

一种异步航迹关联的变异蚁群算法

郭蕴华, 袁 成

(1. 武汉理工大学能源与动力工程学院, 湖北武汉 430063; 2. 船舶动力工程技术交通行业重点实验室, 湖北武汉 430063)

摘 要: 本文提出了一种异步多传感器航迹关联的变异蚁群算法. 该算法基于最优预测公式实现异步航迹的时间同步, 通过动态删除可访问节点索引以规避不可行解, 对最优关联结果进行变异操作以提高收敛速度和降低求解时间, 采用非均匀的初始信息素分布策略以减少无效分配. 仿真结果表明, 本文提出的算法计算代价较小, 并且具有较高的正确关联率.

关键词: 异步航迹关联; 多维分配; 蚁群算法; 变异

中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2012) 11-2200-06

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.11.010

A Mutation Ant Colony Algorithm for the Asynchronous Track Correlation

GUO Yun-hua, YUAN Cheng

(1. School of Energy and Power Engineering, Wuhan University of Technology, Wuhan, Hubei 430063, China;

2. Key Lab. of Marine Power Engineering and Technology Under Ministry of Communications of China,

Wuhan University of Technology, Wuhan, Hubei 430063, China)

Abstract: A mutation ant colony algorithm for the asynchronous multi-sensor track correlation is proposed. The time synchronization of the asynchronous tracks is implemented by optimal prediction. The infeasible solutions is avoided by dynamically deleting the index of the accessible nodes. The mutation operations for the optimal result is employed to elevate convergent speed and reduce the solving time. A non-uniformly distributional strategy of the initial pheromones is adopted to reduce the invalid assignments. The simulation results show that the computational cost of the algorithm is small and its correct percent of track correlation is high.

Key words: asynchronous track correlation; multi-dimension assignment; ant colony algorithm; mutation

1 引言

在分布式多传感器航迹融合系统中, 航迹关联是主要的核心问题之一^[1]. 特别是, 如果传感器节点数大于 2, 可以将多传感器航迹关联问题转化为特殊约束条件下的多维分配问题进行求解, 并已经证明这是一个 NP-Hard 问题^[2,3]. 对此问题, 已有的算法包括 Lagrange 松弛算法^[2,3]、基于灰关联的 Lagrange 松弛算法^[4]、平均场人工神经网络求解法^[5]等, 不过其求解过程仍然比较复杂. 对于异步多传感器的航迹关联问题, 有学者提出了基于最小二乘法进行时间同步并进一步通过 Viterbi 算法求解多维分配的方法^[6].

蚁群算法是由意大利学者 M. Dorigo 提出的一种新的群智能演化算法, 该算法具有全局搜索能力强的优点^[7], 并已经在数据关联^[8]、聚类分析^[9]、二次规划^[10]

等诸多领域得到广泛的应用. 不过, 亦有学者指出蚁群算法有收敛速度慢、求解代数多的缺点, 难以在实时性要求高的场合下应用^[11]. 本文提出了一种用于异步航迹关联的变异蚁群算法, 在每个航迹关联时刻均可以在 10 代之内获得满意解. 仿真结果表明, 本文提出的算法可以在较短时间内完成求解过程, 并且具有较高的正确关联率.

2 问题描述

假定系统内存在 $M (M > 2)$ 个传感器, 并且在公共区域内存在 N 个目标. 由于各传感器的采样频率和开机时间不同等因素, 无法获取恰好时间对准的航迹量测点. 假设需要进行多次航迹关联, 航迹关联时刻分别为 $t_1, t_2, \dots, t_k, \dots$, 则在航迹关联时刻 t_k 可以构造如下的加权局部统计量:

$$d_{i_{s-1}i_s}(t_k) = \frac{k-1}{k}d_{i_{s-1}i_s}(t_{k-1}) + \frac{1}{k}[\hat{X}_{i_{s-1}}(t_k) - \hat{X}_{i_s}(t_k)]^T [P_{i_{s-1}}(t_k) + P_{i_s}(t_k)]^{-1}[\hat{X}_{i_{s-1}}(t_k) - \hat{X}_{i_s}(t_k)]^T \quad (1)$$

其中, k 表示航迹关联的次数, $s = 1, 2, \dots, M$, 是传感器编号, $i_s = 1, 2, \dots, n_s$, 是第 s 个传感器的航迹编号, $\hat{X}_{i_s}(t_k)$ 分别为第 s 个传感器对第 i_s 航迹在航迹关联时刻 t_k 的同步处理后的状态估计矩阵, $P_{i_s}(t_k)$ 则是第 s 个传感器对第 i_s 航迹在航迹关联时刻 t_k 的同步处理后的误差估计协方差矩阵. 需要说明的是, 对于任意 i_s 航迹而言, 由于在航迹关联时刻 t_k 可能没有与之对应的采样点, 因此 $\hat{X}_{i_s}(t_k)$ 和 $P_{i_s}(t_k)$ 可以通过式(2)所描述的最优预测公式获得, 以实现异步航迹的同步化.

$$\begin{cases} \hat{X}_{i_s}(t_k) = F(t_k^r, t_k) \hat{X}_{i_s}(t_k^r) \\ P_{i_s}(t_k) = F(t_k^r, t_k) P_{i_s}(t_k^r) F^T(t_k^r, t_k) + \Gamma(t_k^r, t_k) Q(t_k^r) \Gamma^T(t_k^r, t_k) \end{cases} \quad (2)$$

式(2)中, t_k^r 为第 i_s 航迹在时间区间 $[t_{k-1}, t_k]$ 的实际采样时刻, 根据 kalman 滤波, 可以得到 t_k^r 时刻的状态估计矩阵 $\hat{X}_{i_s}(t_k^r)$ 和误差和误差估计协方差矩阵 $P_{i_s}(t_k^r)$. F 为状态转移矩阵, Q 为过程噪声的协方差矩阵.

基于式(1), 可以构造全局统计量:

$$D_{i_1 i_2 \dots i_M}(t_k) = \sum_{s=2}^M d_{i_{s-1} i_s}(t_k) \quad (3)$$

定义二进制变量

$$x_{i_1 i_2 \dots i_M} = \begin{cases} 1, & H_0 \\ 0, & H_1 \end{cases} \quad (4)$$

其中, $i_s = 1, 2, \dots, n_s; s = 1, 2, \dots, M$. 其中, H_0 表示 i_1, i_2, \dots, i_M 是同一目标, H_1 为对立假设, 表示 i_1, i_2, \dots, i_M 不是同一目标. 因此, 异步航迹关联问题可以化为多维分配问题, 即

$$\min \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_M=1}^{n_M} x_{i_1 i_2 \dots i_M} D_{i_1 i_2 \dots i_M}(t_k) \quad (5)$$

其约束条件为

$$\begin{cases} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \dots \sum_{i_M=1}^{n_M} x_{i_1 i_2 \dots i_M} = 1, & \forall i_1 = 1, 2, \dots, n_1 \\ \sum_{i_1=1}^{n_1} \sum_{i_3=1}^{n_3} \dots \sum_{i_M=1}^{n_M} x_{i_1 i_2 \dots i_M} = 1, & \forall i_2 = 1, 2, \dots, n_2 \\ \vdots \\ \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \dots \sum_{i_{M-1}=1}^{n_{M-1}} x_{i_1 i_2 \dots i_M} = 1, & \forall i_M = 1, 2, \dots, n_M \end{cases} \quad (6)$$

3 变异蚁群算法求解

蚁群算法作为一种群智能演化算法, 被认为是求

解 NP-Hard 问题的有效方法之一. 而式(5)~(6)所描述的多维分配问题就是一个 NP-Hard 问题, 可以尝试通过蚁群算法进行求解.

3.1 多维分配问题的蚁群算法

定义 1 对于 M 个传感器, 定义这些传感器航迹集合为 $U = [u_1, u_2, \dots, u_M]$; 并且, 进一步定义第 s 个传感器第 i 条航迹为 u_s^i . 例如, 传感器 1 的航迹集合为 u_1 , 其航迹 1 为 u_1^1 .

定义 2 公共区域内实际存在的 N 个实际目标, 定义与之所对应的 N 个航迹对集合为 $V = [v_1, v_2, \dots, v_N]$; 并且, 每个航迹对都可以表示为 $v_t = [u_1^1, u_2^3, \dots, u_M^i]$ 的形式, 表示为第 t 条实际航迹与传感器航迹 $u_1^1, u_2^3, \dots, u_M^i$ 相对应, 或者说传感器航迹 $u_1^1, u_2^3, \dots, u_M^i$ 指向同一目标 v_t .

基于上述定义, 并考虑式(6)中的约束条件所限定的可行解空间是一个凸集, 为此设计了一种特殊的蚂蚁寻径方法. 在该方法中, 通过动态删除可访问节点索引, 以确保蚂蚁所找的路径必须对应于可行解. 其过程描述如下:

Step 1 为第 l 只蚂蚁建立一个可访问的全局传感器索引表 $sList$, 其中 $sList$ 的中每个索引对应于一个传感器, 例如其元素 $sList[s]$ 指向 u_s ; 为第 l 只蚂蚁建立一个可访问的传感器航迹索引表 $tList$, $tList$ 是二维索引, 其中每个索引对应于一条传感器航迹, 例如其元素 $tList[s][i]$ 指向 u_s^i ;

Step 2 新建一个航迹对 v_t , 为 v_t 建立一个蚂蚁 l 可访问的当前传感器索引表 $sList_{-v_t}$ (直接将 $sList$ 所有元素复制到 $sList_{-v_t}$);

Step 3 蚂蚁 l 首先从 $sList_{-v_t}$ 随机选取的一个 u_s , 再从 $tList[s]$ 中随机选取某航迹 u_s^i 取出加入到 v_t , 然后从 $tList[s]$ 中删除 u_s^i 对应的索引, 并从 $sList_{-v_t}$ 中删除 u_s 对应的索引, 如果 $tList_{-v_t}[s]$ 为空, 则从 $sList$ 中删除 u_s 对应的索引;

Step 4 转向下一个节点: 蚂蚁 l 从 $sList_{-v_t}$ 随机选取的下一个 u_x (显然 $x \neq s$, 因为 $sList_{-v_t}$ 中 u_s 的索引已被删除), 再按式(7)所描述的节点转移策略从 $tList[x]$ 中取某航迹 u_x^j 加入 v_t , 然后从 $tList[x]$ 中删除 u_x^j 对应的索引, 并从 $sList_{-v_t}$ 中删除 u_x 对应的索引, 如果 $tList[x]$ 为空, 则从 $sList$ 中删除 u_x 对应的索引;

Step 5 如果 $sList_{-v_t}$ 为空, 则表明 v_t 中已经包含了来自所有传感器的航迹, 构成了完整的航迹对, 回到 **Step 2**, 以建立下一个航迹对;

Step 6 如果 $sList$ 为空, 则表明 U 中的所有航迹都已经转移到 V 中, 跳出循环.

$$\begin{cases} j_x = \operatorname{argmax}_{j_x \in \text{allowed}_l} [\tau_{ij_x}^\alpha(T/t_k) \eta_{ij_x}^\beta(t_k)], \text{ if } q \leq q_0 \\ P_{ij_x}^l(T/t_k) = \frac{\tau_{ij_x}^\alpha(T/t_k) \eta_{ij_x}^\beta(t_k)}{\sum_{r_x \in \text{allowed}_l} \tau_{ir_x}^\alpha(T/t_k) \eta_{ir_x}^\beta(t_k)}, \\ j_x \in \text{allowed}_l, \text{ otherwise} \end{cases} \quad (7)$$

式(7)中, $\tau_{ij_x}(T/t_k)$ 表示在航迹关联时刻为 t_k 的第 T 代的 u_s^i 与 u_x^j 之间的信息素强度, $\eta_{ij_x}(t_k)$ 为航迹关联时刻为 t_k 的启发式因子, 可以取 $\eta_{ij_x}(t_k) = 1/d_{ij_x}(t_k)$, $d_{ij_x}(t_k)$ 的计算方法见式(1), α 和 β 分别体现了信息素 $\tau_{ij_x}(T/t_k)$ 和启发式因子 $\eta_{ij_x}(t_k)$ 的重要程度. q 为随机数 ($0 < q < 1$), $q \leq q_0$ 时, 选取 $\tau_{ij_x}^\alpha(T/t_k) \eta_{ij_x}^\beta(t_k)$ 为最大的 u_x^j ; 当 $q > q_0$ 时, $P_{ij_x}^l(T/t_k)$ 为在航迹关联时刻为 t_k 的第 T 代的 u_s^i 与 u_x^j 之间的节点转移概率, 并按轮盘赌规则选取 u_x^j .

当所有蚂蚁完成一次寻径之后, 计算所有蚂蚁的找到的关联结果的目标函数值, 并将最优个体的关联结果保留下来, 然后开始下一代蚂蚁的寻径过程. 在种群演化过程中, 信息素会随着种群进化代数的增加而衰减, 若以 ρ 为衰减因子, 每代的信息素按照式(8)进行更新:

$$\begin{cases} \tau_{ij_x}(T+1/t_k) = \rho \cdot \tau_{ij_x}(T/t_k) + \Delta\tau_{ij_x} \\ \Delta\tau_{ij_x} = \sum_{l=1}^m \Delta\tau_{ij_x}^l \end{cases} \quad (8)$$

式(8)中, $\Delta\tau_{ij_x}^l$ 表示当前代第 l 只蚂蚁在城市 u_s^i 与 u_x^j 之间留下的信息素, m 为当前代在 u_s^i 与 u_x^j 之间经过的蚂蚁数目, $\Delta\tau_{ij_x}^l$ 则可以由式(9)表示:

$$\Delta\tau_{ij_x}^l = \begin{cases} Q/D_l, & \text{第 } l \text{ 只蚂蚁经过 } u_s^i \text{ 和 } u_x^j \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

式(9)中, Q 为一个常数, D_l 为第 l 只蚂蚁在当前代中获得的的目标函数值.

3.2 针对最优关联结果的变异策略

蚁群算法虽然具有全局搜索能力强的特点, 但是由于需要运行较多的代数才能获得较好的结果, 往往限制了其在实时性要求较高的场合下应用. 为了克服这一缺点, 受到遗传算法的启发, 本文提出一种针对最优关联结果的变异策略, 下面给出其具体描述.

假定某一关联时刻的最优航迹关联结果为 $V_{opt} = [v_1, v_2, \dots, v_N]$, 其中 v_t 表示 V_{opt} 中的第 t 个航迹对, 并以 v_i^i 表示 v_t 中的第 i 个航迹节点, 则变异操作描述为:

Step 1 建立可选择索引表 $vList$, $vList$ 中任何一个索引对应于一个航迹对.

Step 2 从 $vList$ 中随机选取第 t 个航迹对 v_t , 为 v_t 建立其邻域索引表 $vList - v_t$.

Step 3 从 v_t 的 $vList - v_t$ 中随机选取第 y 个航迹对 v_y , 并从 $vList - v_t$ 删除所 v_y 对应的索引.

Step 4 航迹对 v_t 与航迹对 v_y 执行如下的对位互换:

```
for (i = 1; i <= M; i++)
{
    for(j = 1; j <= M; j++)
    {
        if (v_t^i 与 v_y^j 属于同一传感器 && d(v_t^i, v_y^j) <
            d_max(t_k))
        {
            计算 D_old(v_t, v_y);
            v_t^i 与 v_y^j 交换, 计算 D_new(v_t, v_y);
            if (D_new(v_t, v_y) < D_old(v_t, v_y))
                保留交换后的结果;
            else
                v_t^i 与 v_y^j 还原;
        }
    }
}
```

上述伪代码中, $d(v_t^i, v_y^j)$ 代表 v_t^i 与 v_y^j 之间的统计距离, $D_{old}(v_t, v_y)$ 代表 v_t^i 与 v_y^j 交换之前的适应值, $D_{new}(v_t, v_y)$ 代表 v_t^i 与 v_y^j 交换之后的适应值. 由于互换的航迹节点必须属于同一传感器, 因此变异处理后的结果不会违反约束条件. 其主要思想就是: 将 v_t 与 v_y 中属于同一传感器且统计距离小于动态门限 $d_{max}(t_k)$ 的航迹节点进行交换, 如果交换后得到了更优的结果, 则保留之, 反之进行还原操作.

Step 5 如果 v_t 与 v_y 的互换操作未能得到更优结果, 回到 **Step 3**; 反之则按最新的航迹关联结果重新更新信息素矩阵, 然后从可选择列表 $vList$ 中删除 v_t 所对应的索引, 回到 **Step 2**.

Step 6 如果 $vList - v_t$ 为空, 则从可选择列表 $vList$ 中删除 v_t 所对应的索引, 然后回到 **Step 2**.

Step 7 如果 $vList$ 为空, 退出循环.

Step 8 将新得到的最优关联结果中的航迹对进行随机重新排序, 然后将上述过程重复 NC 次, NC 按式(10)给定:

$$NC = \operatorname{Ceil}(e^{-0.25(k-1)} NC_{ini}) \quad (10)$$

其中, NC_{ini} 为某一设定的初值, k 表示航迹关联的次数, Ceil 为取整函数. 式(10)表示, 随着航迹关联次数的增加, NC 逐步衰减.

3.3 初始信息素分布

在早期的蚁群算法中, 初始信息素矩阵的各元素设定为一个统一的初值. 由于信息素矩阵反映了节点

之间的关系远近程度,而客观上各节点之间的关系远近程度不可能是等量的,因此等量的初始信息素有可能导致蚁群算法的初期收敛速度较慢^[11].本文针对航迹关联多维分配这一特定问题,提出了一种新的初始信息素的分布策略,即

$$\begin{cases} \tau_{ij_x}(0/t_1) = 0, & d_{ij_x}(t_1) > d_{\max}(t_1) \\ \tau_{ij_x}(0/t_1) = \tau_{ini}, & d_{ij_x}(t_1) \leq d_{\max}(t_1) \end{cases} \quad (11)$$

其中, $\tau_{ij_x}(1/t_1)$ 表示 t_1 关联时刻航迹 u_s^i 与 u_x^j 之间的信息素初值, $d_{ij_x}(t_1)$ 为基于式(1)计算得到的统计距离, τ_{ini} 为某正数, $d_{\max}(t_1)$ 为某个合适的统计门限. 式(11)表示, 将那些统计距离大于 $d_{\max}(t_1)$ 的节点之间的初始信息素设为 0, 而将统计距离小于 $d_{\max}(t_1)$ 的节点之间的初始信息素设为某个正数. 根据节点转移策略式(7), 这种方法可以将导致如下的结果: 那些相互初始信息素为 0 的节点永远不会在任何一只蚂蚁的寻径过程中分配给同一个航迹对. 这种方法基于客观空情: 在给定的传感器量测误差条件下, 那些统计距离大于一定门限的航迹节点显然不可能属于同一真实航迹. 因此, 式(11)所描述的初始信息素分布策略, 有助于消除无效分配, 提高收敛速度.

此外, 信息素矩阵实际上记录了航迹与航迹之间的历史关系信息, 而这种历史关系信息以客观空情事实为基础, 不会随着航迹关联时刻的推演而发生改变. 因此, 当新的关联时刻 t_{k+1} 到来时, 其信息素矩阵直接从上一关联时刻 t_k 的最后一代的信息素矩阵复制, 而非重置信息素矩阵, 即

$$\tau_{ij_x}(0/t_{k+1}) = \tau_{ij_x}(T_{Last}/t_k) \quad (12)$$

4 仿真分析

4.1 仿真条件

用 Monte-carlo 方法对本文算法进行了仿真, 仿真次数为 50 次. 考虑 4 部异地三坐标雷达, 每部雷达的斜距测量方差 $\sigma_r = 50\text{m}$, 方位角测量方差 $\sigma_\beta = 0.5^\circ$, 高低角测量方差 $\sigma_\epsilon = 0.4^\circ$, 探测范围均为 32km. 4 部雷达的位置坐标分别为: $[2431, -3621, 20]$, $[-984, 2304, 33]$, $[-3138, -3357, 38]$ 和 $[2287, 1683, 10]$. 各雷达的采样间隔分别为 2s、2s、3s 和 2s, 且开机时间各不相同. 航迹关联的间隔为 2s. 仿真中蚁群算法的参数设置为: $Q = 10000.0$, $q_0 = 0.9$, $\alpha = 2.0$, $\beta = 6.0$, $\tau_{ini} = 100.0$, 蚂蚁数量为 10, 演化代数数为 10. 关联结果考虑 3 类概率, 即正确关联率 E_c , 错误关联率 E_e 以及漏关联率 E_s , 并且 $E_c + E_e + E_s = 1$. 全部算法采用 C 语言编写. 仿真采用的硬件配置为: CPU 为酷睿双核 T6570(主频 2.1G), 2G 内存、250G 硬盘, 未采用 CPU 超频技术. 计算了 3 个仿真算例.

算例 1 50 条间距为 250m 平行航迹: 其中第 1 条航迹的初始位置位于 4 雷达几何中心之外约 28km 左右, 且起点速度为 220m/s, 飞行方向大致朝向 4 部雷达的几个中心; 其余 49 条航迹按间距 250m 依次平行排列, 初速均为 220m/s. 50 个目标同时出发, 且在飞行过程中保持航速和间距不变, 过程噪声为 $0.1\text{m}^2/\text{s}$. 航迹关联次数为 10 次. 算例 1 的目标航迹与雷达分布见图 1.

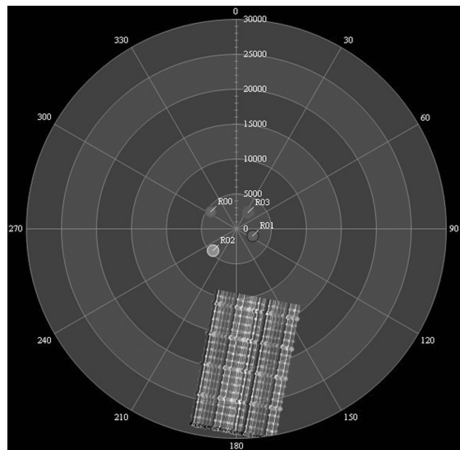


图1 算例1的目标航迹与雷达分布

算例 2 共 50 条航迹, 包括: 30 条机动航迹, 初始位置位于 4 雷达几何中心之外约 28km 左右, 初速在 250 ~ 450m/s 之间均匀分布, 初始飞行方向大致朝向 4 部雷达的几个中心, 在 15s ~ 30s 之间存在有意的机动; 10 条是平行航迹(仿真条件类似算例 1); 10 条是交叉航迹(交叉时间为第 30s), 交叉位置在 X、Y 坐标上完全重叠, 但在 Z 方向上相差 100m. 航迹关联次数为 20 次. 算例 2 的目标航迹与雷达分布见图 2.

算例 3 目标航迹条件与雷达分布同算例 2, 但改变雷达的测量方差, 以测试不同信噪比对本文算法的影响. 设定了 3 组不同的雷达测量方差参数: A ($\sigma_r = 50\text{m}$, $\sigma_\beta = 0.5^\circ$, $\sigma_\epsilon = 0.4^\circ$, 缺省参数); B ($\sigma_r = 60\text{m}$, $\sigma_\beta =$

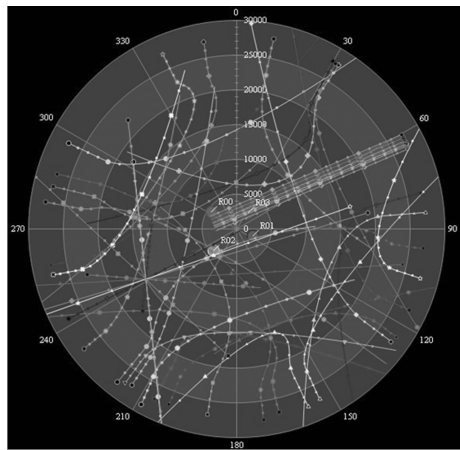


图2 算例2的目标航迹与雷达分布

$0.6^\circ, \sigma_\varepsilon = 0.5^\circ$); $C(\sigma_r = 70\text{m}, \sigma_\beta = 0.7^\circ, \sigma_\varepsilon = 0.6^\circ)$. 显然, 3组雷达条件下的信噪比是逐次降低的.

4.2 仿真结果

算例 1、2 的仿真结果见图 3~图 6, 算例 1、2 的全部关联时刻的平均计算耗时见表 1; 算例 3 的仿真结果见图 7~图 8, 其平均计算耗时与算例 2 基本相当. 在图 3~图 6 和表 1 中, 1 代表既采用了变异策略又采用了

初始信息素分布策略的算法, 2 代表仅采用了变异策略但未采用了初始信息素分布策略的算法, 3 代表未采用变异策略但采用了初始信息素分布策略的算法. 在图 7~图 8 中, A、B、C 分别代表 3 组不同的雷达测量方差参数, 且都为既采用了变异策略又采用了初始信息素分布策略的算法.

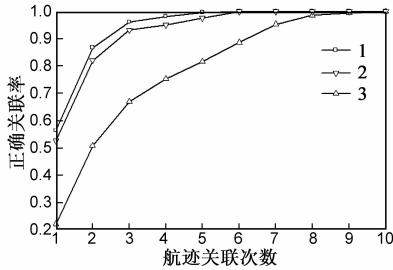


图3 算例1的正确关联率

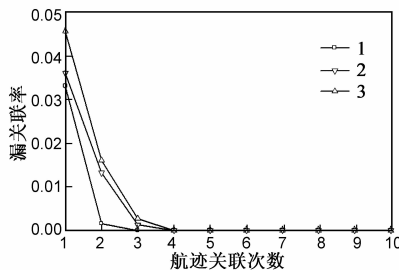


图4 算例1的漏关联率

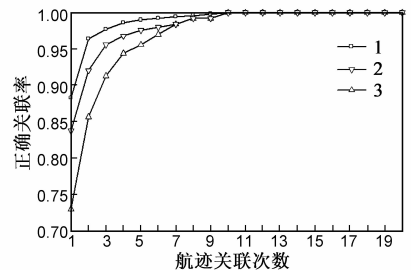


图5 算例2的正确关联率

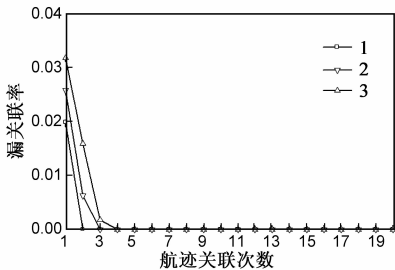


图6 算例2的漏关联率

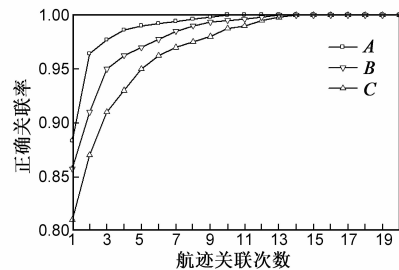


图7 算例3的正确关联率

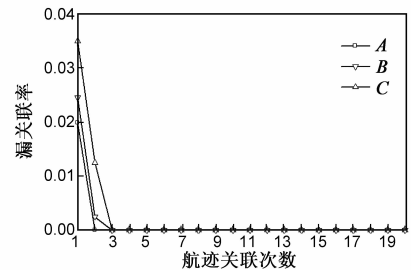


图8 算例3的漏关联率

表 1 算例 1、2 的平均计算耗时

	1	2	3
算例 1	31.6ms	31.6ms	25.3ms
算例 2	33.7ms	33.7ms	27.6ms

从仿真中可以看出:

(1) 算法 1 具有较高的正确关联率和较低的漏关联率, 并且收敛速度较快. 在算例 1、2 所给出的雷达测量方差条件下, 在 5 个步长左右其正确关联率可以达到 99% 以上, 在 3 个步长以内其漏关联率可以降低至 0%. 相比而言, 算法 2 与算法 1 的性能指标有一定差距, 算法 3 与算法 1 的性能指标差距较大. 这说明, 本文提出的针对最优关联结果的变异策略, 对提高算法的性能有显著的影响, 而初始信息素的分配策略也在一定程度上有利于提高算法的性能指标.

(2) 对于与本文算例同等计算规模的航迹关联问题而言, 本文算法所需的计算代价不大, 满足实际工程的实时性要求. 变异策略会导致计算代价的增加(增加量约为 6ms 多一点), 但仍在实际工程可以接受的范围内. 初始信息素的分配策略对计算代价没有影响.

(3) 算例 2、3 中, 在 30s (即图 5~图 8 的第 15 次~16 次关联前后) 发生了航迹交叉, 但并未导致本文算法

的性能指标发生波动. 由于存在信息素矩阵, 历史关系信息发挥了重要作用, 交叉的航迹不会因为偶尔某几个航迹点距离较近就发生错漏关联.

(4) 从算例 3 可以看出, 在同样的航迹条件下, 如果信噪比降低, 本文算法的性能 (特别是正确关联率) 会受到一定的影响: 虽然也能达到 99% 以上的正确关联率, 但所需的步长明显增加.

5 结论

传感器节点数大于 2 的航迹关联问题, 本质上是一个特殊约束条件下的多维分配问题. 该问题是一个 NP-Hard 问题, 目前在工程上仍然是一个难点. 而由于实际工程中的航迹关联问题一般都是异步航迹关联问题, 这就进一步增大了求解该问题的难度. 本文在异步航迹同步化的基础上, 实现了一种异步多传感器航迹关联的变异蚁群算法. 该算法采用了针对最优结果的变异策略和初始信息素的分布策略, 可以在较短时间内完成求解过程, 并且具有较高的正确关联率. 由于群智能演化算法被认为是求解 NP-Hard 问题的有效方法之一, 而群智能演化算法还包括粒子群算法、遗传算法等其它一些算法, 因此下一步可以尝试采用其它群智能

演化算法求解航迹关联多维分配问题,并进行不同算法的相互比较,以及开发几种算法的混合算法。

参考文献

- [1] Kaplan L M, Blair W D, and Bar-Shalom Y. Simulations studies of multisensor track association and fusion methods [A]. IEEE Aerospace Conference[C]. Big Sky, MT, 2006. 1 - 16
- [2] Chen Huimin, Y Bar-Shalom, T Kirubarajan. Tracking of spawning targets with multiple finite resolution sensors [J]. IEEE Transactions on Aerospace and Electronic Systems, 2008, 44(1): 2 - 14.
- [3] 衣晓, 何友, 关欣. 多目标跟踪的动态多维分配算法 [J]. 电子学报, 2005, 33(6): 1120 - 1123.
Yi xiao, He you, Guan xin. Dynamic multidimensional assignment algorithm of multitarget tracking [J]. Acta Electronica Sinica, 2005, 33(6): 1120 - 1123. (in Chinese)
- [4] 熊伟, 张晶炜, 何友. 基于多维分配和灰色理论的航迹关联算法 [J]. 电子与信息学报, 2010, 32(4): 898 - 901.
- [5] 田宝国, 何友, 杨日杰. 平均场网络在航迹关联中的应用 [J]. 航空学报, 2005, 26(1): 94 - 97.
- [6] 朱洪艳, 韩崇昭, 韩红, 等. 分布式多传感信息融合系统的异步航迹关联方法 [J]. 控制理论与应用, 2004, 21(3): 453 - 456, 462.
- [7] Blum C, Dorigo M. The hyper-cube framework for ant colony optimization [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2004, 34(2): 1161 - 1172.
- [8] 康莉, 谢维信, 黄敬雄. 一种基于蚁群算法的多目标跟踪数据关联方法 [J]. 电子学报. 2008, 36(3): 586 - 589.
Kang li, Xie weixin, Huang jingxiong. ACA based data association method for multi-target tracking [J]. Acta Electronica Sinica, 2008, 36(3): 586 - 589. (in Chinese)

- [9] Parag M Kanade, Lawrence O Hall. Fuzzy ants and clustering [J]. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 2007, 37(5): 758 - 769.
- [10] Stelian A, Srinivas K. Ant colony system application to macrocell overlap removal [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2004, 12(10): 1118 - 1123.
- [11] Konstantinidis K, Sirakoulis G C, Andreadis I. Design and implementation of a fuzzy-modified ant colony hardware structure for image retrieval [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 2009, 39(5): 520 - 533.

作者简介



郭蕴华(通讯作者) 男, 1975年6月出生, 四川省成都市. 武汉理工大学能源与动力工程学院副教授, 工学博士. 已发表论文十余篇, 其中被EI收录5篇. 获省部级一等奖1项、二等奖3项. 主要研究方向为信息融合、分布式系统仿真等.
E-mail: wtugyh@163.com



袁成 男, 1986年12月出生, 湖北省潜江人. 武汉理工大学能源与动力工程学院硕士研究生. 主要研究方向为故障预测与诊断、信息融合等.