by a large number of cases.

Key words: capacity scheduling; crow search algorithm; penalty function; Logistic chaotic map; sine-cosine algorithm; cross-mutation

邮件集散中心(简称集散中心)作为全国邮政网络的重要枢纽, 主要负责对邮寄实物进行转发和运输. 数

据预测, 2021 至 2035 年全社会货运量年均增速约为 2%. 稳中有升的货物运输需求之下, 高价值、小批量、

① 收稿时间: 2021-12-15; 修改时间: 2022-01-13; 采用时间: 2022-02-17; csa 在线出版时间: 2022-06-24

时效强的货物运输需求快速攀升。在此背景下,航空快件运输凭借其时效优势成为集散中心的重要业务之一^[1]。当前,集散中心的航空运力主要为全货机(固定运力)和客机腹舱(备选运力)。由于两种运力方式在时间、成本、载运量等方面各异,不同运力调度方案的运输效率、综合成本差异较大。因此,合理调度航空运力成为提升集散中心航空快件保障能力、处理效率,实现降本增效的重要举措。

目前,国内外关于航空运力调度问题(air capacity scheduling problem, ACSP)的研究主要集中在民航公司的飞机排班问题上,包括机型指派、机队分配、航班时刻优化等。Hane等研究了一般形式下的机队分配问题,并建立整数规划模型成功求解了某大型航空公司的机型分配问题^[2]。Rexing等提出了带时间窗的机队分配模型,并开发了速度优先的直接算法和内存优先的迭代算法进行求解^[3]。Khanmirza等针对航空公司的时刻表设计和机队分配提出了一种并行主从遗传算法进行求解,该算法能够在短时间内解决大规模的调度问题^[4]。孙宏等在分析了国内航空公司的航线结构和飞机排班要求后引入了“航班节”的概念,提出了用于描述单枢纽航线结构下飞机排班问题的排序模型,并设计一种分阶段指派算法用于求解^[5]。

相比于目前的研究方向,集散中心的航空运力调度问题(air capacity scheduling problem of mail distribution center, ACSP-MDC)具有一定的差异性。一方面,集散中心航空运力调度的重点不再是客机的排班,而是综合调度两种运力,使既定目标最优。另一方面,飞机排班问题是根据航班计划和现有资源的情况为每个航班指定一架具体的飞机。而ACSP-MDC不仅需要为航线选择合适的飞机,还需要为选中飞机分配载运量,使集散中心当天所有的邮寄实物能顺利运输到各个目的地。

ACSP-MDC作为ACSP的扩展问题,其基本目标是在一定航空运力资源的约束下得到合理的调度方案,使空运成本最低。近年来,元启发式算法因其具有较好的适用性和鲁棒性,已经成为解决此类问题的主流方法。李耀华等将飞机排班计划一体化优化模型问题归结为多目标车辆路径问题,并提出一种自适应单亲遗传算法进行求解^[6]。李红启等构建了航空快递的地面集货运力和航空运力协同调度问题的混合整数规划模型,并设计一种两阶段启发式算法进行求解^[7]。本文在对比

了各种元启发式算法之后,考虑到乌鸦搜索算法(crow search algorithm, CSA)具有结构简单,控制参数少,寻优能力较强等优点^[8],且算法的有效性和实用性已在作业车间调度、工件设计、图像分割等不同工程领域中得到证明^[9-11],选择其作为本文的基础算法。本文针对ACSP-MDC的特点,引入惩罚函数法以消除部分约束,并在CSA的基础上,对初始化种群、乌鸦位置更新等环节进行改进,同时引入交叉变异机制,提出混合交叉策略和自适应变异策略以丰富种群的多样性。

1 ACSP-MDC 问题模型

1.1 问题描述

集散中心经过邮件分拣、邮件过检、封发装箱等前置环节之后,需要调度航空运力,将邮寄实物发往不同的城市。可供集散中心调度的航空运力包括集散中心所拥有的全货机和民航客机的腹舱。全货机通常在夜间飞行,其优势是起飞时间灵活、运载量大,且变动成本(单位:元/(t·km))较低,但是每启动一架全货机,需要额外支出维修费、起降停场费、地面服务费等固定成本,因此,全货机的载运率越高,启动全货机运输会更加划算。客机航班的起飞时间由航空公司制定,变动成本由航空公司或第三方货运代理制定,每架客机能提供的货运量受到飞机型号和当日旅客的行李量所限制。因此,集散中心在选择备选运力时只能在航空公司制定的航班计划中选择合适的航班运载邮寄实物。

综上所述,邮件集散中心航空运力调度问题(ACSP-MDC)可定义为:集散中心从固定运力和备选运力中选择合适的运力资源,合理安排每种资源的飞行航线和载运量,在满足一定约束的前提下(载重约束、航线约束等)最小化运输成本。

1.2 数学模型

为了便于描述ACSP-MDC的数学模型,首先引入如下的符号定义:

K : 航线集合, $K = \{1, 2, \dots, l\}$;

I : 全货机集合, $I = \{1, 2, \dots, n\}$;

J_k : 航线 k 的客机航班集合, $J_k = \{1, 2, \dots, m_k\}$;

d_k : 航线 k 的空距;

w_k : 航线 k 的待运货物量;

$cargo_ml_i$: 全货机 i 的最大载运量;

$cargo_fc_i$: 启用全货机 i 的固定成本;

$cargo_vc_i$: 全货机 i 的变动成本;

$civil_ml_{kj}$: 航线 k 航班 j 的最大载运量;
 $civil_vc_{kj}$: 航线 k 航班 j 的变动成本;
 M : 充分大的正数;
 x_{ik} : 全货机 i 飞航线 k 的载运量;
 y_{ik} : 0-1 变量, 是否使用全货机 i 飞航线 k ;
 u_{kj} : 航线 k 航班 j 的载运量.

ACSP-MDC 的数学模型为:

目标函数:

$$\min F_c = \sum_{i \in I} \sum_{k \in K} x_{ik} d_k cargo_vc_i + \sum_{i \in I} \sum_{k \in K} y_{ik} cargo_fc_i + \sum_{k \in K} \sum_{j \in J_k} u_{kj} d_k civil_vc_{kj} \quad (1)$$

约束条件:

$$\sum_{i \in I} x_{ik} + \sum_{j \in J_k} u_{kj} \geq w_k, \forall k \in K \quad (2)$$

$$0 \leq u_{kj} \leq civil_ml_{kj}, \forall k \in K, \forall j \in J_k \quad (3)$$

$$0 \leq \sum_{k \in K} x_{ik} \leq cargo_ml_i \sum_{k \in K} y_{ik}, \forall i \in I \quad (4)$$

$$\sum_{k \in K} y_{ik} \leq 1, \forall i \in I \quad (5)$$

$$0 \leq x_{ik} \leq M y_{ik}, \forall i \in I, \forall k \in K \quad (6)$$

目标函数式 (1) 由 3 个部分组成, 分别是全货机的变动成本、全货机的固定成本和客机航班的变动成本, 其中, 变动成本的计算方法为: 飞机载运量×航距×飞行的单位成本; 约束 (2) 是航线载重约束, 表示每条航线全货机和客机航班的载运量之和需要满足该航线邮寄实物的待载运总量; 约束 (3)–(4) 分别是客机和全货机的载重约束, 表示每种运力资源的载运量大于 0 且不能超过其最大载重; 约束 (5) 表示一架全货机最多只能飞一条航线; 约束 (6) 约束了 x_{ik} 和 y_{ik} 之间的关系, 表示只有选中全货机 i 飞航线 k , 才会为其分配飞往航线 k 的载运量, 否则全货机 i 到航线 k 的载运量为 0.

2 改进乌鸦搜索算法

乌鸦搜索算法是基于乌鸦藏食和窃食的社会性行为进行开发的, 其位置更新公式如式 (7) 所示:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times fl \times (M_j^t - X_i^t), & r_2 \geq AP \\ \text{a random position,} & \text{otherwise} \end{cases} \quad (7)$$

其中, X_i^t 表示第 i 只乌鸦在第 t 次迭代时所处的位置,

M_j^t 表示第 j 只乌鸦在第 t 次迭代时的最佳位置, 即乌鸦 j 的藏食点. r_1 和 r_2 是 $[0, 1]$ 之间均匀分布的随机数; AP 表示乌鸦的感知概率; fl 表示乌鸦的飞行步长.

朴素 CSA 的算法流程如图 1 所示.

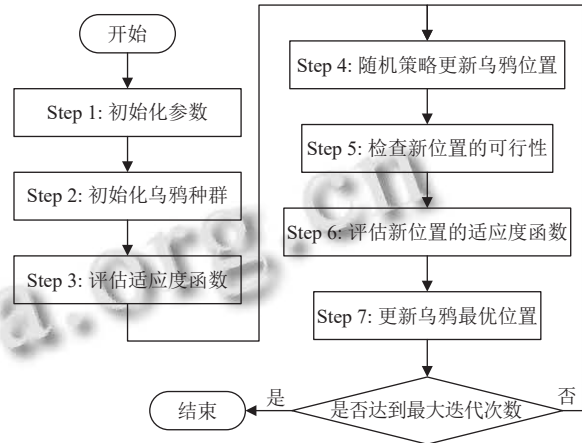


图 1 朴素 CSA 算法流程图

为了使 CSA 能够有效地解决 ACSP-MDC, 本文根据具体场景的约束, 对 CSA 在求解过程中的各个步骤进行一定程度的改进.

(1) 使用惩罚函数法消除部分约束, 拓宽乌鸦搜索的可行域, 避免在 Step 5 检查乌鸦新位置可行性时, 过多乌鸦因为不满足原有约束而被丢弃;

(2) 在 Step 2 初始化种群时引入基于幂函数载波的 Logistic 混沌映射, 提高初始种群的多样性;

(3) 针对 ACSP-MDC 的特殊性, 提出了基于个体最优追随的位置更新策略取代 Step 4 的随机跟随策略, 并嵌入正弦算子, 提高 CSA 的搜索能力;

(4) 引入交叉变异机制, 根据 ACSP-MDC 的特点, 提出两种不同的交叉策略和一种自适应变异策略, 维持搜索过程中种群的丰富性, 让算法有更大概率跳出局部最优, 从而找到全局最优解.

2.1 基于惩罚函数法的约束消除

在解决约束优化问题时, 通过适当地引入约束处理技术, 可以平衡目标函数和约束条件的信息, 使优化算法更加高效. 惩罚函数法是其中一种常见且有效的技术, 其基本原理是: 将问题的约束条件转换为一个惩罚函数并加在原目标函数中, 构造出一个增广目标函数. 若当前解不满足约束条件, 增广目标函数就会加大惩罚, 淘汰可行域之外的解^[12]. 通过引入惩罚函数, 可以减少 ACSP-MDC 中的约束条件, 扩大解的可行域,

避免部分乌鸦因为不满足约束而被视为无效个体,同时也可以省去算法中处理特定约束的步骤和时间。

本问题的约束条件为式(2)–式(6),其中,约束(3)和约束(4)在搜索过程中通过边界修复来实现,如式(8)和式(9)所示:

$$u_{kj} = \min \{ \max \{ u_{kj}, 0 \}, \text{civil_ml}_{kj} \} \quad (8)$$

$$\sum_{k \in K} x_{ik} = \min \left\{ \max \left\{ \sum_{k \in K} x_{ik}, 0 \right\}, \text{cargo_ml}_i \sum_{k \in K} y_{ik} \right\} \quad (9)$$

由于存在约束(5)–式(6),使得 $x_i = [x_{i0}, x_{i1}, \dots, x_{iL}]$ 中只有一个维度的值大于0,其余维度值均为0,故可将式(9)转化为式(10):

$$x_{ik} = \min \{ \max \{ x_{ik}, 0 \}, \text{cargo_ml}_i \} \quad (10)$$

约束(2)和约束(5)则通过进入惩罚项来消除,结果如式(11)所示:

$$F_p = \sigma_1 \sum_{k \in K} \left(\sum_{i \in I} x_{ik} + \sum_{j \in J_k} u_{kj} - w_k \right)^2 + \sigma_2 \sum_{i \in I} \max \left(\sum_{k \in K} y_{ik} - 1, 0 \right) \quad (11)$$

其中, σ_1 和 σ_2 为惩罚因子,在本文中,将其设置为足够大的正数即可。

引入惩罚项后,评价乌鸦位置优劣的适应度函数可定义为:

$$F = F_c + F_p \quad (12)$$

其中, F_c 为目标函数, F_p 为惩罚项,式(12)表明,适应度值越小,乌鸦的位置越优。

2.2 初始化种群

初始化种群的好坏直接影响智能群体算法的寻优精度和收敛速度,朴素CSA采用随机方法生成初始化种群,这种生成方式有两个弊端。其一,容易造成乌鸦的位置分布不均匀,导致乌鸦种群难以搜索到整个空间,易陷入早熟,最终只能得到局部最优解;其二,对于约束较为复杂的问题,随机生成的乌鸦容易逃逸出搜索空间的可行域。尽管本文通过引入惩罚函数,边界修复等方式消除了约束,扩大了乌鸦搜索的可行域,初始化高质量的种群仍是必要的,因为初始化乌鸦个体如果能尽量落在满足原约束的可行域内,它们将更靠近最优解,搜索效率会更高。

因此,本文引入了基于幂函数载波技术的Logistic混沌映射^[13]来提高乌鸦种群的多样性,同时借鉴文献^[14]对0-1变量的混沌初始化方法,使初始的乌鸦

个体更加靠近最优位置,为后续的搜索打下基础。

2.2.1 幂函数载波技术

Logistic映射方程为:

$$z_{n+1} = \mu z_n (1 - z_n) \quad z_n \in [0, 1], n = 0, 1, \dots \quad (13)$$

其中, z_n 为生成的 $[0, 1]$ 区间的混沌值,迭代初始时 z_0 为随机生成的混沌初值,但不可取0、0.25、0.75、1; $\mu \in [0, 4]$,为控制参数,当取4时,系统处于完全混沌状态。虽然Logistic映射产生的混沌变量具有遍历性,但由于其轨道点分布不均匀,导致落在区间两端的点要比区间内部的点更多。为了纠正这一问题,充分发挥Logistic混沌变量的遍历性,文献^[14]提出了式(14)所示的幂函数载波技术:

$$z_n' = \begin{cases} z_n^p, & z_n \in [0, a] \\ z_n, & z_n \in [a, b] \\ z_n^q, & z_n \in [b, 1] \end{cases} \quad (14)$$

其中, $0 < a < b < 1, 0 < p < 1, q > 1, z_n$ 为式(13)产生的混沌变量, z_n' 为载波后的混沌变量。通过幂函数载波,靠近区间左边的点会右移,靠近区间右边的点会左移,从而使得载波后的点分布更加均匀,遍历性更好,图2展示了这一点。本文取 $a = 0.15, b = 0.96, p = 0.46, q = 13.6$ 。

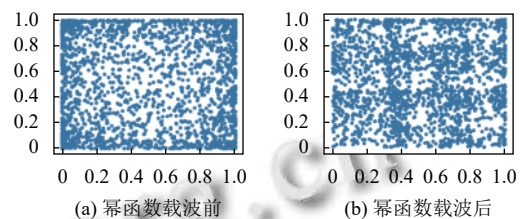


图2 幂函数载波前后 Logistic 映射的遍历性

2.2.2 基于幂函数载波的初始化方法

本文对变量 x_{ik} 和 u_{kj} 进行实数编码,组成一个乌鸦个体。对于备选运力 u_{kj} ,直接使用式(15)进行初始化。

$$u_{kj} = \text{rand} \times \text{civil_ml}_{kj}, \text{rand} \in [0, 1] \quad (15)$$

对于固定运力 x_{ik} ,由于其受到了一架全货机只能飞行一条航线的限制,因此,可以先对0-1变量 y_{ik} 进行初始化,使其满足约束(5),然后再根据 y_{ik} 的值对 x_{ik} 进行赋值,如式(16)所示:

$$x_{ik} = \begin{cases} \text{rand} \times \text{cargo_ml}_i, & \text{if } y_{ik} = 1, \text{rand} \in [0, 1] \\ 0, & \text{if } y_{ik} = 0 \end{cases} \quad (16)$$

假设全货机数量为 n ,航线数量为 l ,则对于每只乌鸦,初始化 $y_{ik} (i = 1, 2, \dots, n, k = 1, 2, \dots, l)$ 的步骤如下:

Step 1. 赋给式(13)随机初值,生成 n 维的混沌序列 seq 。

Step 2. 对混沌序列应用式 (14), 将其载波成新的序列 seq' .

Step 3. 将区间 $[0, 1]$ $l+1$ 等分, 生成区间编号为 $0-l$.

Step 4. 判断序列 $seq'[i]$ 所在的区间范围, 并置:

$$y_{ik} = \begin{cases} 0, & \text{第}i\text{个混沌变量不落在第}k\text{个区间} \\ 1, & \text{第}i\text{个混沌变量落在第}k\text{个区间} \end{cases} \quad (17)$$

由上面 4 个步骤实现的 y_{ik} 的初始化可以满足约束 (5), 然后应用式 (16) 初始化 x_{ik} 的值, 应用式 (15) 初始化 u_{kj} 的值, 将 x_{ik} 和 u_{kj} 组成一个完整的乌鸦个体, 由此生成的乌鸦比随机生成的乌鸦有更大的概率在原约束下的可行域内或更加靠近可行域, 且分布更加均匀. 重复迭代即可生成整个种群.

2.3 改进的位置更新策略

2.3.1 个体最优追随机制

初始生成的乌鸦已经满足了约束 (5), 此时得到的个体是较好的, 如果仍然应用式 (7) 所示的朴素 CSA 的随机跟随策略更新乌鸦位置, 会导致大量乌鸦在探索新位置时, 远离原约束下的可行域.

假设如下情况: 集散中心共有 2 架全货机, 当天有 2 条航线需要飞行. 在初始化种群后, 进入搜索阶段. 仅考虑全货机的调度, 设某两个乌鸦个体初始化情况如式 (18) 所示, 乌鸦 i 跟随乌鸦 j 的可能结果如图 3 所示. 可以看到, 跟随后乌鸦 i 已经不能满足约束 (5), 即同一架全货机飞行两条航线, 此时已偏离最优解, 可认为是一次无效跟随.

$$Crow_i = \begin{pmatrix} 20 & 0 \\ 0 & 8 \end{pmatrix}, Crow_j = \begin{pmatrix} 0 & 15 \\ 10 & 0 \end{pmatrix} \quad (18)$$

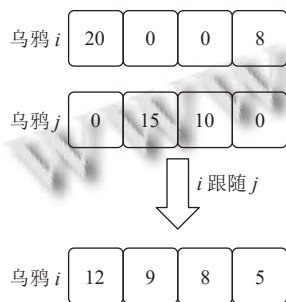


图 3 乌鸦 i 跟随乌鸦 j 的个体编码

因此, 为了保证乌鸦每次跟踪都能更加靠近最优解, 本文提出了个体最优追随机制, 即在跟踪阶段, 每只乌鸦不再随机跟踪种群中的另一只乌鸦, 而是追随上一代的个体最优位置, 这种调整能够保证乌鸦不断靠近最优解, 并提高算法的收敛速度. 改进后的位置更

新策略如式 (19) 所示.

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times fl \times (M_i^t - X_i^t), & r_2 \geq AP \\ \text{a random position,} & \text{otherwise} \end{cases} \quad (19)$$

2.3.2 正余弦优化策略

考虑到固定的参数设置使得朴素 CSA 的搜索能力受限, 本文引入正余弦算法 (sine cosine algorithm, SCA) 作为局部优化算子来提高 CSA 的全局探索和局部开发的能力^[15,16]. 位置更新公式如式 (20) 所示:

$$X_i^{t+1} = \begin{cases} X_i^t + R_1 \times \cos R_2 \times |R_3 \times M_i^t - X_i^t|, & R_4 \geq 0.5 \\ X_i^t + R_1 \times \sin R_2 \times |R_3 \times M_i^t - X_i^t|, & R_4 < 0.5 \end{cases} \quad (20)$$

其中, R_1 为控制参数, 决定位置更新的方向, 其计算方法为 $R_1 = a - a \times \frac{t}{t_{\max}}$. R_1 的大小控制着乌鸦的搜索范围, 随着迭代次数增加, R_1 逐渐减小, 乌鸦的搜索范围随之减小, 最终收敛在一个最优位置, a 为常数, 本文取 $a = 2$; R_2, R_3, R_4 是均匀分布的随机数, $R_2 \in [0, 2\pi]$, 控制位置更新的步长, $R_3 \in [0, 2]$, 控制个体最优位置的影响程度, $R_4 \in [0, 1]$, 控制算法是使用正弦还是余弦操作.

正余弦算法本身具有参数自适应的机制, 能够很好地弥补 CSA 参数固定的缺点. 在迭代前期, R_1 较大, 算法探索全局的能力较强, 在迭代后期, R_1 较小, 算法的局部开发能力较强; 同时 R_2 的随机性也控制着正余弦函数的振幅, 使得同一个方向上的不同区段范围都有可能被搜索到, 以此更好地平衡搜索过程中的全局探索能力和局部开发能力, 并最终收敛于最优解或较满意的解.

2.4 交叉变异机制

改进后的位置更新策略能够大大加快算法的收敛速度, 但却不利于维持搜索过程中种群的多样性, 导致算法易陷入局部最优解. 为了让算法有更大的机会跳出局部最优, 本文受遗传算法启发, 引入了交叉变异机制来提高算法全局寻优的性能.

2.4.1 混合交叉策略

在 ACSP-MDC 中, 全货机和客机的约束不同, 故本文提出一种混合交叉策略, 将乌鸦个体按全货机和客机分为两段, 分别记为 X 段和 U 段, 各自采用不同的交叉策略.

对于 X 段, 为了保证交叉后的个体满足约束 (5), 采用如下交叉策略: 将两个父个体的 X 段按航线数量进行切分, 切分点即为交叉点, 对于每个交叉点, 随机生成 $[0, 1]$ 之间均匀分布的随机数 $random$, 若 $random < P_c$ (给定概率), 则进行交叉操作.

对于U段,如果生成的[0,1]区间的随机数 $random < P_c$,则采用式(21)进行实数交叉,更新个体的U段。

$$u_i^{t+1} = \alpha \times u_i^t + (1 - \alpha) \times u_j^t \quad (21)$$

其中, α 为[0,1]之间的随机数, u_i^t 为种群中的一个乌鸦个体, u_j^t 为种群中随机选择的另外一个乌鸦个体. 对于第*i*只乌鸦,交叉操作的具体过程为:

Step 1. 在种群中随机选取另一只乌鸦*j*;

Step 2. 按照X段的交叉策略进行交叉,交叉后可得到乌鸦*i*的X段和两个子X段;

Step 3. 按照U段的交叉策略生成新的U段;

Step 4. 将Step 2和Step 3生成的X段和U段组合起来,生成3只乌鸦,应用适应度函数并保留适应度值最小的乌鸦。

混合交叉示意图如图4所示。

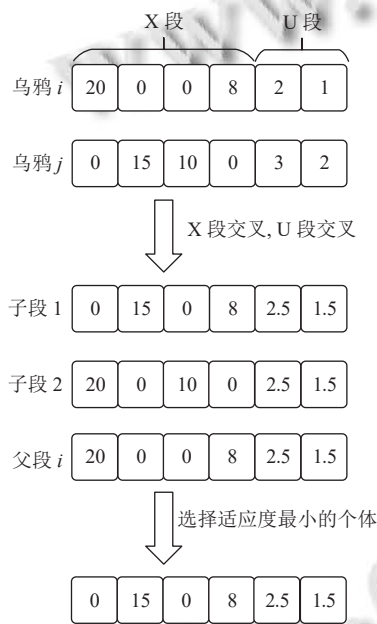


图4 混合交叉示例

2.4.2 自适应变异策略

常见的变异操作包括两点交换变异和单点特殊变异,受本文实际场景的约束,一架全货机只能飞行一条航线,交换变异有很大概率会造成不可行解,故本节采用单点特殊变异,具体操作如式(22)所示:

$$X_{i,j}^{t+1} = \begin{cases} M_{i,j}^t + N\left(0, |M_{i,j}^t - X_{i,j}^t|\right), & \text{if } random < P_m \\ X_{i,j}^t, & \text{otherwise} \end{cases} \quad (22)$$

其中, $M_{i,j}^t$ 为乌鸦*i*第*j*维的个体最优值, $X_{i,j}^t$ 为交叉后的乌鸦*i*的第*j*维的值, $N\left(0, |M_{i,j}^t - X_{i,j}^t|\right)$ 表示期望为0,方

差为 $|M_{i,j}^t - X_{i,j}^t|$ 的正态分布, $random$ 为[0,1]区间的随机数, P_m 为变异概率。

为了保持种群的优良性,对较好的个体减少变异的概率,对较差的个体增加变异的概率,本文提出了如下的自适应变异策略。

$$P_m = \frac{i}{N} \times p_{max} + p_{min}, \quad i = 1, 2, \dots, N \quad (23)$$

其中, i 为乌鸦按适应度值的大小升序排列后的序号, N 为种群大小, p_{min} 为设定的最小变异概率, p_{max} 为设定的最大变异概率, P_m 的取值范围为 $(p_{min}, p_{min} + p_{max})$. 式(23)所示的自适应变异概率是线性递减的,当个体越优时, P_m 越小,变异的机会小,有利于优良个体的保留;当个体较差时, P_m 变大,变异的机会随之增大,有利于较差个体的进化. 采用自适应变异策略可以在保持种群优良性的同时,丰富种群的多样性。

2.5 算法总流程

改进乌鸦搜索算法的详细步骤如图5所示。

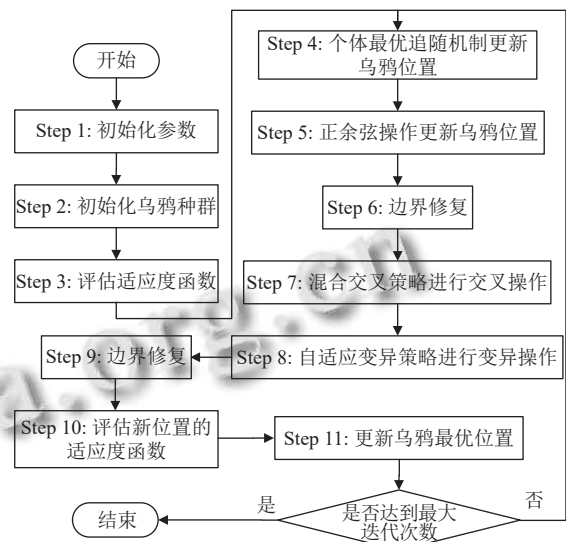


图5 改进CSA算法流程图

3 问题求解

以某物流公司所拥有的全货机数据和该公司某段时间的物流情况为参考,设计如表1-表4的测试用例,以验证本文算法的有效性。

表1给出了航线的信息,表2给出了可供调度的全货机信息,表3给出对应航线的客机航班信息,表4给出了7组运输量组合,每组都给出了不同航线需要运输的货物量的情况。

表1 航线信息

航线	空距 (km)
L1	1223
L2	986
L3	1104

表2 全货机信息

全货机	最大载荷 (t)	固定成本 (元)	变动成本 (元/(t·km))
Cargo-1	16	22349	2.23
Cargo-2	18	23767	2.17
Cargo-3	36	35259	1.62

表3 客机航班信息

客机航班	最大载荷 (t)	变动成本 (元/(t·km))
L1-Civil-1	1.4	2.55
L1-Civil-2	2.5	2.55
L1-Civil-3	2.75	2.55
L1-Civil-4	1.5	2.55
L1-Civil-5	1.2	2.65
L1-Civil-6	1	2.65
L1-Civil-7	1.5	2.65
L1-Civil-8	2.5	2.65
L1-Civil-9	2.4	2.52
L1-Civil-10	3.75	2.52
L2-Civil-1	2.7	2.64
L2-Civil-2	3	2.64
L2-Civil-3	2	2.64
L2-Civil-4	3	2.64
L2-Civil-5	1.5	2.76
L2-Civil-6	2.5	2.76
L2-Civil-7	1.5	2.76
L2-Civil-8	2	2.76
L3-Civil-1	1.6	2.38
L3-Civil-2	2.5	2.38
L3-Civil-3	2.75	2.38
L3-Civil-4	1.5	2.38
L3-Civil-5	1.2	2.45
L3-Civil-6	1	2.45
L3-Civil-7	1.5	2.45
L3-Civil-8	2.5	2.30
L3-Civil-9	2.4	2.30
L3-Civil-10	3.75	2.30

表4 航线运输量 (t)

货物量组合编号	L1货物量	L2货物量	L3货物量
W1	6	5	8
W2	6	18	46
W3	10	45	12
W4	38	8	19
W5	40	25	11
W6	35	50	20
W7	35	30	40

设置种群数为 60, 迭代次数为 100, 乌鸦跟踪步长 $f1$ 为 2, 感知概率 AP 为 0.1, 正余弦算子的辅助常数 α 为

2, 交叉概率 P_c 为 0.5, 最大变异概率 p_{max} 为 0.5, 最小变异概率 p_{min} 为 0.

使用本文改进的 CSA 算法求解结果如表 5 所示, 其中, 平均值和最优值是运行 50 次所得到的结果, 最优的调度方案记录为: 航线: 飞机编号 (载重). 从表中可以看到, 每组算例的调度方案都能满足各自航线的载重需求, 同时, 每组结果的平均值和最优值相差很小, 最大的一组 W7 仅相差了 318.96, 表明本文改进的 CSA 算法具有较好的鲁棒性.

表5 改进 CSA 算法求解结果

组合编号	平均值 (元)	最优值 (元)	最优调度方案
W1	51 871.001	51 870.139	L1: Civil-9 (2.4), Civil-10 (3.6)
			L2: Civil-1 (2.7), Civil-2 (2.3) L3: Civil-1 (0.6), Civil-8~10 (满载)
W2	191 476.050	191 473.430	L1: Civil-9 (2.4), Civil-10 (3.6)
			L2: Civil-1~7 (满载), Civil-8 (1.8) L3: Cargo-3 (36), Civil-1, 8~10 (满载), Civil-2 (1)
W3	178 027.128	178 024.177	L1: Civil-1, 9~10 (满载), Civil-2 (2.45)
			L2: Cargo-3 (36), Civil-1~3 (满载), Civil-4 (1.3) L3: Civil-1~2, 8~10 (满载), Civil-3 (0.5)
W4	183 065.962	183 064.319	L1: Cargo-3 (36), Civil-9 (2)
			L2: Civil-1~3 (满载), Civil-4 (0.3) L3: Civil-1~5, 8~10 (满载), Civil-6 (0.4)
W5	227 690.116	227 662.792	L1: Cargo-3 (36), Civil-9~10 (满载)
			L2: Cargo-2 (18), Civil-1~2 (满载), Civil-3 (1.3) L3: Civil-1, 8~10 (满载), Civil-2 (2)
W6	306 470.770	306 449.088	L1: Cargo-2 (18), Civil-1~6, 9~10 (满载), Civil-7 (0.5)
			L2: Cargo-3 (36), Civil-1~5 (满载), Civil-6 (1.8) L3: Civil-1~5, 8~10 (满载), Civil-6 (1.4)
W7	329 141.530	328 822.572	L1: Cargo-2 (18), Civil-1~6, 9~10 (满载), Civil-7 (0.5)
			L2: Cargo-1 (16), Civil-1~5 (满载), Civil-6 (1.8) L3: Cargo-3 (36), Civil-8 (2), Civil-9 (2)

图 6 展示了朴素 CSA, SCA-CSA 和改进的 CSA 三种算法在求解 W1-W7 时的表现. 从图中可以看出

改进的 CSA 相比于朴素 CSA 和 SCA-CSA 具有更好的求解精度, 在面对不同问题时改进的 CSA 也表现出更好的鲁棒性.

针对 W7 这组测试用例, 图 7 展示了 3 种算法的收敛曲线, 可以看出改进 CSA 算法的求解精度和收敛速度都优于其他两种算法.

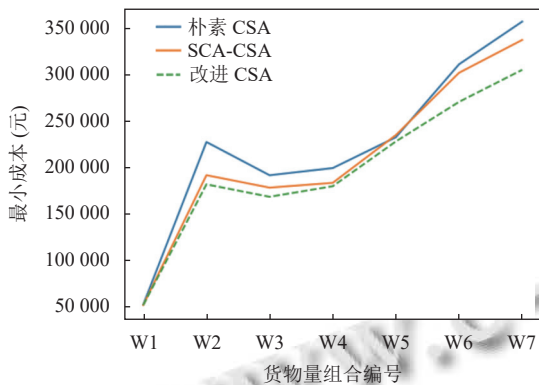


图 6 不同算法在不同货物量组合下的最小成本对比

进一步扩展问题的规模, 表 6 所示的 8 个问题分别是对航线, 全货机和客机航班进行一定程度的扩充, 并为每条航线随机分配该航线的货物量. 对每个问题都进行 50 次的测试, 测试结果如表 7 所示. 从表中可以看出, 本文改进的 CSA 在求解 ACSP-MDC 问题时的表现优于朴素的 CSA 和 SCA-CSA.

4 结束语

本文针对邮件集散中心航空运力调度问题, 在航空运力资源充足的情况下, 建立了以最小化运输成本为目标的优化模型. 针对 ACSP-MDC 的特点, 本文在

朴素 CSA 的基础上, 做了一系列的改进, 包括引入惩罚函数消除部分约束, 扩大乌鸦搜索的可行域, 根据问题的实际场景提出新的位置更新策略, 并引入交叉变异机制, 丰富种群的多样性, 提高全局寻优的性能. 改进后的 CSA 相较于朴素 CSA 和 SCA-CSA 的收敛速度更快, 求解结果更优, 鲁棒性更好, 且能有效地解决邮件集散中心的航空运力调度问题, 为集散中心空运业务降本增效提供合理的解决方案.

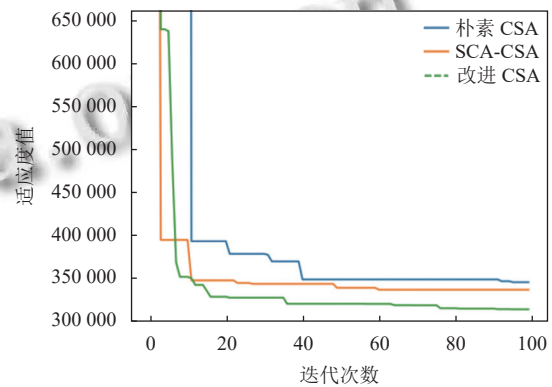


图 7 W7 测试用例下 3 种算法的收敛曲线

表 6 扩充仿真实例

问题名称	问题规模 (航线×全货机数量×某航线客机航班数量)
Q01	3×3×12
Q02	3×4×12
Q03	4×4×16
Q04	4×5×16
Q05	5×4×16
Q06	5×5×16
Q07	10×6×20
Q08	15×10×20

表 7 3 种算法结果 (适应度值) 对比

问题名称	朴素CSA		SCA-CSA		改进CSA	
	平均值	最优值	平均值	最优值	平均值	最优值
Q01	300960.367	246051.280	244052.289	228035.443	227569.000	227403.486
Q02	387825.613	365914.187	382642.019	369448.019	363005.640	361528.874
Q03	552106.010	512588.810	530026.129	520600.596	504017.461	497835.328
Q04	687576.373	631556.606	644197.173	614650.975	616349.831	602288.801
Q05	568543.435	531674.568	544765.097	511265.430	488187.689	478596.023
Q06	1179713.876	588524.386	622298.476	594413.010	582950.077	574948.055
Q07	8863332.794	679751.593	672580.417	645840.328	649923.425	623579.002
Q08	34623278.211	30062778.042	1242399.949	996161.973	1047232.115	929586.679

参考文献

1 郭雪林, 柴鹏伟. 中国邮政航空南京航空物流枢纽建设实践与思考. 空运商务, 2021, (10): 35-40. [doi: 10.3969/j.issn.1671-3095.2021.10.011]

2 Hane CA, Barnhart C, Johnson EL, et al. The fleet assignment problem: Solving a large-scale integer program. Mathematical Programming, 1995, 70(1-3): 211-232. [doi: 10.1007/BF01585938]

- 3 Rexing B, Barnhart C, Kniker T, *et al.* Airline fleet assignment with time windows. *Transportation Science*, 2000, 34(1): 1–20. [doi: [10.1287/trsc.34.1.1.12277](https://doi.org/10.1287/trsc.34.1.1.12277)]
- 4 Khanmirza E, Nazarahari M, Haghbeigi M. A heuristic approach for optimal integrated airline schedule design and fleet assignment with demand recapture. *Applied Soft Computing*, 2020, 96: 106681. [doi: [10.1016/j.asoc.2020.106681](https://doi.org/10.1016/j.asoc.2020.106681)]
- 5 孙宏, 杜文. 航空公司飞机排班问题的分阶段指派算法. *系统工程学报*, 2003, 18(2): 168–172. [doi: [10.3969/j.issn.1000-5781.2003.02.013](https://doi.org/10.3969/j.issn.1000-5781.2003.02.013)]
- 6 李耀华, 谭娜. 基于遗传算法的飞机一体化排班优化方法. *控制工程*, 2017, 24(2): 435–440.
- 7 李红启, 刘寅莹, 简晓荣. 航空快递的地面集货运力与航空运力调度问题. *中国民航大学学报*, 2018, 36(2): 37–44. [doi: [10.3969/j.issn.1674-5590.2018.02.009](https://doi.org/10.3969/j.issn.1674-5590.2018.02.009)]
- 8 Askarzadeh A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Computers & Structures*, 2016, 169: 1–12.
- 9 Han XX, Xu QX, Yue L, *et al.* An improved crow search algorithm based on spiral search mechanism for solving numerical and engineering optimization problems. *IEEE Access*, 2020, 8: 92363–92382.
- 10 刘凯, 黄辉先, 赵骥. 求解模糊作业车间调度问题的混沌乌鸦搜索算法. *传感器与微系统*, 2021, 40(6): 110–113, 117.
- 11 康丽锋, 吴锋. 基于乌鸦搜索优化算法的多级阈值图像分割方法. *西南师范大学学报(自然科学版)*, 2021, 46(1): 38–43.
- 12 王兴华. 惩罚函数法的改进算法及应用研究 [硕士学位论文]. 秦皇岛: 燕山大学, 2009.
- 13 唐巍. 基于幂函数载波的混沌优化方法及其应用. *控制与决策*, 2005, 20(9): 1043–1046. [doi: [10.3321/j.issn:1001-0920.2005.09.017](https://doi.org/10.3321/j.issn:1001-0920.2005.09.017)]
- 14 桑晓丹, 罗兴国, 禹春来, 等. 求解 0-1 整数规划问题的混沌遗传算法. *计算机应用研究*, 2011, 28(7): 2443–2445. [doi: [10.3969/j.issn.1001-3695.2011.07.010](https://doi.org/10.3969/j.issn.1001-3695.2011.07.010)]
- 15 Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-based Systems*, 2016, 96: 120–133. [doi: [10.1016/j.knosys.2015.12.022](https://doi.org/10.1016/j.knosys.2015.12.022)]
- 16 肖子雅, 刘升, 韩斐斐, 等. 正弦余弦指引的乌鸦搜索算法研究. *计算机工程与应用*, 2019, 55(21): 52–59. [doi: [10.3778/j.issn.1002-8331.1808-0077](https://doi.org/10.3778/j.issn.1002-8331.1808-0077)]

(校对责编: 孙君艳)