

混合蝙蝠算法在包装件配送中的应用研究

彭维

(重庆城市管理职业学院, 重庆 401331)

摘要: **目的** 使蝙蝠算法 (BA) 适应包装件配送车辆路径问题 (VRP) 的求解, 并提高该算法的求解性能。**方法** 在标准 BA 算法的基础上提出混合蝙蝠算法 (HBA)。首先, 设计改进的蝙蝠算法 (IBA), 使其能够适用于包装件配送 VRP 问题的求解。其次, 引入混沌系统, 对 IBA 算法进行混沌初始化。然后, 设计裂变算子和变异算子。在 IBA 算法迭代前半段, 将蝙蝠种群中较差的一半蝙蝠重新混沌初始化, 以提高种群多样性。在 IBA 算法迭代后半段, 对陷入局部最优解的蝙蝠进行鲶鱼扰动。最后, 提出 HBA 算法并对企业实例进行仿真测试。**结果** HBA 算法求得的最优配送距离为 773.01 km, 相对于 GA 算法 (781.25 km) 和 IBA 算法 (786.04 km) 分别节约了 8.24 km 和 13.03 km。**结论** 与 IBA 算法和 GA 算法相比, HBA 算法求解包装件配送 VRP 问题的全局优化能力更强、收敛速度更快。

关键词: 包装件配送; 车辆路径问题; 蝙蝠算法

中图分类号: TP301 **文献标识码:** A **文章编号:** 1001-3563(2018)13-0105-06

DOI: 10.19554/j.cnki.1001-3563.2018.13.018

Application of Hybrid Bat Algorithm in Package Distribution

PENG Wei

(Chongqing City Management College, Chongqing 401331, China)

ABSTRACT: The work aims to enable the bat algorithm (BA) to be applied to solve the vehicle routing problem (VRP) in package distribution, and improve the solving performance of BA. Based on the standard BA, a hybrid BA (HBA) was proposed. Firstly, the improved BA (IBA) was designed to solve VRP in package distribution. Secondly, the chaotic system was introduced to initialize IBA. Then, the fission operator and mutation operator were designed. In order to improve the population diversity, the worse half bats in the group were reinitialized by chaotic system during the first half iteration. In the second half iteration, the catfish disturbance was carried out on the bats fallen into the local optimal solution. Finally, the HBA was put forward and simulation tests were carried out on an enterprise instance. The optimal distribution distance obtained by HBA was 773.01 km, which respectively saved 8.24 km and 13.03 km compared with the GA (781.25 km) and BA (786.04 km). Compared with IBA and GA, the HBA has stronger global optimization ability and faster convergence speed when used to solve the VRP in package distribution.

KEY WORDS: package distribution; vehicle routing problem; bat algorithm

包装件物流活动包括包装件的仓储、装载、配送、卸货等系列环节^[1]。包装件配送直接影响着包装件物流活动的成本和服务质量, 是包装件物流活动的重要组成部分, 因此, 国内外学者对包装件配送问题进行了大量研究, 其中的车辆路径问题 (VRP) 及求解算法更成为了学者们关注的重点对象。目前, 包装件配送 VRP 问题的求解算法非常多, 大致可以分为精确

算法和智能算法 2 类^[2]。精确算法通常指可以寻找到全局最优解的算法, 但其计算量会随着问题规模的增大而呈指数增加, 因此精确算法主要用于小规模 VRP 问题求解。智能算法是学者们研究的重点, 国内外学者先后提出了遗传算法 (GA)、粒子群算法、蚁群算法等求解大规模 VRP 问题。

蝙蝠算法 (BA) 由剑桥大学学者 Yang Xinshe

收稿日期: 2018-01-12

基金项目: 重庆市教育委员会 2017 年度科学技术研究项目 (1609155488)

作者简介: 彭维 (1984—), 女, 硕士, 重庆城市管理职业学院讲师, 主要研究方向为物流与供应链管理。

于2010年首次提出^[3-5],模拟了蝙蝠通过超声波收搜、捕食猎物的过程。由于具有模型简单、鲁棒性强、收敛性好、适应性强等优点,BA算法自提出以后,就受到了学者的广泛关注。GANDOMI A H等^[3]采用BA算法对带约束的函数问题进行优化,并通过标准函数和工程问题验证了算法可行性。HASANCEBI O等^[4]将BA算法应用到钢结构优化领域中,拓展了算法应用范围。KASHI S等^[5]则利用BA算法求解核反应堆堆芯的燃料布置模型,并通过实例测试了算法求解性能。截止目前,学者们对BA算法的研究主要集中在连续函数优化问题上,很少有文献采用BA算法对离散型问题特别是VRP问题进行求解。同时,与现有智能算法类似,BA算法也存在收敛速度较慢、容易陷入局部最优解等问题,必然会对BA算法求解VRP问题的性能造成影响。基于上述分析,文中提出了求解包装件配送VRP问题的混合蝙蝠算法(HBA)。

1 包装件配送VRP问题的描述与模型

包装件配送VRP问题可描述为^[6-9]:配送中心需要为客户配送一批包装件。客户需求量、位置及配送中心位置已知,且车辆载质量给定,要求在满足如下几个条件的情况下合理规划配送线路,使整体配送距离最短:所有客户需求均得以满足,且各客户只能由一辆车辆服务;各客户需求量均没有超过车辆最大载质量;配送车辆不能超载。

用 U 表示配送网络节点集合, $U=\{0,1\dots L\}$,其中0表示配送中心,其余节点表示客户; K 表示配送车辆集合, $K=\{1,2\dots m\}$, m 表示完成配送任务所需车辆数; Q 表示车辆最大载质量; d_{ij} 表示客户 i 到 j 的距离; q_i 表示客户 i 的需求量。

设定决策变量:

$$x_{ijk} = \begin{cases} 1, & \text{车辆}k\text{配送完客户}i\text{后前往客户}j\text{配送} \\ 0, & \text{否则} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{车辆}k\text{配送客户}i \\ 0, & \text{否则} \end{cases}$$

则包装件配送VRP问题的数学模型为:

$$\min Z = \sum_{k \in K} \sum_{i \in U} \sum_{j \in U} d_{ij} x_{ijk} \quad (1)$$

$$\text{条件: } \sum_{i \in U \setminus \{0\}} q_i y_{ik} \leq Q, \forall k \in K \quad (2)$$

$$\sum_{k \in K} \sum_{i \in U} x_{ijk} = 1, \forall j \in U \setminus \{0\} \quad (3)$$

$$\sum_{k \in K} \sum_{j \in U} x_{ijk} = 1, \forall i \in U \setminus \{0\} \quad (4)$$

$$\sum_{i \in U \setminus \{0\}} x_{i0k} = \sum_{j \in U \setminus \{0\}} x_{0jk} = 1, \forall k \in K \quad (5)$$

$$\sum_{k \in K} \sum_{i \in U \setminus \{0\}} x_{i0k} = \sum_{k \in K} \sum_{j \in U \setminus \{0\}} x_{0jk} = m \quad (6)$$

$$\sum_{k \in K} \sum_{i \in S} \sum_{j \in \bar{S}} x_{ijk} \geq 1, \forall S \subseteq U \setminus \{0\}, |S| \geq 2 \quad (7)$$

$$\sum_{j \in U} x_{ijk} = \sum_{j \in U} x_{jik}, \forall k \in K, \forall i \in U \setminus \{0\} \quad (8)$$

$$x_{ijk}(x_{ijk} - 1) = 0, \forall i, j \in U, \forall k \in K \quad (9)$$

$$y_{ik}(y_{ik} - 1) = 0, \forall i \in U, \forall k \in K \quad (10)$$

其中:式(1)表示配送距离最短的目标函数;式(2)表示配送车辆不能超载;式(3)和(4)表示客户只能由一辆车辆进行服务;式(5)和(6)表示配送中心约束;式(7)表示子回路约束;式(8)表示客户车辆流守恒约束;式(9)和(10)表示0-1变量约束。

2 混合蝙蝠算法在包装件配送VRP问题中的应用

2.1 基本蝙蝠算法

蝙蝠拥有高级回声定位能力,它们发出声音脉冲并聆听反弹回来的回声,然后利用双耳时间差及回声响度变化来建立周围环境的三维场景^[3]。通过这种能力,蝙蝠不仅能够迅速捕捉到猎物,还能够在黑暗中准确找到自己的栖息地。BA算法就是模拟蝙蝠回声定位过程而提出的一种智能启发式算法,主要包括搜索过程和寻优过程2部分。

2.1.1 BA算法的搜索过程

在搜索过程中,算法主要通过频率 f 来控制蝙蝠的移动步伐和范围,并对蝙蝠位置和速度进行更新。用 $V_i^t = (v_{i1}, v_{i2} \dots v_{iD})$ 和 $X_i^t = (x_{i1}, x_{i2} \dots x_{iD})$ 分别表示 D 维空间中蝙蝠 i 在时刻 t 的速度和位置,则该蝙蝠在时刻 $t+1$ 的速度和位置可更新为^[5]:

$$V_i^{t+1} = V_i^t + (X_i^t - X^*)f_i \quad (11)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (12)$$

式中: X^* 为当前位置最优的蝙蝠; $f_i = f_{\min} + (f_{\max} - f_{\min})\beta$ (β 为一个服从均匀分布的随机因子且 $\beta \in [0,1]$)。

2.1.2 BA算法的寻优过程

与其他智能启发式算法相似,BA算法可以对蝙蝠寻优行为进行设定^[5]。在此过程中,每只蝙蝠 X_{old} 可以在最优解附近随机游走,并通过式(13)更新位置向量。

$$X_{\text{new}} = X_{\text{old}} + \varepsilon A^t \quad (13)$$

式中: $A^t = \langle A_i^t \rangle$,为时刻 t 所有蝙蝠的平均脉冲响度, $\varepsilon \in [0,1]$ 为随机数。当蝙蝠 i 捕食到猎物时,脉冲响度降低,同时脉冲发射频率增加,具体更新公

式为:

$$A_i^{t+1} = \alpha A_i^t \quad (14)$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad (15)$$

式中: $\alpha \in [0, 1]$ 为脉冲响度衰减系数; $\gamma > 0$ 为脉冲发射频率增加系数, 且两者均为常量。在实际问题中, 通常取值为 $\alpha = \gamma = 0.9$ 。结合蝙蝠飞行生理机制可知, 对于任意 α 和 γ , 当 $t \rightarrow \infty$ 时有 $A_i^t \rightarrow 0$ 和 $r_i^t \rightarrow r_i^0$ 。

2.2 改进蝙蝠算法在包装件配送 VRP 问题中的应用

2.2.1 构造蝙蝠表达式

为最大限度压缩解空间范围, 且实现客户点与配送车辆的对应, 文中利用 $2L$ 维空间向量对应具有 L 个客户的包装件配送 VRP 问题^[10-12]。每个 $2L$ 维向量均由 L 维的 X_a 向量和 X_b 向量构成, 前者表服务客户的配送车辆, 后者表示配送车辆在对应路径中服务客户的次序。假设配送中心用 3 辆车向 7 个客户配送包装件, 某只蝙蝠的位置向量 X 为:

$$\begin{array}{l} \text{客户点:} \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \\ X_a \quad : \quad 1 \quad 2 \quad 2 \quad 3 \quad 1 \quad 3 \quad 2 \\ X_b \quad : \quad 2 \quad 1 \quad 2 \quad 2 \quad 1 \quad 1 \quad 3 \end{array}$$

则该蝙蝠表示的配送路径应该为:

车辆 1: 0→5→1→0

车辆 2: 0→2→3→7→0

车辆 3: 0→6→4→0

相应地, 每只蝙蝠的速度向量 V 也用 2 个 L 维向量 V_a 和 V_b 表示。

2.2.2 改进蝙蝠算法在包装件配送 VRP 问题中的实现步骤

1) 种群初始化。随机生成 N 个 (种群规模) $1 \sim m$ (车辆数) 之间的 L 维整数和 $1 \sim L$ (客户数) 之间的 L 维不同实数作为蝙蝠位置向量 $X = (X_a, X_b)$ 。随机生成 N 个 $-(m-1) \sim (m-1)$ 之间的 L 维整数和 $-(L-1) \sim (L-1)$ 之间的 L 维实数作为蝙蝠速度向量 $V = (V_a, V_b)$ 。

2) 目标函数值计算。利用 Eval 函数计算每只蝙蝠 $X(i)$ 对应的目标函数值 $Z(i)$, 并寻找当前种群最优蝙蝠 X^* 及对应目标函数值 gZ , $i=1, 2, \dots, N$ 。

3) 判断算法是否达到最大迭代次数。若达到, 则输出种群最优解, 算法结束; 否则, 转入步骤 4)。

4) 根据式 (11) 至 (12) 更新蝙蝠 i 的速度和位置。其中, V_a, V_b 向上取整, 且当 X, V 超出边界时按边界取值。

5) 生成随机数 rand1, 若 $\text{rand1} > r(i)$, 则根据式 (13) 对蝙蝠 i 进行随机扰动得到待定最新解 X_{new} , 转入步骤 6)。否则, 转入步骤 7)。

6) 生成随机数 rand2, 若 $\text{rand2} > A(i)$ 且 $Z(X_{\text{new}}) < Z(X_{\text{old}})$, 则接受新解 X_{new} 。否则, 拒绝 X_{new} 。

7) 根据式 (14) 和 (15) 更新 $A(i)$ 和 $r(i)$ 。

8) 排列蝙蝠并寻找种群最优解, 返回步骤 3)。

2.2.3 Eval 函数

利用 Eval 函数求取蝙蝠目标函数值的主要步骤如下所述。

1) 对算法进化中得到的 X_b 向量进行升幂排序, 并将其整数化处理。假设某一蝙蝠的位置向量为:

$$\begin{array}{l} \text{客户点:} \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \\ X_a \quad : \quad 1 \quad 2 \quad 2 \quad 2 \quad 1 \quad 3 \quad 2 \quad 2 \quad 3 \\ X_b \quad : \quad 3.1 \quad 2.2 \quad 4.3 \quad 3.1 \quad 1.4 \quad 5.4 \quad 4.2 \quad 3.3 \quad 6.4 \end{array}$$

则表示客户 2, 3, 4, 7, 8 均由车辆 2 进行配送服务, 这些客户点对应的 X_b 值分别为 (2.2 4.3 3.1 4.2 3.3), 且客户点 2 的 X_b 值最小 (2.2), 客户点 4 次之 (3.1), 而客户点 8, 7 和 3 最大 (分别为 3.3, 4.2, 4.3), 因此, 可以将这些客户点的 X_b 整数化处理为 (1 5 2 4 3)。以类似方式, 可以得到该蝙蝠整数化后的 X_b 向量为 (2 1 5 2 1 1 4 3 2)。

2) 根据式 (16) 计算个体目标函数值, 其中 C 表示非常大的正实数。

$$\min Z = \sum_{k \in K} \sum_{i \in I'} \sum_{j \in I'} d_{ij} x_{ijk} + C \sum_{k \in K} \max(\sum_{i \in I'} q_i y_{ik} - Q, 0) \quad (16)$$

2.3 混合蝙蝠算法在包装件配送 VRP 问题中的应用

文中对 IBA 算法进行仿真实验, 发现该算法虽然适应 VRP 问题的求解, 但容易陷入局部最优解, 且收敛速度较慢。对此, 设计了求解包装件配送 VRP 问题的 HBA 算法。

2.3.1 混沌机制的引入

混沌状态是一种具有随机性、遍历性和规律性的非线性现象, 广泛存在于社会和自然界中^[13]。在 IBA 算法中引入混沌初始化, 能够利用混沌状态的上述特点优化蝙蝠初始分布, 提高初始蝙蝠的遍历性和个体质量, 进而改善算法全局寻优能力和收敛速度。一般来说, 混沌状态由式 (17—18) 产生。

$$y_{n+1} = \mu y_n (1 - y_n) \quad (17)$$

$$n = 0, 1, 2, \dots; 0 \leq \mu \leq 4 \quad (18)$$

IBA 算法的混沌初始化包括 2 个步骤: 根据式 (17—18) 生成大量差异较大的点, 并用其构造蝙蝠的位置向量; 择优选出部分个体组成初始蝙蝠种群。

2.3.2 裂变和变异机制的引入

除提高初始种群质量外, 保持算法进化过程中的种群多样性也是避免 IBA 算法陷入局部最优解的重要方式^[4]。对此, 文中提出裂变算子和变异算子, 以提高 IBA 算法中蝙蝠种群的多样性。

裂变算子是指在算法迭代前半段, 将种群中较差的一半蝙蝠重新初始化, 进而扩大算法搜索空间, 摆脱局部最优点, 具体操作: 如果种群最优解连续多次

(具体数值可设定,由大量仿真实验可知,以 3—5 次为宜)未发生变化,则将所有蝙蝠按照适应度值大小排列,并对适应度值较差的一半蝙蝠进行重新混沌初始化。

变异算子是指在算法迭代后半段,为保证算法搜索精度和运行速度,对陷入局部极值的蝙蝠进行鲉鱼扰动^[14],帮助其跳出局部最优解区域,具体操作为:如果个体或种群最优解连续多次(同样以 3 至 5 次为宜)未发生变化,则根据式(19)对相应蝙蝠进行鲉鱼扰动:

$$V_i^{t+1} = c_1 \phi_i V_i^t + (X_i^t - c_2 \phi_2 X_*) f_i \quad (19)$$

式中: ϕ_1, ϕ_2 为 (0,1) 之间的随机数; c_1, c_2 为鲉鱼对个体最优解和种群最优解的冲撞强度; $c_1 \phi_1$ 和 $c_2 \phi_2$ 则称之为鲉鱼算子,具体定义为:

$$c_1 \phi_1 = \begin{cases} c_1 \phi_1, & \text{个体最优解连续多次未变化} \\ 1, & \text{否则} \end{cases} \quad (20)$$

$$c_2 \phi_2 = \begin{cases} c_2 \phi_2, & \text{种群最优解连续多次未变化} \\ 1, & \text{否则} \end{cases} \quad (21)$$

分析式(20—21)可知,如果个体最优解或种群最优解连续多次相同,则认定蝙蝠种群产生了聚集,鲉鱼算子就会对蝙蝠进行扰动。反之,鲉鱼算子取值为 1,对算法没有任何影响。

2.3.3 混合蝙蝠算法在包装件配送 VRP 问题中的实现步骤

1) 参数设置。设置候选蝙蝠规模 M 、初始种群规模 N 、最大迭代次数 $\underline{M}$;蝙蝠 i 的初始脉冲响度 A_i^0 ,脉冲发射频率 r_i^0 ,脉冲频率最小值 $f_{\min}$ 和最大值 $f_{\max}$,脉冲响度衰减系数 α 和脉冲发射频率增加系数 γ ;鲉鱼冲撞强度 c_1 和 c_2 ,最优解连续未变次数上限 T 。

2) 种群混沌初始化。随机生成一个 0~1 之间的 $2L$ 维变量 $X_1 = (X_{1a}, X_{1b})$,并根据式(17—18)得到其他 M 个向量 $X_1, X_1 \dots X_M$;向量 X_{1a} 乘以 m 向上取整, X_{1b} 乘以 L (确保 X_{1a} 为 0~ m 之间的整数,且 X_{1b} 为 1~ L 之间的实数);利用 Eval 函数计算所有蝙蝠目标函数值,择优选择 N 个作为初始蝙蝠种群;随机生成 N 个 $-(m-1) \sim (m-1)$ 之间的 L 维整数和 $-(L-1) \sim (L-1)$ 之间的 L 维实数作为蝙蝠速度向量 $V = (V_a, V_b)$ 。

3) 记录每只蝙蝠 $X(i)$ 对应的目标函数值 $Z(i)$,并寻找当前种群最优蝙蝠 X^* 及对应的目标函数值 $gZ, i=1,2 \dots N$ 。

4) 判断算法是否达到 $\underline{M}$ 。若达到,则输出种群最优解,算法结束。否则,判断算法是否达到 $\underline{M}/2$ 。若达到,转入步骤 5)。否则,转入步骤 6)。

5) 根据式(19)和式(12)更新蝙蝠 i ,转入步骤 7)。其中, X_a, V_a 向上取整,且当 X, V 超出边界时

按边界取值。

6) 种群最优解是否连续 T 次未发生变化。若是,启动裂变算子,按照步骤 2) 将种群较差的一半蝙蝠重新初始化,另一半则根据式(11)和式(12)进行更新;否则,根据式(11)和式(12)更新蝙蝠式。同理, X_a, V_a 向上取整,且当 X, V 超出边界时按边界取值。

7) 生成随机数 rand1 ,若 $\text{rand1} > r(i)$,则根据式(13)对蝙蝠 i 进行随机扰动得到待选最新解 X_{new} ,转入步骤 8)。否则,转入步骤 9)。

8) 生成随机数 rand2 ,如果 $\text{rand2} > A(i)$,且 $Z(X_{\text{new}}) < Z(X_{\text{old}})$,接受新解 X_{new} 。否则,拒绝 X_{new} 。

9) 根据式(14)和式(15)更新 $A(i)$ 和 $r(i)$ 。

10) 排列蝙蝠并寻找种群最优解,返回步骤 4)。

3 仿真分析

为验证文中算法的有效性,采用包装件配送实例作为测试算例进行仿真实验。实例中,已知配送中心坐标为 (40, 45),需要向 30 个客户进行包装件配送服务,各客户需求量及位置信息见表 1,配送车辆最大载重为 8。

表 1 客户信息

Tab.1 Customer information

序号	[X, Y]	需求量	序号	[X, Y]	需求量
1	[16,83]	3.01	16	[23,58]	2.34
2	[73,63]	0.56	17	[60,28]	1.19
3	[26,100]	1.26	18	[18,75]	0.23
4	[81,69]	1.57	19	[77,31]	3.89
5	[12,36]	2.17	20	[45,59]	3.14
6	[44,36]	1.02	21	[11,25]	2.99
7	[25,10]	0.98	22	[17,37]	1.37
8	[19,40]	1.14	23	[40,75]	1.80
9	[23,88]	1.56	24	[52,4]	0.25
10	[18,87]	0.38	25	[89,51]	2.22
11	[64,18]	1.00	26	[26,37]	0.37
12	[45,98]	4.01	27	[34,40]	0.78
13	[12,55]	2.14	28	[89,23]	2.50
14	[52,53]	0.29	29	[44,57]	3.70
15	[74,61]	1.79	30	[71,39]	2.54

为合理安排配送线路,需要预先估计完成配送任务所需车辆数。文中采用式(22)确定参与配送任务的车辆数。

$$m = \left\lceil \sum_{i \in U} q_i / \partial Q \right\rceil + 1 \quad (22)$$

式中： $\lceil \cdot \rceil$ 为取不大于括号内数字的最大整数； $0 < \delta < 1$ ，是车容量的装载系数，与装车（或卸车）的复杂程度及约束有关。文中设置 $\delta = 0.95$ ，由此可得完成配送任务所需车辆数 $m=8$ 。

将 HBA 算法参数设置为：候选蝙蝠规模 $M=100$ ，初始种群规模 $N=80$ ，最大迭代次数 $M=60$ ，脉冲响应度衰减系数 $\alpha=0.9$ ，脉冲发射频率增加系数 $\gamma=0.9$ ；鲛鱼冲撞强度 $c_1=c_2=1.494$ ，最优解连续未变次数上限 $T=4$ 。在“Intel i5-8250U，8 GB RAM，Win XP，Matlab 2012”环境下，利用 HBA 算法对上述算例进行求解，求得最优配送距离为 773.01 km，具体配送方案见表 2。

表 2 最优配送方案
Tab.2 The optimal distribution scheme

序号	配送路径	配送 距离/km
1	0→9→10→1→18→16→0	103.18
2	0→28→19→0	107.69
3	0→27→8→22→5→13→0	80.25
4	0→29→20→0	29.75
5	0→25→4→2→15→14→0	119.13
6	0→23→12→3→0	129.40
7	0→26→21→7→24→11→17→6→0	140.46
8	0→30→0	63.15

由表 2 可知，HBA 算法对包装件配送 VRP 问题的求解是可行的，且求解效果较佳。为进一步验证文中算法求解性能，分别采用 HBA 算法、IBA 算法和 GA 算法^[15]对上述实例进行仿真实验，将求得最好解的收敛状况进行对比，结果见图 1。其中，IBA 算法

参数设置与 HBA 算法对应参数相同，GA 算法参数采用对应文献参数设置。

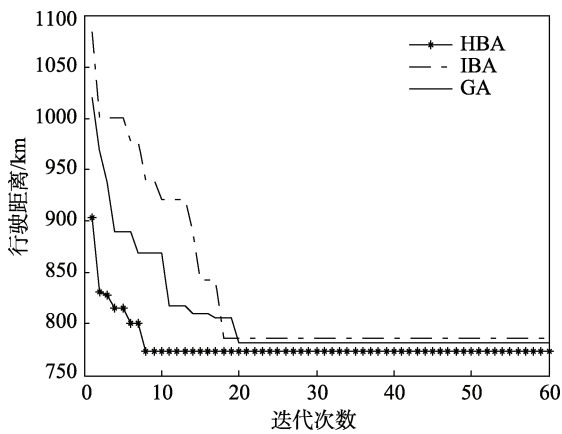


图 1 各算法收敛对比
Fig.1 Convergence comparison of each algorithm

在图 1 中，与 HBA 算法相比，IBA 算法和 GA 算法均陷入了局部最优解，且 IBA 算法求得解的质量最差，说明 HBA 算法的全局寻优能力最强，GA 算法次之，而 IBA 算法最弱。在收敛速度上，HBA 算法大概在 8 代左右就收敛到了最优解，而 IBA 算法和 GA 算法分别约在 18 代和 20 代开始收敛，说明收敛速度由快到慢依次为 HBA 算法、IBA 算法和 GA 算法。可见，HBA 算法求解性能优于 IBA 算法和 GA 算法。为验证算法的稳定性，采用上述 3 种算法分别对算例运行 20 次，统计求得最优解、最劣解、运行时间、搜索成功率等相关指标，结果见表 3。其中，搜索成功率指 20 次运算中算法求得已知最优解（根据上文仿真结果可知，该算例已知最优解为 773.01 km）次数的比例。

表 3 各算法仿真结果对比分析
Tab.3 Comparison and analysis of simulation results of each algorithm

算法	最优解/km	最劣解/km	平均值/km	计算时间/s	搜索成功率/%	平均收敛代数
IBA	786.04	1004.60	985.31	10.91	0	22.6
HBA	773.01	781.23	776.17	4.32	95	10.5
GA	781.25	1002.14	920.54	13.70	0	29.4

分析表 3 可知，在 20 次运算中，HBA 算法的搜索成功率最高，GA 算法和 IBA 算法次之，最劣值和平均值由小到大依次为 HBA 算法、GA 算法和 IBA 算法，表明 HBA 算法的全局寻优能力稳定优于 GA 算法和 IBA 算法。各算法的平均收敛代数和计算时间从小到大分别为 HBA 算法、IBA 算法和 GA 算法，可见 HBA 算法在增强全局寻优能力的同时，收敛速度并没有降低，依然稳定强于 IBA 算法和 GA 算法。综上分析，HBA 算法在全局寻优能力和收敛速度上均优于 IBA 算法和 GA 算法。

4 结语

文中主要对求解包装件配送 VRP 问题的蝙蝠算法进行了研究。针对标准 BA 算法不能适应 VRP 问题求解的缺陷，对 BA 算法进行改进，提出了 IBA 算法。然后，在 IBA 算法基础上，引入混沌、裂变和变异机制，设计了 HBA 算法。仿真结果表明，HBA 算法在全局寻优能力和收敛速度上均优于 IBA 算法和 GA 算法，为 VRP 问题的研究提供了一种新的思路。

参考文献:

- [1] 樊贵香. 混合模拟植物生长算法在包装件配送中的应用[J]. 包装工程, 2016, 37(13): 43—49.
FAN Gui-xiang. Application of Hybrid Plant Growth Simulation Algorithm in Packaged Products Distribution[J]. Packaging Engineering, 2016, 37(13): 43—49.
- [2] 裴小兵, 贾定芳. 基于模拟退火算法的城市物流多目标配送车辆路径优化研究[J]. 数学的实践与认识, 2016, 46(2): 105—113.
PEI Xiao-bing, JIA Ding-fang. Optimizing Multi-objective Vehicle Routing Problem in City Logistics Based on Simulated Annealing Algorithm[J]. Mathematics in Practice and Theory, 2016, 46(2): 105—113.
- [3] GANDOMI A H, YANG X S, ALAVI A H, et al. Bat Algorithm for Constrained Optimization Tasks[J]. Neural Computing and Applications, 2013, 22(6): 1239—1255.
- [4] HASANCEBI O, TEKE T, PEKCAN O. A Bat-inspired Algorithm for Structural Optimization[J]. Computers and Structures, 2013, 128(128): 77—90.
- [5] KASHI S, MINUCHEHR A, PORSALHI N, et al. Bat Algorithm for the Fuel Arrangement Optimization of Reactor Core[J]. Annals of Nuclear Energy, 2014, 64: 144—151.
- [6] LUO J B, WU Y. Modeling of Dual-cycle Strategy for Container Storage and Vehicle Scheduling Problems at Automated Container Terminals[J]. Transportation Research Part E: Logistics and Transportation Review, 2015, 79: 49—64.
- [7] MARINAKI M, MARINAKIS Y. A Glowworm Swarm Optimization Algorithm for the Vehicle Routing Problem with Stochastic Demands[J]. Expert Systems with Applications, 2016, 46: 145—163.
- [8] REN C Y. Tabu Search Algorithm for Capacitated Vehicle Routing Problem[J]. Advanced Materials Research, 2013, 753/754/755: 3060—3063.
- [9] GIANLUCA. Ant Colony Optimization for Capacitated Vehicle Routing Problem[J]. Journal of Computer Science, 2012, 8(6): 846—852.
- [10] AI T J, KACHITVICHYANUKUL V. Particle Swarm Optimization and Two Solution Representations for Solving the Capacitated Vehicle Routing Problem[J]. Computers & Industrial Engineering, 2009, 56(1): 380—387.
- [11] GOKSAL F P, ALTIPARMAK F, KARAOGLAN I. A Hybrid Particle Swarm Optimization for Vehicle Routing Problem with Simultaneous Pickup and Delivery[J]. Computers & Industrial Engineering, 2013, 65(1): 39—53.
- [12] BELMECHERI F, PRINS C, YALAOU F, et al. Particle Swarm Optimization Algorithm for a Vehicle Routing Problem with Heterogeneous Fleet, Mixed Backhauls, and Time Windows[J]. Journal of Intelligent Manufacturing, 2013, 24(4): 775—789.
- [13] 谈晓勇, 林鹰. 基于混沌蚁群算法的应急救援车辆调度优化[J]. 计算机应用研究, 2014, 31(9): 2640—2643.
TAN Xiao-yong, LIN Ying. Vehicle Scheduling Optimization Based on Chaos Ant Colony System Algorithm in Emergency Rescue[J]. Application Research of Computers, 2014, 31(9): 2640—2643.
- [14] 易文周, 田立伟. 一种基于混沌搜索和鲶鱼效应策略的粒子群算法[J]. 计算机应用与软件, 2013, 30(5): 311—315.
ZHOU Wen-yi, TIAN Li-wei. Particle Swarm Optimization Algorithm Based on Chaotic Search and Catfish Effect Strategy[J]. Computer Application and Software, 2013, 30(5): 311—315.
- [15] 周生伟, 蒋同海, 张荣辉. 改进遗传算法求解 VRP 问题[J]. 计算机仿真, 2013, 30(12): 140—143.
ZHOU Sheng-wei, JIANG Tong-hai, ZHANG Rong-hui. Improved Genetic Algorithm for VRP[J]. Computer Simulation, 2013, 30(12): 140—143.