DOI: 10.7641/CTA.2018.70421

泰森多边形的离散蝙蝠算法求解多车场车辆路径问题

戚远航[1], 蔡延光[1†], 蔡 颢[2], 黄何列[1], OLE Hejlesen[2]

(1. 广东工业大学 自动化学院, 广东 广州 510006; 2. 奥尔堡大学 健康科学与工程系, 丹麦 奥尔堡 9220)

摘要: 本文提出一种泰森多边形的离散蝙蝠算法求解多车场车辆路径问题(multi-depot vehicle routing problem, MDVRP). 所提出算法以离散蝙蝠算法为核心, 融入了一种基于多车场多车辆问题的编解码策略. 所提出算法还使用基于泰森多边形的初始化策略加快算法的前期收敛速度, 采用基于向量比较机制的适应度函数来控制算法收敛的方向, 引入基于近邻策略和优先配送策略的局部搜索算法来提高算法的寻优能力. 实验结果表明: 在合理的时间耗费内, 所提出的算法能有效地求解MDVRP, 尤其是带配送距离约束的MDVRP; 相对于对比算法, 所提出的算法表现出较强的寻优能力和稳定性.

关键词: 泰森多边形; 蝙蝠算法; 多车场车辆路径问题; 车辆路径

引用格式: 戚远航, 蔡延光, 蔡颢, 等. 泰森多边形的离散蝙蝠算法求解多车场车辆路径问题. 控制理论与应用, 2018, 35(8): 1142 – 1150

中图分类号: TP301 **文献标识码**: A

Voronoi diagram-based discrete bat algorithm for multi-depot vehicle routing problem

QI Yuan-hang[1], CAI Yan-guang[1†], CAI Hao[2], HUANG He-lie[1], OLE Hejlesen[2]

(1. School of Automation, Guangdong University of Technology, Guangzhou Guangdong 510006, China;
2. Department of Health Science and Technology, Aalborg University, Aalborg 9220, Denmark)

Abstract: This paper presents a voronoi diagram-based discrete bat algorithm to solve the multi-depot vehicle routing problem (MDVRP) which is a well-known NP-hard problem. The proposed algorithm takes the discrete bat algorithm as the core and integrates encoding and decoding strategies based on the multi-depot and multi-vehicle problem. The proposed algorithm also uses an initialized strategy based on the voronoi diagram to accelerate the previous convergent speed, and adopts a fitness function based on the vectorial comparison mechanism to control the convergent direction, as well as utilizing a local search algorithm based on the nearest neighbor strategy and the prior distribution strategy to enhance the optimization capability. Experimental results show that the proposed algorithm can effectively solve the MDVRP within a reasonable time consumption, especially the MDVRP with a delivery distance constraint; compared with contrast algorithms, the proposed algorithm has the stronger optimization ability and stability.

Key words: voronoi diagram; bat algorithm; multi-depot vehicle routing problem; vehicle routing

Citation: QI Yuanhang, CAI Yanguang, CAI Hao, et al. Voronoi diagram-based discrete bat algorithm for multi-depot vehicle routing problem. *Control Theory & Applications*, 2018, 35(8): 1142 – 1150

1 引言(Introduction)

多车场车辆路径问题(multi-depot vehicle routing problem, MDVRP)[1]是经典的组合优化问题之一, 其任务是在多个车场、多客户的情况下, 满足车辆数量、车辆容量、车辆配送距离、每个客户有且仅由一个车场的一辆车配送货物等约束条件, 设计适当的调度方

收稿日期: 2017−06−21; 录用日期: 2018−02−26.

†通信作者. E-mail: caiyg99@163.com; Tel.: +86 20-39322481.

本文责任编委: 贾英民.

国家自然科学基金项目(61074147), 广东省自然科学基金项目(S2011010005059), 广东省教育部产学研结合项目(2012B091000171, 2011B090400460), 广东省科技计划项目(2012B050600028, 2014B010118004, 2016A050502060), 广州市花都区科技计划项目(HD14ZD001), 广州市科技计划项目(201604016055)资助.

Supported by the National Natural Science Foundation of China (61074147), the Natural Science Foundation of Guangdong Province (S2011010005059), the Foundation of Enterprise-University-Research Institute Cooperation from Guangdong Province and Ministry of Education of China (2012B091000171, 2011B090400460), the Science and Technology Program of Guangdong Province (2012B050600028, 2014B010118004, 2016A050502060), the Science and Technology Program of Huadu District, Guangzhou (HD14ZD001) and the Science and Technology Program of Guangzhou (201604016055).

案来最小化配送成本. MDVRP是一个典型的NP难问题, 算法很难在有效的时间内得到最优解. 因此, 如何快速有效的求解该问题成为了学者们关注的焦点. 例如, 针对MDVRP的特点, 文献[2]以模拟退火算法为核心, 提出了一种有效的随机混合启发式算法; 文献[3]根据客户与车场的距离远近将MDVRP分成几个单车场子问题, 然后采用一种协同合作进化算法进行求解; 文献[4]总结了不同类型的遗传算法之间的差异, 使用不同类型遗传算法求解MDVRP并分析其求解效果; 文献[5]提出了一种混合颗粒禁忌搜索算法求解MDVRP, 该算法采用容量约束的位置–路径问题的启发式框架, 并引入多种策略进行初始化.

蝙蝠算法(bat algorithm, BA)是一种根据蝙蝠回声定位原理演变而成的元启发式算法[6]. 与其他元启发式算法相比, BA具有两个特点: 频率调谐, 可便于控制算法的寻优方向; 全局搜索和局部搜索的动态转换, 可有效避免算法过早收敛. 经过几年研究, 蝙蝠算法被应用到物流运输领域并体现出了其优异的寻优性能. 文献[7]使用蝙蝠算法进行多变量优化的社区房车路径调度, 在考虑速度、存储空间和旅行地点等因素的情况下来获得花费最少的旅行路径; 文献[8]提出了一种自适应的离散蝙蝠算法进行基于图的道路网络路径搜索优化; 文献[9]提出了改进的离散蝙蝠算法, 并应用其求解对称和非对称旅行商问题; 针对带容量约束的VRP问题, 文献[10]提出了一种基于路径重连的混合蝙蝠算法进行求解. MDVRP作为车辆路径问题的基本问题之一, 但目前还没有学者应用BA对其进行求解, 而现有的BA也不能直接求解MDVRP. 进一步地, 本文引入泰森多边形(又称Voronoi图)[11–13]来加强BA的寻优能力. 在物流运输调度领域, 学者们常使用泰森多边形实现智能算法的初始化、局部搜索策略的优化等功能并取得了较好的效果. 文献[11]利用泰森多边形减少无效的邻域搜索来优化禁忌搜索算法, 并用该算法求解带容量约束的VRP问题; 文献[12]提出了一种基于泰森多边形空间邻域搜索的启发式算法求解大型的带容量约束的VRP问题; 文献[13]使用泰森多边形进行路径的初始化, 以此加强离散粒子群算法的路径规划能力.

因此, 本文提出了一种泰森多边形的离散蝙蝠算法(voronoi diagram-based discrete bat algorithm, VDBA)求解MDVRP. VDBA结合了泰森多边形和离散蝙蝠算法, 提出了相应的编解码策略和适应度比较机制, 引入基于近邻策略和优先配送策略的局部搜索算法, 并通过仿真实验表明VDBA的有效性和稳定性.

2　模型描述(Model description)

设有N个客户, 客户编号为$1, 2, \cdots, N$; 每个客户的货物需求为$g_i(i = 1, 2, \cdots, N)$; M个车场, 车场编号为$N+1, N+2, \cdots, N+M$; 车场的车辆数为K_m ($m = N+1, N+2, \cdots, N+M$); 车辆的容量为$u$、最大行驶距离为$l$; 客户到客户、车场到客户之间的距离为$d_{ij}(i, j = 1, 2, \cdots, N+M)$. 因此, MDVRP的数学模型[3,14–15]为

$$\min \sum_{i=1}^{N+M}\sum_{j=1}^{N+M}\sum_{m=N+1}^{N+M}\sum_{k=1}^{K_m} d_{ij}x_{ij}^{mk}, \tag{1}$$

其中:

$$x_{ij}^{mk}=\begin{cases}1, & \text{车场}m\text{的车}k\text{从点}i\text{行驶到点}j,\\ 0, & \text{否则},\end{cases} \tag{2}$$

$$\begin{cases}\sum\limits_{j=1}^{N}\sum\limits_{k=1}^{K_m} x_{ij}^{mk} \leqslant K_m,\\ i, m \in \{N+1, N+2, \cdots, N+M\},\end{cases} \tag{3}$$

$$\begin{cases}\sum\limits_{j=1}^{N} x_{ij}^{mk} = \sum\limits_{j=1}^{N} x_{ji}^{mk} \leqslant 1,\\ i, m \in \{N+1, N+2, \cdots, N+M\},\\ k \in \{1, 2, \cdots, K_m\},\end{cases} \tag{4}$$

$$\begin{cases}\sum\limits_{j=1}^{N+M}\sum\limits_{m=N+1}^{N+M}\sum\limits_{k=1}^{K_m} x_{ij}^{mk} = 1,\\ i \in \{1, 2, \cdots, N\},\end{cases} \tag{5}$$

$$\begin{cases}\sum\limits_{i=1}^{N+M}\sum\limits_{m=N+1}^{N+M}\sum\limits_{k=1}^{K_m} x_{ij}^{mk} = 1,\\ j \in \{1, 2, \cdots, N\},\end{cases} \tag{6}$$

$$\begin{cases}\sum\limits_{i=1}^{N} g_i \sum\limits_{j=1}^{N+M} x_{ij}^{mk} \leqslant u,\\ m \in \{N+1, N+2, \cdots, N+M\},\\ k \in \{1, 2, \cdots, K_m\},\end{cases} \tag{7}$$

$$\begin{cases}\sum\limits_{i=1}^{N+M}\sum\limits_{j=1}^{N+M} x_{ij}^{mk} d_{ij} \leqslant l,\\ m \in \{N+1, N+2, \cdots, N+M\},\\ k \in \{1, 2, \cdots, K_m\},\end{cases} \tag{8}$$

$$\begin{cases}\sum\limits_{j=N+1}^{N+M} x_{ij}^{mk} = \sum\limits_{j=N+1}^{N+M} x_{ji}^{mk} = 0,\\ i, m \in \{N+1, N+2, \cdots, N+M\},\\ k \in \{1, 2, \cdots, K_m\},\end{cases} \tag{9}$$

$$\begin{cases}\sum\limits_{i\in S}\sum\limits_{j\in S} x_{ij}^{mk} \leqslant |S| - 1,\\ \forall S \subseteq \{1, 2, \cdots, N\},\\ 2 \leqslant |S| \leqslant N,\\ m \in \{N+1, N+2, \cdots, N+M\},\\ k \in \{1, 2, \cdots, K_m\}.\end{cases} \tag{10}$$

式(1)表示优化目标函数; 式(2)表示决策变量的取值约束; 式(3)表示车场的已使用车辆数不能超过该车场的车辆数; 式(4)表示车辆从车场出发, 配送完货物之后需回到原车场; 式(5)–(6)保证每个客户有且仅由一辆车辆配送; 式(7)定义了车辆容量约束; 式(8)定义了车辆配送距离约束; 式(9)表示车辆不能从车场到车

场; 式(10)排除子环约束.

3 泰森多边形的离散蝙蝠算法(Voronoi diagram-based discrete bat algorithm)

3.1 离散蝙蝠算法(Discrete bat algorithm)

3.1.1 蝙蝠位置与速度(Bat position and velocity)

蝙蝠i的位置定义:

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{iw}), \tag{11}$$

其中: 维度$w \in \mathbb{N}^+$, $i = 1, 2, \cdots, Q$, Q为种群规模.

蝙蝠i的速度定义:

$$v_i = \{v_{i1}, v_{i2}, \cdots, v_{iw}\}, \tag{12}$$

其中: $1 \leqslant v_{ij} \leqslant w, i = 1, 2, \cdots, Q, j = 1, 2, \cdots, w$.

3.1.2 蝙蝠位置、速度和频率的更新操作 (Updated operations for bat position, velocity and frequency)

设蝙蝠i的位置为$x_i = (x_{i1}, x_{i2}, \cdots, x_{iw})$, 速度为$v_i = \{v_{i1}, v_{i2}, \cdots, v_{iw}\}$, 频率为$f_i$; 全局最优的蝙蝠位置为$x_* = \{x_{*1}, x_{*2}, \cdots, x_{*w}\}$. 则蝙蝠位置、速度和频率的更新操作为

Step 1 $x_i - x_* = v_i^1 = \{v_{i1}^1, v_{i2}^1, \cdots, v_{iw}^1\}$:

$$v_{ij}^1 = \begin{cases} 0, & x_{ij} = x_{*j}, \\ x_{*j}, & x_{ij} \neq x_{*j}, \end{cases} \tag{13}$$

$$i = 1, 2, \cdots, Q,\ j = 1, 2, \cdots, w.$$

Step 2 $v_i^1 \times f_i = v_i^2 = \{v_{i1}^2, v_{i2}^2, \cdots, v_{iw}^2\}$:

$$v_{ij}^2 = \begin{cases} 0, & f_{\rm r} < f_i, \\ v_{ij}^1, & f_{\rm r} \geqslant f_i, \end{cases} \tag{14}$$

$$i = 1, 2, \cdots, Q,\ j = 1, 2, \cdots, w,$$

$$f_i^{\rm new} = \begin{cases} f_i, & f_{\rm r} < f_i, \\ f_i + \dfrac{f_{\rm r} - f_i}{\theta}, & f_{\rm r} \geqslant f_i, \end{cases} \tag{15}$$

$$i = 1, 2, \cdots, Q,\ j = 1, 2, \cdots, w,$$

其中: $f_{\rm r}$随机生成且$f_{\min} < f_{\rm r} < f_{\max}$, 频率影响因子$\theta > 1$.

Step 3 $v_i + v_i^2 = v_i^{\rm new} = \{v_{i1}^{\rm new}, v_{i2}^{\rm new}, \cdots, v_{iw}^{\rm new}\}$:

$$v_{ij}^{\rm new} = \begin{cases} v_{ij}, & {\rm rand}(\cdot) < 0.5, \\ v_{ij}^2, & {\rm rand}(\cdot) \geqslant 0.5, \end{cases} \tag{16}$$

$$i = 1, 2, \cdots, Q,\ j = 1, 2, \cdots, w.$$

Step 4 $x_i + v_i^{\rm new} = x_i^{\rm new}$:

令$x_i^{\rm new} = x_i$, 当$v_{ij}^{\rm new} \neq 0$, 交换$x_i^{\rm new}$中的第x_{ij}位置的分量和第$v_{ij}^{\rm new}$位置的分量, $i = 1, 2, \cdots, Q, j = 1, 2, \cdots, w$.

通过上述的步骤, 可得到蝙蝠新的位置、速度和频率, 分别为$x_i^{\rm new}$, $v_i^{\rm new}$, $f_i^{\rm new}$.

3.1.3 蝙蝠脉冲响度和发射频度的更新操作(Updated operations for bat loudness and pulse emission rate)

设蝙蝠i的初始发射频度为R_i^0, 在t代其脉冲响度为A_i^t, 则在$t+1$代蝙蝠的脉冲响和发射频度为

$$A_i^{t+1} = \alpha A_i^t,\ i = 1, 2, \cdots, Q, \tag{17}$$

$$R_i^{t+1} = R_i^0 \times [1 - \exp(-\gamma t)],\ i = 1, 2, \cdots, Q, \tag{18}$$

其中: $A_i^t \in [0, 1]$, $R_i^t \in [0, 1]$, 脉冲响度影响因子α和发射频度影响因子γ均为常量且$0 < \alpha < 1, \gamma > 0$.

3.1.4 蝙蝠变异的更新操作(Updated operation for bat variation)

设蝙蝠i的位置为$x_i = \{x_{i1}, x_{i2}, \cdots, x_{iw}\}$, 随机产生的不相等的两个整数$1 \leqslant j \leqslant w, 1 \leqslant k \leqslant w$. 具体步骤: 将$x_i$中第$j$个分量抽出来, 再插入到第$k$个分量的位置, x_i的其他分量做相应的移动.

3.2 基于多车场多车辆的编解码策略 (Encoding and decoding strategies based on the multi-depot and multi-vehicle problem)

针对多车场多车辆问题的特点, 结合第3.1节的离散蝙蝠算法, 提出了一种基于多车场多车辆的编解码策略. 在不改变所采用的算法的寻优结构、不增加算法空间复杂度的情况下, 实现了蝙蝠位置与对应车辆(车场)配送路径的转换.

3.2.1 编码策略(Encoding strategy)

设客户数为N, 车场数为M, 车场的车辆数为K_m, 总车辆数$W = \sum\limits_{m=N+1}^{N+M} K_m$. 令维度$w = N + W - 1$, 蝙蝠位置$x_i = (x_{i1}, x_{i2}, \cdots, x_{iw})$是$(1, 2, 3, \cdots, w)$的一次置换.

3.2.2 解码策略(Decoding strategy)

针对本文的编码策略, 结合MDVRP的相关约束和条件, 对应的解码策略如下:

Step 1 在蝙蝠位置x_i的分量x_{i1}前插入一个值为$w+1$的分量, 分量x_{iw}后插入一个值为$w+2$的分量, 以此得到序列$y_i = (y_{i1}, y_{i2}, \cdots, y_{i(w+2)})$. 其中, 任何$y_{ij} > N$均认为是一台车辆, $i = 1, 2, \cdots, Q, j = 1, 2, \cdots, w + 2$.

Step 2 从左往右搜索y_i的每个分量, 车辆y_{ip}到车辆y_{iq}之间经过的客户点构成车辆y_{ip}的配送路径(从y_{ip}出发并回到y_{iq}). 其中: $y_{ip} > N$, $y_{iq} > N$, $1 \leqslant p \leqslant w + 2$, $1 \leqslant q \leqslant w + 2$.

Step 3 在W个车辆配送路径中, 从左往右排序, 顺序为$[\sum\limits_{i=N+1}^{m-1} K_i, \sum\limits_{i=N+1}^{m} K_i]$的车辆是车场$m$的车辆.

例如, 设车场数为$M=2$, 客户数为$N=9$, 车辆数为$K_{10}=2$, $K_{11}=2$, 总车辆数为$W=4$, 由此可得, 维度$w=N+W-1=12$. 进一步地, 再设蝙蝠i位置为$x_i=(5,11,4,6,1,12,2,3,8,10,7,9)$, 可得到$y_i=(13,5,11,4,6,1,12,2,3,8,10,7,9,14)$. 由此可见, 车辆13的配送路径为(13, 5, 13), 车辆11的配送路径为(11, 4, 6, 1, 11), 车辆12的配送路径为(12, 2, 3, 8, 12), 车辆10的配送路径为(10, 7, 9, 10). 其中, 车辆13、车辆11是车场10的车辆, 车辆12、车辆10是车场11的车辆.

本文定义了$w=N+W-1$维的蝙蝠位置, 并通过$W-1$个$x_{ij}>N$的点来顺序划分W台车的归属, 确保了每个客户必将被配送, 同时也保证了每台车的使用概率相等. 解码后的W个车辆配送路径, 其只需要使用每台车对应的属性(坐标、最大载重量和最大配送距离)并可求其对应的适应度. 由此可见, 本文的编解码策略满足MDVRP的定义及其相关约束.

3.3 基于泰森多边形的初始化策略 (Initialized strategy based on voronoi diagram)

3.3.1 泰森多边形(Voronoi diagram)

泰森多边形描述[12]为: 设二维欧氏平面的离散点集合为$P=\{P_1,P_2,\cdots,P_i,\cdots,P_n\}$, 泰森多边形根据集合$P$的信息, 将空间分解为$n$的voronoi区域; 每个$P_i$对应一个voronoi区域, 且在该区域内的所有点到P_i的距离均小于这些点到集合P内其他点的距离. 其中, 每个P_i的voronoi区域$V(p_i)$的定义为

$$V(p_i)=\{x||p_i-x|\leqslant|p_j-x|,\ \forall j\neq i,\ 1\leqslant i\leqslant n,\ 1\leqslant j\leqslant n\}. \quad (19)$$

3.3.2 车场的优先配送区域(Voronoi diagram)

以M个车场为站点, N为客户数, $V(m)$为车场m $(m=N+1,\ N+2,\ \cdots,\ N+M)$的voronoi区域. $M=4$, $N=50$的泰森多边形如图1所示.

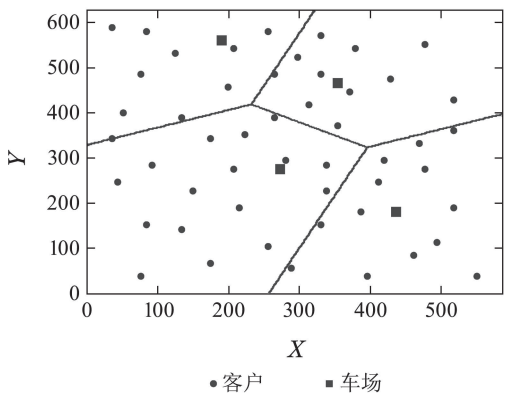

图 1 $M=4,\ N=50$的泰森多边形

Fig. 1 Voronoi diagram with $M=4,\ N=50$

从图1中可以看出: $V(m)$内的点到车场m的距离最近; 位于voronoi区域边上的点到其两边的车场的距离相等. 由此可见, 在生成车场的配送路径的过程中, 车场优先考虑其voronoi区域内的客户, 更迎合MDVRP的配送路径最短的目标. 因此, 本文定义$V(m)$是车场m的优先配送区域, $P(m)$是车场m的优先配送客户集合, 如式(20)所示:

$$P(m)=\{x|x\in V(m)\},\quad x\in\{1,2,\cdots,N\},\quad m\in\{N+1,N+2,\cdots,N+M\}. \quad (20)$$

3.3.3 初始化步骤(Initialization steps)

使用基于泰森多边形的初始化策略, 在初始化阶段生成特定的蝙蝠位置, 把各车场的优先配送客户优先分配给各车场的车辆配送, 以此加快算法的前期收敛速度.

设客户数为N, 客户编号为$1,2,\cdots,N$, 车场数为M, 车场编号为$N+1,N+2,\cdots,N+M$, 车场的车辆数为$K_m(m=N+1,N+2,\cdots,N+M)$, 总车辆数为$W=\sum_{m=N+1}^{N+M}K_m$.

Step 1　根据式(19)和(20)生成每个车场的优先配送客户集合$P(m)$.

Step 2　设$B_{\rm rand}$是$(N+1,N+2,\cdots,N+W)$的一种置换. 从左往右搜索$B_{\rm rand}$的分量, 顺序为

$$[\sum_{i=N+1}^{m-1}K_i,\ \sum_{i=N+1}^{m}K_i]$$

的分量是属于车场m的车辆编号, 设属于车场m的车辆编号集合为$D(m)$.

Step 3　设集合$C(m)=P(m)\cup D(m)$. 在确保第一个分量是车辆编号的情况下, 由$C(m)$随机生成一组包含$C(m)$所有元素且不重复的序列$C^m_{\rm rand}$.

Step 4　合并所有车场对应的$C^m_{\rm rand}$, 得到序列

$$C^{\rm ALL}=(C^{N+1}_{\rm rand},C^{N+2}_{\rm rand},\cdots,C^{N+M}_{\rm rand}).$$

Step 5　交换$C^{\rm ALL}$中值为$N+W$的分量和第1个分量的位置, 然后删除第1个分量, 得到X.

例如, 设车场数$M=2$, 客户数$N=9$, 车辆数为$K_{10}=2$, $K_{11}=2$, 总车辆数为$W=4$, 客户编号为$1,2,\cdots,9$, 车场编号为10, 11, 车辆编号为10, 11, 12, 13, 此外, $P(m=10)=\{1,3,5,7,9\}$, $P(m=11)=\{2,4,6,8\}$, $B_{\rm rand}=(10,13,11,12)$, 则$D(m=10)=\{10,13\}$, $D(m=11)=\{11,12\}$, 进一步地, 得到

$$C(m=10)=\{1,3,5,7,9,10,13\},$$

$$C(m=11)=\{2,4,6,8,11,12\}.$$

再设$C(m=10)$对应的$C^{10}_{\rm rand}=(10,3,1,9,5,13,7)$, $C(m=11)$对应的$C^{11}_{\rm rand}=(12,6,8,11,2,4)$, 则合并得$C^{\rm ALL}$后, 交换中第1个分量与值为13的分量的位置, 再删除第1个分量, 得到$X=(3,1,9,5,10,7,12,6,8,$

$11, 2, 4)$.

由此可见, X的维度是$N+W-1$, 同时X是$(1, 2, 3, \cdots, N+W-1)$的一种置换, 符合第3.2节蝙蝠位置的编码定义, 其对应着一个MDVRP路径, 其可以作为初始的蝙蝠位置.

3.4 基于向量比较机制的适应度函数(Fitness function based on the vectorial comparison mechanism)

通过编解码策略可以得到MDVRP的配送路径, 但无法证明该路径是一个可行解, 因为路径可能会违反式(7)–(8)的约束, 从而成为不可行解. 常用的方法[5]为采用惩罚机制处理式(7)–(8)后, 使用式(7)–(8)与式(1)的和定义适应度值, 然后比较适应度值, 适应度值越小, 路径越优, 反之路径越差. 但是, 式(1)(7)–(8)三者之间并没有存在定量的关系, 仅仅是把三者的和直接进行适应度比较不利于控制算法的寻优方向, 而且较优路径必须先满足式(7)–(8), 其次才是往式(1)方向收敛. 因此, 本文在使用惩罚机制的基础上加入了向量比较机制改进了式(1), 得到式(21):

$$\begin{aligned} &G_i = \{G_{i1}, G_{i2}, G_{i3}\} = \\ &\{\sum_{m=N+1}^{N+M}\sum_{k=1}^{K_m}\max\{0, \sum_{i=1}^{N}\sum_{j=1}^{N+M} x_{ij}^{mk} g_i - u\}, \\ &\sum_{m=N+1}^{N+M}\sum_{k=1}^{K_m}\max\{0, \sum_{i=1}^{N+M}\sum_{j=1}^{N+M} x_{ij}^{mk} d_{ij} - l\}, \\ &\sum_{i=1}^{N+M}\sum_{j=1}^{N+M}\sum_{m=N+1}^{N+M}\sum_{k=1}^{K_m} d_{ij} x_{ij}^{mk}\}, \end{aligned} \tag{21}$$

其中: 第1项是最大载重量约束, 第2项是最大配送距离约束, 第3项是原目标函数.

适应度比较: 设适应度1为$G_1 = \{G_{11}, G_{12}, G_{13}\}$, 适应度2为$G_2 = \{G_{21}, G_{22}, G_{23}\}$, 较优适应度为$G_b$. 令$j=1$; 如果$G_{1j} < G_{2j}$, 则$G_b = G_1$; 如果$G_{1j} > G_{2j}$, 则$G_b = G_2$; 如果$G_{1j} = G_{2j}$且$j \neq 3$, 则$j = j+1$, 进入下一个分量的比较; 如果所有分量均相等, 则$G_b = G_1$.

在初始阶段, VDBA优先选择满足最大载重量约束的路径, 其次是满足最大配送距离约束的路径; 随着迭代次数的增加, 当两个约束同时满足, 即G_{i1}和G_{i2}均变为0时, 适应度的比较将会重新变为式(1)的大小比较; 往后的迭代中, 全局最优解必定会满足式(7)–(8)的约束, 其将往式(1)的目标继续优化. 由此可见, 本文适应度定义及其比较方法能有效合理地控制算法的寻优方向, 加快算法的收敛速度.

3.5 局部搜索算法(Local search algorithm)

在求解VRP问题上, 常用的局部搜索算法主要有2–Opt搜索、0–1搜索和1–1搜索等算法[13, 17–18]. 根据MDVRP的特点, 本文提出近邻策略和优先配送策略, 并把两种策略融入到局部搜索中, 减少大量不必要的配送路径运算, 以此提高局部搜索的效率.

1) 近邻策略: 设$L(i, m)$为近邻标识, 如果客户i是车场m的最近的L个客户之一(L是近邻范围, $L < N$), 则i是车场m的近邻, $L(i, m) = \text{TRUE}$, 否则

$$L(i, m) = \text{FALSE}.$$

2) 优先配送策略: 设$T(i, m)$为优先配送标识, 如果客户 i 是车场 m 优先配送客户, 即$i \in P(m)$, 则$T(i, m) = \text{TRUE}$, 否则

$$T(i, m) = \text{FALSE}.$$

3) 2–Opt搜索算法[13]: 其是对某台车的配送路径分别进行2–Opt搜索, 直到该车辆的配送路径不能再改进为止. 示意图如图2所示.

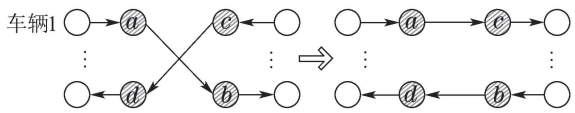

图 2 2–Opt搜索示意图

Fig. 2 2–Opt search schematic

4) 基于近邻策略的0–1搜索算法: 把一台车的某个客户插入到另一台车的配送路径中, 形成新的两条配送路径. 如果两个路径均满足近邻策略和最大载重约束, 则对两个新路径进行2–Opt搜索. 进一步, 如果2–Opt搜索后的两个路径均满足最大配送距离约束且路径有优化, 则接受新路径, 否则重新进行搜索直到达到最大搜索次数L. 示意图如图3所示.

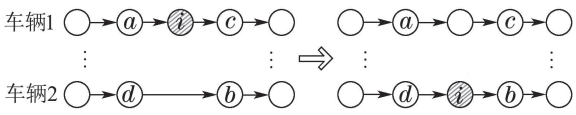

图 3 0–1搜索示意图

Fig. 3 0–1 search schematic

5) 基于优先配送策略的1–1搜索算法: 把一台车的某个客户和另一台车的一个客户进行交换, 形成新的两条配送路径. 如果两个路径均满足优先配送策略和最大载重约束, 则对两个新路径进行2–Opt搜索. 进一步, 如果2–Opt搜索后的两个路径均满足最大配送距离约束且路径有优化, 则接受新路径, 否则重新进行搜索直到达到最大搜索次数L. 示意图如图4所示.

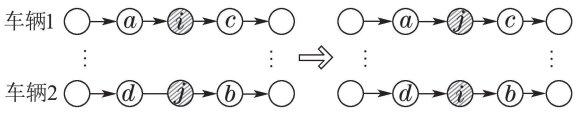

图 4 1–1搜索示意图

Fig. 4 1–1 search schematic

本文的局部搜索算法首先对所有车辆进行2–Opt搜索, 然后再依次进行基于近邻策略的0–1搜索和基于优先配送策略的1–1搜索.

3.6 算法步骤(Algorithm steps)

VDBA的算法步骤如下:

Step 1 初始化蝙蝠种群, 计算每个蝙蝠的适应度, 通过适应度比较得到全局最优的蝙蝠位置.

Step 2 应用蝙蝠速度、位置和频率的更新操作来更新蝙蝠的位置、速度和频率.

Step 3 如果rand(·)> R_i, 使用蝙蝠变异更新蝙蝠位置.

Step 4 应用局部搜索算法更新蝙蝠位置.

Step 5 如果蝙蝠位置的适应度更优且rand()< A_i, 接受当前的蝙蝠位置, 更新R_i和A_i.

Step 6 更新全局最优的蝙蝠位置.

Step 7 重复Step 2到Step 7直到达到最大迭代次数ND.

4 实验与分析(Experiments and analysis)

4.1 实验环境与算法参数设置(Experimental environment and parameter settings)

本文算法的相关实验均在同一实验环境中进行, 其中CPU主频为2.30 GHz, 内存为4 GB, 操作系统为32位Windows 7, 编程语言为C++. 根据BA相关的参数设置[6,9–10,16], 再经过实验得到VDBA较好的参数设置: $Q = 30$; $L = 2000$; $f \in [0, 1]$; $A \in [0, 1]$; $R \in [0, 0.9]$; $\theta = 2 \times w$; $\alpha = 0.999$; $\gamma = 0.001$.

4.2 实验结果与分析(Experimental results and analysis)

实验 1 使用16个MDVRP的基准算例[6,9–10,19]来验证VDBA的有效性, 实验结果如表1所示.

表 1 VDBA求解基准算例结果

Table 1 Results of VDBA for solving benchmark instances

No.(Group)	N	M	K_m	u	l	BKS	ND	L	$S_{\rm best}$	$S_{\rm avg}$	$G\_S_{\rm best}$	$G\_S_{\rm avg}$	T
1	50	4	4	80	∞	576.87	100	40	576.87	576.87	0.00	0.00	5.59
2	50	4	2	160	∞	473.53	100	40	473.53	473.87	0.00	0.07	8.37
3	75	5	3	140	∞	641.19	100	45	641.18	643.10	0.00	0.30	9.54
4	100	2	8	100	∞	1001.59	2500	80	1001.59	1004.46	0.00	0.29	143.78
5	100	2	5	200	∞	750.03	1000	80	750.03	752.26	0.00	0.30	210.23
6	100	3	6	100	∞	876.50	3000	60	876.50	879.80	0.00	0.38	156.57
7	100	4	4	100	∞	885.80	3000	60	881.97	889.09	−0.43	0.37	130.92
8(R1)	80	2	5	60	∞	1318.95	150	50	1318.95	1322.06	0.00	0.24	21.43
9(R1)	80	2	5	60	200	1318.95	150	50	1318.95	1318.95	0.00	0.00	22.32
10(R1)	80	2	5	60	180	1360.12	150	50	1360.11	1363.08	0.00	0.22	22.57
11(R2)	160	4	5	60	∞	2505.42	4000	60	2531.70	2551.08	1.05	1.82	290.36
12(R2)	160	4	5	60	200	2572.23	1000	60	2572.22	2579.97	0.00	0.30	112.53
13(R2)	160	4	5	60	180	2709.09	1000	60	2709.08	2752.38	0.00	1.60	113.63
14(R3)	240	6	5	60	∞	3702.85	4000	60	3822.85	3875.34	3.24	4.66	307.92
15(R3)	240	6	5	60	200	3827.06	4000	60	3828.60	3866.62	0.04	1.03	265.83
16(R3)	240	6	5	60	180	4058.07	4000	60	4113.92	4174.29	1.38	2.86	267.39

在表1中, 第1列No.(Group)表示序号(组序号)第6列∞表示车辆没有最大配送距离约束, 第7列BKS是已知最优解, 第10列$S_{\rm best}$是算法独立求解10次后得到的最优解, 第11列$S_{\rm avg}$是算法独立求解10次后得到的平均解, 第12列$G_S_{\rm best}$是最优解误差, 第13列$G_S_{\rm avg}$是平均解误差, 第14列T是平均时间耗费(单位: s). 其中, 最优解误差、平均解误差, 由式(22)–(23)计算可得

$$G\_S_{\rm best}(\%) = (S_{\rm best} - {\rm BKS})/{\rm BKS} \times 100, \quad (22)$$

$$G\_S_{\rm avg}(\%) = (S_{\rm avg} - {\rm BKS})/{\rm BKS} \times 100. \quad (23)$$

1) 在16个标准算例中, 12个算例能获得已知最优解, 所有算例的$G_S_{\rm best}$均不大于3.24%, $G_S_{\rm avg}$均不大于4.66%, T均小于308秒, 求解规模最大涉及240个客户点和6个车场(共30台车). 特别地, No.7算例的$G_S_{\rm best}$为−0.43%, 比BKS更好; No.1和No.9算例的$G_S_{\rm avg}$为0%, 即VDBA独立求解算例10次均能获得已知最优解.

2) 在R1, R2, R3这3组数据中, 每个组内的算例配送距离不同, 其他参数均相同. 可以看出, 相对于无配送距离约束的算例, VDBA求解带配送距离约束的算例的寻优效果更好, T相差不大甚至更少. 特别地, 在R3中, 求解No.14算例的$G_S_{\rm best}$为3.24%, $G_S_{\rm avg}$为4.66%, T为307.92 s, 但是求解No.15算例

的$G_S_{\rm best}$和$G_S_{\rm avg}$仅为0.04%, 1.03%, 而T却减少了42 s.

由此可见, 在合理的时间耗费内, VDBA能有效地求解MDVRP. 特别地, 相对于无配送距离约束的MDVRP, VDBA求解带配送距离约束的MDVRP的效果更好.

实验 2 智能算法是求解NP难问题的有效手段之一, 受到学者们的广泛关注, 因此本文引用遗传聚类算法(genetic clustering algorithm, GCA)[19]、并行改进蚁群算法(parallel improved ant colony optimization, PIACO)[20]、带可变聚类群的禁忌搜索算法(tabu search algorithm with variable cluster grouping, TSVCG)[14]与VDBA进行对比实验, 算法设置如表2所示, 实验结果如表3、图5和图6所示. 进一步地, 使用t–检验中的“成对双样本均值分析方法”来检验4个算法之间的算法性能是否存在显著性差异(设显著性水平为0.05, 平均差为0), 实验结果如表4–5所示.

表 2 GCA, PIACO, TSVCG的参数设置

Table 2 Parameter settings for GCA, PIACO and TSVCG

算法	参数设置
GCA [19]	种群数为50; 交叉概率为0.6; 变异概率为0.01
PIACO [20]	并行数为8; 种群数为30; 并行迁移更新间隔为10; 迁移数量为1
TSVCG [14]	迭代次数为40; 规模系数为1.3; 修正系数为0.7

表 3 VDBA与GCA, PIACO, TSVCG的对比实验结果

Table 3 Results of TLBAVNS, GCA, PIACO and TSVCG

No.	VDBA		GCA [19]		PIACO [20]		TSVCG [14]	
	$G\_S_{\rm best}$	$G\_S_{\rm avg}$	$G\_S_{\rm best}$	$G\_S_{\rm avg}$	$G\_S_{\rm best}$	$G\_S_{\rm avg}$	$G\_S_{\rm best}$	$G\_S_{\rm avg}$
1	0.00	0.00	2.58	—	0.00	0.29	2.35	3.13
2	0.00	0.07	−2.19	—	0.00	1.81	1.86	3.21
3	0.00	0.30	8.31	—	0.00	1.00	3.31	4.23
4	0.00	0.29	6.07	—	0.00	1.04	3.92	4.90
5	0.00	0.30	0.64	—	0.03	2.32	7.56	8.25
6	0.00	0.38	11.35	—	0.00	2.51	3.51	4.14
7	−0.43	0.37	10.24	—	−0.01	0.39	1.99	2.96
8	0.00	0.24	7.81	—	0.00	0.86	1.74	2.22
9	0.00	0.00	0.00	—	0.00	1.88	1.88	2.45
10	0.00	0.22	0.00	—	0.41	2.53	−1.34	−0.78
11	1.05	1.82	22.10	—	1.84	3.90	7.38	8.10
12	0.00	0.30	5.74	—	0.00	0.32	4.30	5.46
13	0.00	1.60	6.85	—	0.00	1.38	−1.10	0.16
14	3.24	4.66	47.53	—	2.11	10.24	9.93	10.84
15	0.04	1.03	3.39	—	0.00	4.97	4.84	6.90
16	1.38	2.86	7.07	—	0.96	8.32	−0.01	1.01

1) 除了No.2算例, VDBA求解其他算例的$G_S_{\rm best}$均明显优于GCA, 如图5所示. 特别地, 在No.14算例, GCA的$G_S_{\rm best}$为47.53%, 但VDBA的仅为3.24%, 降低了44.29%.

2) PIACO与VDBA的$G_S_{\rm best}$整体相差不大. 在$G_S_{\rm avg}$方面, 除了PIACO求解No.13算例时其$G_S_{\rm avg}$略优于VDBA之外, VDBA求解其他算例的$G_S_{\rm avg}$均优于PIACO. 再结合图5–6观察, PIACO与VDBA的$G_S_{\rm best}$是几乎重合的, 但是PIACO的$G_S_{\rm avg}$曲线却明显比VDBA的$G_S_{\rm avg}$曲线变化得更明显、振荡得更剧烈. 由此可见, VDBA比PIACO在求解MDVRP时表现出更强的稳定性.

3) TSVCG求解某些算例表现出较强的寻优能力, 如No.10算例、No.13算例和No.16算例, 但其求解大部分算例的$G_S_{\rm best}$大于1.74%, $G_S_{\rm avg}$大于2.22%. 而VDBA除了No.14算例和No.16算例之外, 求解其他算例的$G_S_{\rm best}$均不大于1.05%, $G_S_{\rm avg}$均不大于1.82%.

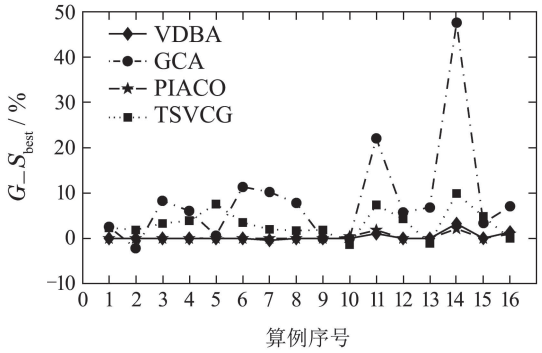

图 5 G_S_{best}的曲线图

Fig. 5 G_S_{best} curve

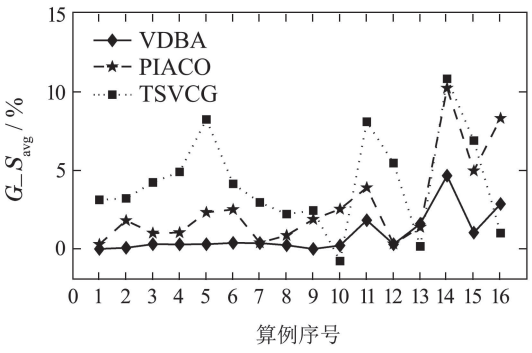

图 6 G_S_{avg}的曲线图

Fig. 6 G_S_{avg} curve

表 4 G_S_{best}的t–检验

Table 4 Student t-test for G_S_{best}

统计数值	VDBA与GCA的对比		VDBA与PIACO的对比		VDBA与TSVCG的对比	
	DBA	GCA	DBA	PIACO	DBA	TSVCG
平均	0.33	8.593125	0.33	0.33375	0.33	3.2575
方差	0.796573	141.3219	0.796573	0.476412	0.796573	9.56734
P(双尾)	0.009528		0.970728		0.000635	

表 5 G_S_{avg}的t–检验

Table 5 Student t-test for G_S_{avg}

统计数值	VDBA与PIACO的对比		VDBA与TSVCG的对比	
	DBA	PIACO	DBA	TSVCG
平均	0.9025	2.735	0.9025	4.19875
方差	1.63618	8.28436	1.63618	9.801185
P(双尾)	0.001049		0.000327527	

4) 在假设检验中, $P > 0.05$表示差异性不显著; $0.01 < P < 0.05$表示差异性显著; $P < 0.01$表示差异性极显著. 因此, VDBA的最优解相对于GCA和TSVCG均表现出极显著的差异性, 而平均解相对于TSVCG也表现出极显著的差异性; 相对于PIACO, 虽然VDBA的最优解的差异性不显著, 但是VDBA的平均解却表现出极显著的差异性.

因此, 虽然VDBA求解个别算例的结果不如对比算法, 但是整体而言, VDBA表现出了更强的寻优能力和稳定性, 与对比算法之间也存在着显著性的差异.

5　结论(Conclusions)

本文从初始化、编解码、适应度、局部搜索等4个方面入手, 设计了一种泰森多边形的离散蝙蝠算法求解MDVRP. 实验结果表明: 在合理的时间耗费内, VDBA求解所有算例的G_S_{best}均不大于3.24%, G_S_{avg}均不大于4.66%, T 均小于308 s; VDBA具有较强的寻优能力和稳定性.

参考文献(References):

[1] WREN A, HOLLIDAY A. Computer scheduling of vehicles from one or more depots to a number of delivery points [J]. *Journal of the Operational Research Society*, 1972, 23(3): 333 – 344.

[2] MIRABI M, FATEMI G S M T, JOLAI F. Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem [J]. *Robotics and Computer Integrated Manufacturing*, 2010, 26(6): 564 – 569.

[3] DE O F B, ENAYATIFAR R, SADAEI H J, et al. A cooperative coevolutionary algorithm for the multi-depot vehicle routing problem [J]. *Expert Systems with Applications*, 2016, 43(C): 117 – 130.

[4] KARAKATI, PODGORELEC V. A survey of genetic algorithms for solving multi depot vehicle routing problem [J]. *Applied Soft Computing*, 2015, 27(C): 519 – 532.

[5] ESCOBAR J W, LINFATI R, TOTH P, et al. A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem [J]. *Journal of Heuristics*, 2014, 20(5): 483 – 509.

[6] YANG X S. A new metaheuristic bat-inspired algorithm [M] *//Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Berlin: Springer, 2010: 65 –74.

[7] OCHOA A, MARGAIN L, ARREOLA J, et al. Improved solution based on bat algorithm to vehicle routing problem in a caravan range community [C] *//International Conference on Hybrid Intelligent Systems*. Gammarth: IEEE, 2013: 18 – 22.

[8] SUR C, SHUKLA A. Adaptive & discrete real bat algorithms for route search optimization of graph based road network [C] *//International Conference on Machine Intelligence and Research Advancement*. Katra: IEEE, 2013: 120 – 124.

[9] OSABA E, YANG X S, DIAZ F, et al. An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems [J]. *Engineering Applications of Artificial Intelligence*, 2016, 48(C): 59 – 71.

[10] ZHOU Y, LUO Q, XIE J, et al. A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem [M] *//Metaheuristics and Optimization in Civil Engineering*. Cham: Springer International Publishing, 2016: 255 – 276.

[11] KWON Y J, KIM J G, SEO J, et al. A tabu search algorithm using the voronoi diagram for the capacitated vehicle routing problem [C] //*International Conference Computational Science and ITS Applications*. Kuala Lampur: IEEE, 2007: 480 – 488.

[12] TONG H, CHAO W W, QIANG H C, et al. Path planning of UAV based on voronoi diagram and DPSO [J]. *Procedia Engineering*, 2012, 29: 4198 – 4203.

[13] FANG Z, TU W, LI Q, et al. A voronoi neighborhood-based search heuristic for distance/capacity constrained very large vehicle routing problems [J]. *International Journal of Geographical Information Science*, 2013, 27(4): 741 – 764.

[14] HE Y, MIAO W, XIE R, et al. A tabu search algorithm with variable cluster grouping for multi-depot vehicle routing problem [C] //*International Conference on Computer Supported Cooperative Work in Design*. Hsinchu: IEEE, 2014: 12 – 17.

[15] ZOU Tong, LI Ning, SUN Debao, et al. Genetic algorithm for multiple-depot vehicle routing problem [J]. *Computer Engineering and Applications*, 2004, 40(21): 82 – 83.
(邹彤, 李宁, 孙德宝, 等. 多车场车辆路径问题的遗传算法 [J]. 计算机工程与应用, 2004, 40(21): 82 – 83.)

[16] QI Yuanhang, CAI Yanguang, CAI Hao, et al. Chaotic hybrid discrete bat algorithm for traveling salesman problem [J]. *Acta Electronica Sinica*, 2016, 44(10): 2543 – 2547.
(戚远航, 蔡延光, 蔡颢, 等. 旅行商问题的混沌混合离散蝙蝠算法 [J]. 电子学报, 2016, 44(10): 2543 – 2547.)

[17] LI Yang, FAN Houming, ZHANG Xiaonan, et al. Two-phase variable neighborhood scatter search for the capacitated vehicle routing problem with stochastic demand [J]. *Control Theory & Applications*, 2017, 34(12): 1594 – 1604.
(李阳, 范厚明, 张晓楠, 等. 随机需求车辆路径问题及混合变邻域分散搜索算法求解 [J]. 控制理论与应用, 2017, 34(12): 1594 – 1604.)

[18] HIERMANN G, PUCHINGER J, ROPKE S, et al. The electric fleet size and mix vehicle routing problem with time windows and recharging stations [J]. *European Journal of Operational Research*, 2016, 252(3): 995 – 1018.

[19] THANGIAH S, SAIDSALHI. Genetic clustering: an adaptive heuristic for the multidepot vehicle routing problem [J]. *Applied Artificial Intelligence*, 2001, 15(4): 361 – 383.

[20] YU B, YANG Z Z, XIE J X. A parallel improved ant colony optimization for multi-depot vehicle routing problem [J]. *Journal of the Operational Research Society*, 2011, 62(1): 183 – 188.

作者简介:

戚远航 (1993–), 男, 博士研究生, 主要研究方向为供应链物流及智能算法, E-mail: qiyuanhang77@163.com;

蔡延光 (1963–), 男, 博士, 教授, 博士生导师, 主要研究方向为复杂网络系统建模、控制与优化、物流控制与优化、智能交通系统、组合优化、智能优化、物联网信息处理与优化控制, E-mail: caiyg99@163.com;

蔡 颢 (1987–), 男, 博士, 主要研究方向为大数据分析、智能信息处理, E-mail: howardbrutii@foxmail.com;

黄何列 (1990–), 男, 硕士研究生, 主要研究方向为数据挖掘、智能交通, E-mail: m15915854571@163.com;

OLE Hejlesen (1953–), 男, 博士, 教授, 博士生导师, 主要研究方向为远程医疗、决策支持系统, E-mail: okh@hst.aau.dk.