

文章编号: 2095-2163(2021)02-0145-05

中图分类号: TP312

文献标志码: A

基于网格的改进序列鸡群算法

张航, 杨靖

(贵州大学 电气工程学院, 贵阳 550025)

摘要: 针对传统鸡群优化算法存在求解精度偏低、局部搜索能力弱等问题, 提出了一种改进的鸡群优化算法。改进算法选择利用动态簇解决单一工作节点能力有限问题, 提出一种基于网格的序列鸡群算法, 优化标准鸡群算法的种群分组更新机制, 仿真和实验结果表明该算法相比传统算法具有定位精度高、收敛速度快、实时性好等优点。

关键词: 无线传感器网络; 鸡群算法; 动态簇; 定位精度

Improved sequential chicken flock algorithm based on grid

ZHANG Hang, YANG Jing

(The Electrical Engineering College, Guizhou University, Guiyang 550025, China)

[Abstract] Aiming at the problems of low accuracy and weak local search ability in traditional chicken flock optimization algorithms, an improved chicken flock optimization algorithm is proposed. Improved algorithm selection uses dynamic clusters to solve the problem of limited capacity of a single working node. A grid-based sequential chicken flock algorithm is proposed to optimize the population grouping update mechanism of the standard chicken flock algorithm. Simulation and experimental results show that compared with traditional algorithms, this algorithm has the advantages of high positioning accuracy, fast convergence speed and good real-time performance.

[Key words] wireless sensor network; chicken swarm algorithm; dynamic cluster; positioning accuracy

0 引言

群智能算法作为一种新的优化技术,是从鸟类、鱼类等生物群的行为研究中获得的。这是一种受自然生物群智能现象启发的智能方法,并具有自组织、自学习、自适应和内部并行的特点。与传统数学方法相比,“适者生存”思想采用通用、智能的方法更容易解决问题。这些算法,如粒子群优化(PSO)^[1]、蚁群优化(ACO)^[2]、人工蜂群算法(ABC)^[3]、人工鱼群算法(AFSA)^[4]、蝙蝠算法(BA)^[5]等,使用工程、管理和计算科学等领域的优化又有了新的优化技术。文献[6]采用改进的人工鱼群算法对系统的硬件和软件进行了划分。文献[7]采用粒子群优化算法对热电目标进行动态跟踪。文献[8]模拟电路测试点优化通过动态蚁群算法实现。智能算法在不断更新,包括磷虾群优化算法^[9]、混沌果蝇算法^[10]、蜘蛛算法^[11]。这些都是基于大自然生物的生命周期表现的规律进行总结而得到的新兴算法。

2014年, Meng等人^[12]共同提出鸡群优化算法(CSO)来模拟鸡群的等级和行为。一只雄鸡和多只

母鸡组成一个亚组,多个亚组构成一个鸡群,各种鸡移动的规律是不同的。各个鸡群间在提前设置的规定下存在一定竞争。CSO优化算法是针对全局的,和其他优化算法如差分、蝙蝠等算法相比,在收敛准确性和速度上的优势都十分显著,然而传统鸡群算法在局部收敛和抗干扰方面还存在很多不足之处。

本文利用动态簇解决单一工作节点能力有限问题,提出一种基于网格的序列鸡群算法,优化标准鸡群算法的种群分组更新机制,同时还引入了新的小鸡学习公式。仿真和实验结果表明该算法具有定位精度高、收敛速度快、实时性好等优点。

1 鸡群算法

1.1 算法简介

2014年, Meng等人针对鸡寻觅食物的过程进行模仿,发现鸡群觅食的规律,并以此为基础推出CSO算法。关于鸡群觅食的规律,主要可概述为如下4点:

(1) 每个鸡群都由多个小组构成,领导者是雄鸡,组中还有小鸡和雌鸡,每个小组都有对应的领导者。

作者简介: 张航(1996-),男,硕士研究生,主要研究方向:物联网技术与应用;杨靖(1973-),男,博士,教授,主要研究方向:物联网技术与应用。

通讯作者: 张航 Email:1057360811@qq.com

收稿日期: 2020-11-19

(2)鸡群中雄鸡、雌鸡和小鸡的身份是由适应度作为标准进行区分的,雄鸡适应度最强,小鸡适应性最弱,母鸡分组原则实施随机分配。

(3)鸡群中雌鸡和雏鸡的母子关系保持,只是每隔一段时间这种关系就会按照特定的等级秩序和规则重新分配。

(4)雄鸡外出觅食时,雌鸡会跟着一同去觅食,而雏鸡会在雌鸡附近寻觅食物。具有较高适应度值的鸡在争夺食物的过程中将占有优势。

1.2 位置更新方法

鸡群中雄鸡、雌鸡和雏鸡身份等级的划分是按照适应度强弱实现的,实验后得知雏鸡和雄鸡占据比例都是20%,雌鸡占据比例高达60%,经过该算法进行优化,效果较为理想,雄鸡、雌鸡和雏鸡位置都会根据各自的位置更新算法完成更新。研究内容详述如下。

(1)雄鸡觅食范围最广,而且适应值最大,雄鸡觅食效率最高,下面是雄鸡位置更新算法:

$$x_{ij}(t+1) = x_{ij}(t) \times (1 + \text{randn}(0, \sigma^2)), \quad (1)$$

$$\sigma^2 = \begin{cases} 1, & f_i \leq f_l, \\ \exp\left(\frac{f_l - f_i}{|f_l| + \varepsilon}\right), & f_i > f_l. \end{cases} \quad l \in [1, NA], l \neq i. \quad (2)$$

其中, $x_{ij}(t)$ 表示第 i 只鸡原来位置; $x_{ij}(t+1)$ 表示更新后位置; $\text{randn}(0, \sigma^2)$ 高斯函数方差和期望值分别为 σ^2 和 0; f_i 、 f_l 分别表示雄鸡 i 、 l 的适应度大小,雄鸡中随机选择得到的 l , NA 表示雄鸡数量总值; ε 表示无穷小数。

(2)母鸡会跟着领导去搜索,和其余的鸡竞争抢食。领导者、雄鸡和雌鸡都会对位置更新产生一定影响,下面是位置更新算法:

$$x_{ij}(t+1) = x_{ij}(t) + s_1 \times \text{rand} \times (x_{r_1}(t) - x_{ij}(t)) + s_2 \times g \times (x_{r_2}(t) - x_{ij}(t)), \quad (3)$$

$$s_1 = \exp\left(\frac{f_i - f_{r_1}}{|f_i| + \varepsilon}\right), \quad (4)$$

$$s_2 = \exp(f_{r_2} - f_i), \quad (5)$$

其中, g 的值在 $0 \sim 1$ 之间随机选取, $g \in [0, 1]$; r_1 表示第 i 只雌鸡的鸡群中雄鸡; r_2 表示鸡群内全部雌鸡和雄鸡内任意一只鸡,而且满足 $r_1 \neq r_2$ 。 $s_1 = 0$,其他鸡觅食时第一只雌鸡会随着一起觅食。 $s_2 = 0$,雏鸡并不会跟随其他鸡群外出偷取或抢取食物,仅是在附近寻找食物。

$$\left\{ l_1^{(t+1)} \times \frac{A}{C} \sum_{i=L(1)}^{M(1)} p_i^{(t)}, \dots, l_j^{(t+1)} \times \sum_{i=L(j)}^{M(j)} p_i^{(t)}, \dots, l_{N^{(t+1)}}^{(t+1)} \times \frac{A}{C} \sum_{i=L(N^{(t+1)})}^{M(N^{(t+1)})} p_i^{(t)} \right\}. \quad (10)$$

(3)小鸡会跟随着母鸡寻找食物,其位置更新方法如下:

$$x_{ij}(t+1) = x_{ij}(t) + FL \times (x_{mr}(t) - x_{ij}(t)). \quad (6)$$

其中, $x_{mr}(t)$ 表示小鸡的母亲, FL 表示 $0 \sim 2$ 之间的随机数。

2 改进的鸡群算法

2.1 网格法的引入

传统鸡群算法中闭合表达式实现是很困难的,因此提出了网格法思想。网格法实现了函数离散化,使得算法的实现更加直观化。方法的具体内容如下。

(1)根据目标 $p(z^{(t+1)} | x^{(t)})$ 的分布情况,确定 $t+1$ 时刻可以移动到某一网格 $g_b^{(t+1)}$ ($b \in \{1, 2, \dots, N^{(t+1)}\}$) 的 t 时刻的区域 $G_a^{(t)}$, $G_a^{(t)}$ 定义为 $\{g_L^{(t)}, g_{L+1}^{(t)}, \dots, g_M^{(t)}\}$, 其中 $1 \leq L \leq M \leq N^{(t)}$ 。 $t+1$ 时刻如果目标移动到 $g_b^{(t+1)}$ 网格区域, t 时刻目标一定出现在 $G_a^{(t)}$ 区域 ($G_a^{(t)} \subseteq G^{(t)}$)。

(2)在 $g_b^{(t+1)}$ 范围内求 $\int p(x^{(t+1)} | x^{(t)}) p(x^{(t)} | z^{(t)}) dx^{(t)}$ 的值,也就是求 $G_a^{(t)}$ 范围内的积分:

$$\int_{G_a^{(t)}} p(x^{(t+1)} | x^{(t)}) p(x^{(t)} | z^{(t)}) dx^{(t)} = \sum_{i=L}^M \int_{g_i^{(t)}} p(x^{(t+1)} | x^{(t)}) p(x^{(t)} | z^{(t)}) dx^{(t)}, \quad (7)$$

其中, $G_a^{(t)}$ 网格序号最低和最高值分别为 L 、 M 。

(3)在每个网格区域 $g_i^{(t)}$ 内的 $p(x^{(t)} | z^{(t)})$ 为常数 $p_i^{(t)}$,以每个节点运行方法对公式(7)进行精简,得到公式(8):

$$\int_{G_a^{(t)}} p(x^{(t+1)} | x^{(t)}) p(x^{(t)} | z^{(t)}) dx^{(t)} = \frac{A}{C} \sum_{i=L}^M p_i^{(t)}, \quad (8)$$

(4) $g_b^{(t+1)}$ 区域内的信任度表达为:

$$p(x^{(t+1)} | z^{(t+1)}) = l_b^{(t+1)} \times \frac{A}{C} \sum_{i=L}^M p_i^{(t)}, \quad (9)$$

研究可知, $g_b^{(t+1)}$ 范围内近似函数值为 $l_b^{(t+1)}$ 。

(5) $G(t+1)$ 中的 $g_i^{(t+1)}$ ($0 < i \leq N^{(t+1)}$) 网格,重复运行前述各步的运算过程。利用公式(9)可将所有网格的 $p(x^{(t+1)} | z^{(t+1)})$ 计算出来, $G^{(t+1)}$ 的值如下:

集合中的每个元素 $M^{(j)}$ 和 $L^{(j)}$ 都和 $L_j^{(t+1)}$ 的序号 j 有关,并且结合中的元素不为 0。

2.2 仿真流程

基于网格,对目标轨迹进行侦测。首先,对环境及相关参数初始化。研究中假设模拟的观测区域是一个 $20 \times 20 \text{ m}^2$ 的正方形区域,分成了 20×20 个正方形网格,每个网格的区域的面积为 1 m^2 。在这个区域内随机分布了 60 个声波幅度传感器节点,假设跟踪目标在这个区域内的最大移动速度为 2 m/s ,假设声波的传播范围为米,设置传感器节点的门限值为 4,在侦测区域内初始目标轨迹。每次节点侦测完目标后将信息传递给下一个节点,循环直到目标超出侦测范围为止。仿真流程如图 1 所示。

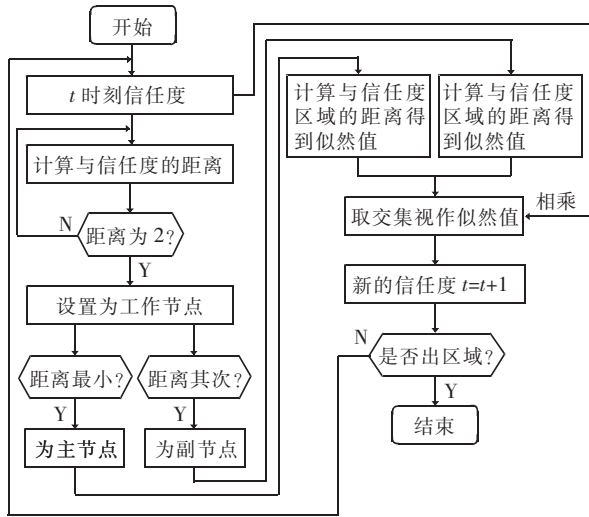


图1 仿真流程图

Fig. 1 Simulation flowchart

2.3 仿真结果

研究最终得到的仿真结果和结果分析可阐释分述如下。

(1) $t + 1$ 时刻的信任度 $G^{(t+1)}$ 分布区域,如图 2 所示。

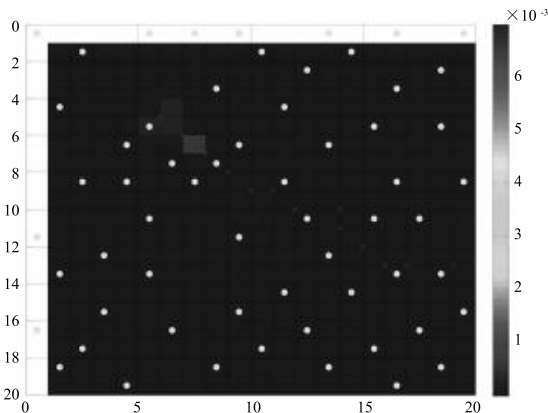


图2 $t + 1$ 时刻信任度分布区域

Fig. 2 Distribution area of trust at $t + 1$

(2) $t + 2$ 时刻的 $G^{(t+2)}$ 信任度分布区域,如图 3 所示。

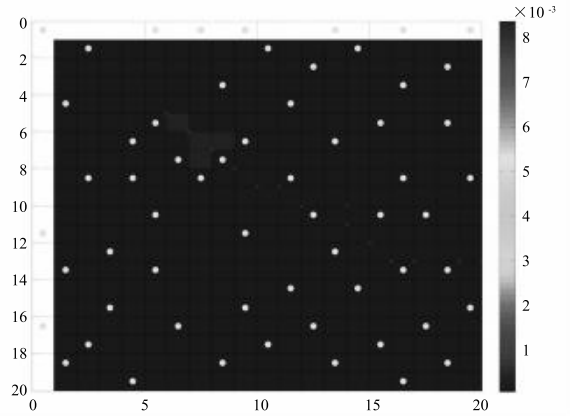


图3 $t + 2$ 时刻信任度分布

Fig. 3 Trust distribution at time $t + 2$

(3) 即将出侦测区域时的信任度区域,如图 4 所示。

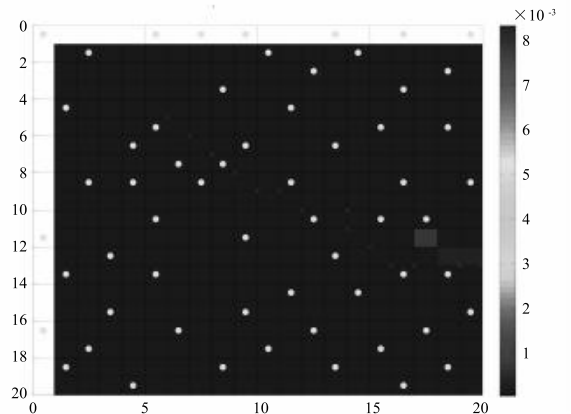


图4 即将出侦测区域是的信任度分布区域

Fig. 4 The trust distribution area that the detection area is about to come out

图 2 ~ 图 4 中,黄色圆点表示侦测区域内的传感器节点,白色小叉表示目标的实际运动轨迹。颜色的深浅比则表示了信任度的值,颜色越深就表示目标在该区域出现的可能性越高。

3 算法仿真

3.1 精确定位阶段

这里,研究拟给出定位步骤具体如下。

Step 1 100 个传感器节点(包括未知节点和信标节点)随机分布在立方体空间中,空间是边长为 50 m 的正方体。

Step 2 信标节点将信号强度数值传递给未知节点,未知节点将其转变成距离值。

Step 3 对第 n 个位置节点坐标进行计算,该

节点接收 m 个信号,如果 m 值是 0,未知节点不能捕获则跳转到 Step 7。如果 m 取值在 0~4 之间,坐标计算利用质心算法完成,然后跳转到 Step 7。如果 m 取值大于等于 4,那么继续 Step 4。

Step 4 将 m 个信标节点按每四个分成一组,组数为 k ,并且多个组并非出于相同平面。

Step 5 利用三面节点估计法计算出 k 个坐标值,即: $(x_1, y_1, z_1), \dots, (x_k, y_k, z_k)$ 。

Step 6 传统鸡群算法完成优化。

Step 7 如果未知节点数量为 n ,则算法终止;如果未知节点数量不为 n ,则 $n = n + 1$ 回转到 Step 3 继续执行算法。

3.2 仿真分析

节点随机分布在 $100 \text{ m} \times 100 \text{ m}$ 的平面,分别对比传统鸡群算法与改进鸡群算法在通讯半径、锚节点占比以及总节点数三个方面对定位精度的影响,依次参见图 5~图 7。

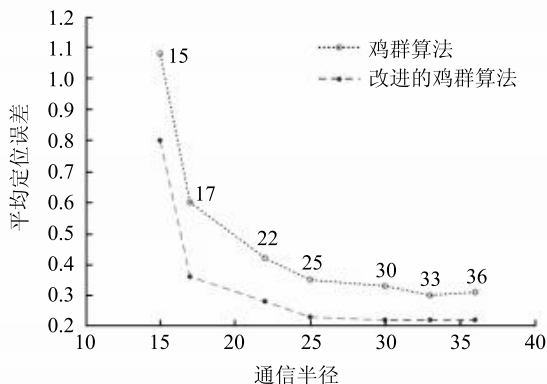


图 5 平均定位误差与通信半径之间的仿真曲线

Fig. 5 The simulation curve between average positioning error and communication radius

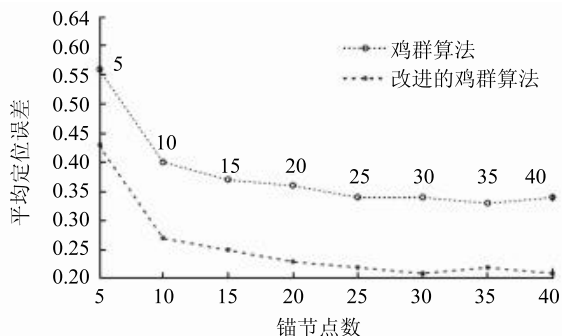


图 6 平均定位误差与锚节点数的仿真曲线

Fig. 6 The simulation curve between average positioning error and number of anchor nodes

由图 5~图 7 可以看出,在通信半径从 15 变化到 35 之间,改进的鸡群算法相较于传统鸡群算法定位精度提高了 15%;在锚节点的个数从 15 变化到

40 之间,改进鸡群算法的定位精度提高了 15% 左右;当总节点数目从 100 变化到 200 之间,改进鸡群算法的总体定位精度提高了约 14%。故可以得出,改进鸡群算法相较于传统鸡群算法在不同的通信半径、锚节点数、总节点数目等情况下,定位精度都有了明显改善。

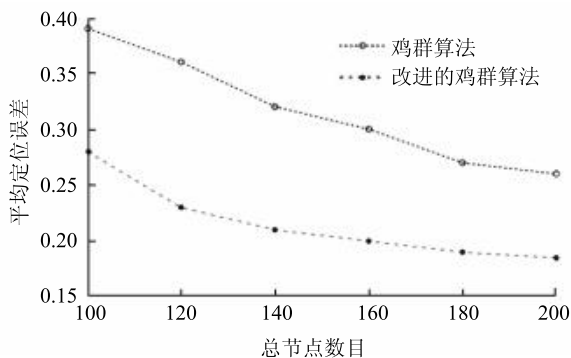


图 7 平均定位误差与总节点数目的仿真曲线

Fig. 7 The simulation curve between average positioning error and total number of nodes

4 结束语

文章以网格法和鸡群算法为基础提出的优化算法可以很好地解决传感器定位误差偏大的不足。该算法利用动态簇解决单一工作节点能力有限问题,优化了标准鸡群算法的种群分组更新机制,全面提高了定位精度。经过仿真验证文中提出的改进算法可以确保较好的收敛性,而且定位精度也有了明显改善,实现上也较为简单。不过密度高的障碍物环境并没有考虑进来,所以还需要在障碍物数量随意设定的状态下对算法做进一步的优化。

参考文献

- [1] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of IEEE International Conference on Neural Networks. Perth, Australia: IEEE Press, 1995: 1942-1948.
- [2] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on Systems Man and Cybernetics, Part B: Cybernetics, 1996, 26(1): 29-41.
- [3] Karaboga D. An idea based on honey bee swarm for numerical optimization [R]. Kayseri: Erciyes University, 2005.
- [4] LI Xiaolei, SHAO Zhijiang, QIAN Jixin. An optimizing method based on autonomous animats; fish-swarm algorithm [J]. Systems Engineering Theory and Practice, 2002, 22(11): 32-38.
- [5] YANG X S. A new metaheuristic bat-inspired algorithm [C]// Nature Inspired Cooperative Strategies for Optimization (NICSO 2010). Berlin: Springer, 2010, 284: 65-74.