

3-22-2023

Multiagent Following Multileader Algorithm Based on K-means Clustering

Guodong Yuan

Command And Control Engineering College, Army Engineering University of PLA, Nanjing 210001, China;
305551544@qq.com

Ming He

Command And Control Engineering College, Army Engineering University of PLA, Nanjing 210001, China;
paper_review@126.com

Ziyu Ma

Command And Control Engineering College, Army Engineering University of PLA, Nanjing 210001, China;

Weishi Zhang

Command And Control Engineering College, Army Engineering University of PLA, Nanjing 210001, China;

See next page for additional authors

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Multiagent Following Multileader Algorithm Based on K-means Clustering

Abstract

Abstract: Three K-means clustering algorithms are proposed to prevent chaos in the formation of a multi-agent system (MAS) with multiple leaders. *The algorithm divides the cluster into communities with the same number of leaders, and the agents within the community will follow the same leader.* Among the three proposed algorithms, algorithm #1 is suitable for scenarios with widely distributed agents wherein rapid consensus can be achieved in the shortest time; algorithm #2 is suitable for scenarios with a sparse agent distribution and effectively prevented agent collisions; and algorithm #3 exhibits rapid convergence and considerably reduces the MAS control cost, but will sacrifice the convergence speed of the system.. Unlike the traditional method in which the agents are numbered and the leader-follower relationship is fixed, the proposed clustering methods can shorten the MAS convergence time and effectively adapt to task changes by rapidly assigning new suitable tasks to agents.

Keywords

multi-agent, multiple leaders, K-means, pinning control, clustering methods

Authors

Guodong Yuan, Ming He, Ziyu Ma, Weishi Zhang, Xueda Liu, and Wei Li

Recommended Citation

Guodong Yuan, Ming He, Ziyu Ma, Weishi Zhang, Xueda Liu, Wei Li. Multiagent Following Multileader Algorithm Based on K-means Clustering[J]. Journal of System Simulation, 2023, 35(3): 616-622.

基于 K-means 聚类的多智能体跟随多领导者算法

袁国栋, 何明*, 马子玉, 张伟士, 刘学达, 李伟

(陆军工程大学 指挥控制工程学院, 江苏 南京 210001)

摘要: 为防止多智能体集群跟随多个领导者时编队混乱, 提出了 3 种 K-means 聚类算法, 将集群分成与领导者数量相同的社区, 社区内的智能体将跟随同一领导者。所提出的 3 种算法中, 算法 1 适用于智能体分布空间广的场景, 系统达到一致性所需时间最短; 算法 2 则适用于智能体分布稀疏的场景, 可有效避免智能体碰撞等危险; 算法 3 则大大降低多智能体集群的控制成本, 但将会牺牲系统的收敛速度。相较于传统预先对智能体编号, 领导-跟随关系固定的方法, 本文提出的分簇方法使系统收敛时间更短, 且有效应对中途任务变更的情况, 可快速分配给智能体新的合适任务。

关键词: 多智能体; 多领导者; K-means; 编队控制; 分簇方法

中图分类号: TP391.9 文献标志码: A 文章编号: 1004-731X(2023)03-0616-07

DOI: 10.16182/j.issn1004731x.joss.21-1158

引用格式: 袁国栋, 何明, 马子玉, 等. 基于 K-means 聚类的多智能体跟随多领导者算法[J]. 系统仿真学报, 2023, 35(3): 616-622.

Reference format: Yuan Guodong, He Ming, Ma Ziyu, et al. Multiagent Following Multileader Algorithm Based on K-means Clustering[J]. Journal of System Simulation, 2023, 35(3): 616-622.

Multiagent Following Multileader Algorithm Based on K-means Clustering

Yuan Guodong, He Ming*, Ma Ziyu, Zhang Weishi, Liu Xueda, Li Wei

(Command And Control Engineering College, Army Engineering University of PLA, Nanjing 210001, China)

Abstract: Three K-means clustering algorithms are proposed to prevent chaos in the formation of a multi-agent system (MAS) with multiple leaders. The algorithm divides the cluster into communities with the same number of leaders, and the agents within the community will follow the same leader. Among the three proposed algorithms, algorithm #1 is suitable for scenarios with widely distributed agents wherein rapid consensus can be achieved in the shortest time; algorithm #2 is suitable for scenarios with a sparse agent distribution and effectively prevented agent collisions; and algorithm #3 exhibits rapid convergence and considerably reduces the MAS control cost, but will sacrifice the convergence speed of the system.. Unlike the traditional method in which the agents are numbered and the leader-follower relationship is fixed, the proposed clustering methods can shorten the MAS convergence time and effectively adapt to task changes by rapidly assigning new suitable tasks to agents.

Keywords: multi-agent; multiple leaders; K-means; pinning control; clustering methods

收稿日期: 2021-11-11 修回日期: 2022-01-21

基金项目: 江苏省重点研发计划(BE2018754、BE2019762、BE2020729、BE2021729); 军内科研项目(LJ2021Z2010032 LJ2021C011129, LJ2020C02049); 军队重点课题(JYKYA2021029)

第一作者: 袁国栋(1992-), 男, 硕士生, 研究方向为无人机集群、弹性。E-mail: 305551544@qq.com

通讯作者: 何明(1978-), 男, 教授, 博士, 研究方向为无人化指挥控制、人工智能。E-mail: paper_review@126.com

0 引言

在自然界中群体行为是物种常见的生存方式, 如协同狩猎的狼群、“人”字迁徙的雁群、结队觅食的鱼群、分工协作的蜂群等等^[1-4]。近年来集群的协同控制研究, 特别是牵制控制激发了学者的广泛兴趣。与单个智能体独立工作相比, 多智能体协同工作将简单的局部规则进化到协调的全局行为, 将大大提高系统的运行效率、灵活性和容错能力。

基于智能体集群中是否有领导者和领导者的多少, 多智能体一致性可分为3类: 即无领导一致性^[5]、单领导的领导-跟随一致性^[6]和多领导的控制问题^[7]。在无领导者集群一致性研究中, 文献[8-9]研究了通信时延下的一致性, 文献[10-11]提出基于事件触发机制实现一致性。单领导者一致性, 即领导-跟随问题是多智能体领域主要研究方向。文献[12]提出虚拟领导者概念, 实现自由状态下和避障条件下的集群一致性。而后文献[13]等提出牵制控制算法, 证明集群内仅需若干个智能体接收到领导者运动信息便可实现集群一致性。多领导者的控制问题, 则是将整个多智能体系统拆分, 实现多个小集群的一致性。文献[14]等假设非同簇的智能体互不交换信息, 设计一种不依赖初始状态且收敛时间固定的算法。文献[15]发现集群规模越大, 引导全局运动的智能体数量所占比例则越小, 且比重越小准确度越高。文献[16]针对固定拓扑的特殊网络中领导者数量冗余的问题, 提出领导者最小数目算法。然而目前所做工作中, 无人机规模通常小于10个, 离数量超过100个的集群仍有较大差距, 领导者只引导若干个跟随者, 且领导-跟随关系预先确定, 未考虑到复杂网络的随机性, 距离实际应用与推广仍有很大距离。

社区挖掘算法是发现复杂系统模块结构特性的重要算法^[17], 在疫情防控、基因调控和预防网络犯罪等领域发挥着重要作用。传统社区挖掘算

法包含LPA^[18]和HANP^[19]等算法, 但这些算法不能准确预测网络中社区数量。而在多智能体系统中, 领导者的数量一般预先给定且智能体初始位置随机分布, 如果预先规定领导-跟随关系, 运动过程中易发生碰撞危险。或者领导者与跟随者相距较远, 直接影响系统达到一致性所需时间。

本文提出基于K-means聚类算法^[20]将集群分割为与领导者数量相同的社区, 社区内的智能体将跟随同一领导者。文章共设计3种算法以满足不同场景的需求。实验证明, 相较于传统预分配式跟随多个领导者的算法, 基于K-means算法更适合一般性拓扑结构的智能体网络。且集群中智能体数量越多, 本文所提出的算法效率越高, 系统的一致性越稳定。

1 基础理论

1.1 图论

将智能体看作是一个点, 存在通信的智能体间用直线连接, 则多智能体系统的拓扑结构可用图论知识表示: 图 $G(V, E, A)$, 点集合 $V=\{1, 2, \dots, n\}$ 代表智能体, 边集合 $E=\{(i, j)\}$ 代表智能体 i 和 j 之间存在通信。 $A=[a_{ij}] \in \mathbb{R}^n \times \mathbb{R}^n$ 代表邻接矩阵, 其中 $a_{ij} = \begin{cases} 1, & (i, j) \in E \\ 0, & \text{otherwise} \end{cases}$ 。在无向图 G 一个节点的度被定义

为其直接相连点的个数, 即 $d_i = \sum_{j=1}^n a_{ij}$, 图

的度矩阵则是对角线为各节点度的对角矩阵:

$$D = \text{diag}(d_1, d_2, \dots, d_n).$$

Laplacian矩阵被定义为

$$L = D - A \quad (1)$$

其谱含有关于图结构特性的信息, 最小非零特征值直接反映了图连通的鲁棒性。满足

$$z^T (L \otimes I_m) z = \frac{1}{2} \sum_{j \in N_i} a_{ij} (z_j - z_i)^2 \quad (2)$$

1.2 K-means算法

K-means作为数据挖掘中的一种无监督聚类

算法, 实现简单且效果优秀, 因而得到了广泛的应用。算法的核心思想是将给定的样本集, 按照样本之间的距离远近划分为 K 个社区。让社区内的点尽量紧密的连在一起, 而让社区间的距离尽量的大。在多智能体系统中, K-means 算法对如何划分集群使得智能体跟随不同的领导者具有很强的指导作用。设样本集包含 n 个样本, 位置信息为 $\mathbf{x}_i (i=1, 2, \dots, n)$, K-means 算法的基本流程为:

(1) 随机选取 k 个初始聚类中心

$$\mathbf{M}^0 = \{\boldsymbol{\mu}_1^0, \boldsymbol{\mu}_2^0, \dots, \boldsymbol{\mu}_k^0\} \quad (3)$$

(2) 样本 i 选择距离最近的聚类中心

$$C^t(i) \leftarrow \arg \min_j \|\mathbf{x}_i - \boldsymbol{\mu}_j^t\|^2, j = 1, 2, \dots, k \quad (4)$$

(3) 重新计算聚类中心

$$\boldsymbol{\mu}_j^{t+1} = \frac{\sum_{C^t(i)=j} \mathbf{x}_i}{\|C_j^t\|} \quad (5)$$

(4) 重复步骤(2)和(3), 直到聚类中心集合 \mathbf{M} 发生改变。

设势能函数 $F(\mathbf{C}, \mathbf{x})$

$$F(\mathbf{C}, \mathbf{x}) = \sum_{i=1}^n \|\boldsymbol{\mu}_{C(i)} - \mathbf{x}_i\|^2 \quad (6)$$

当聚类中心集合 \mathbf{M} 不变时, 势能函数 $F(\mathbf{C}, \mathbf{x})$ 达到最小值。但为了防止势能函数长时间不收敛, 有时会限制迭代次数或对势能函数设置阈值以降低计算复杂度。

2 基于 K-means 的多智能体分簇方法

假设由 N 个独立个体组成的多智能体系统, 需要跟随 k 位领导者。

算法 1: 基于 K-means 算法将集群分成 k 个社区并记录聚类中心位置, 领导者选择距离合适的聚类中心, 相应社区内的智能体将作为跟随者跟随该领导者。对于领导者如何选择距离合适的聚类中心问题, 本文设想两种方法: 距离极小极小值法和距离极小极大值法。

假设 k 个聚类中心位置为 $\mathbf{M} = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_k\}$, 领导者位置信息为 $\mathbf{p}_{\gamma_i}, i = 1, 2, \dots, k$.

距离极小极小值法流程为:

(1) 每位领导者选择距离最近的聚类中心并记录具体数值

$$d_i \leftarrow \arg \min_j \|\mathbf{p}_{\gamma_i} - \boldsymbol{\mu}_j\|^2 \quad (7)$$

(2) 设 $\{d_i\}_{\min} = d_k$, 则领导者 \mathbf{p}_{γ_k} 与式(9)中对应的 $\boldsymbol{\mu}_j$ 确定关系。

(3) 删除 \mathbf{p}_{γ_k} 和 $\boldsymbol{\mu}_j$, 剩余领导者和聚类中心重复步骤(1)和(2)。

而距离极小极大值法则是在步骤(2)中选择 $\{d_i\}_{\max} = d_k$ 的一对。

实验效果如图 1 所示, 红色代表聚类中心, 蓝色则是领导者位置。

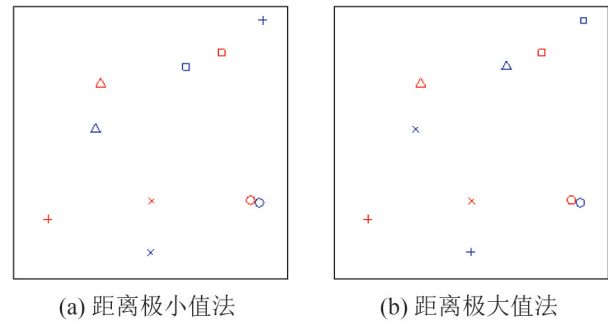


图 1 领导者和聚类中心之间的关系
Fig. 1 Relationship between leader and cluster center

如图 1(a) 中所示, 由于距离极小极小值法中领导者优先选用距离最近的聚类中心, 这导致后续的领导者无法选择理想的社区, 系统总体分配不合理, 可能存在某一对相距较大的问题。而距离极小极大值法虽牺牲个别领导者利益, 但整体分配更为合理且有效。所以本文将选用距离极大极小值法来帮助选择聚类中心。

算法 1 实现的分簇如图 2 所示, 其中符号 ‘☆’ 表示领导者。由图 2 可以看出, 虽然各个社区之间距离较大, 但因为分类过程与领导者无直接关联, 所以有时个别领导者出现在另一个社区的范围内。

算法 2: 直接将 k 位领导者作为初始聚类中心 $\mathbf{M}^0 = \{\mathbf{p}_{\gamma_1}, \mathbf{p}_{\gamma_2}, \dots, \mathbf{p}_{\gamma_k}\}$, 对 K-means 算法的迭代过程设置上限:

$$\|\boldsymbol{\mu}_i - \mathbf{p}_{\gamma_i}\|^2 < \|\boldsymbol{\mu}_i - \mathbf{p}_{\gamma_j}\|^2, i \neq j \quad (8)$$

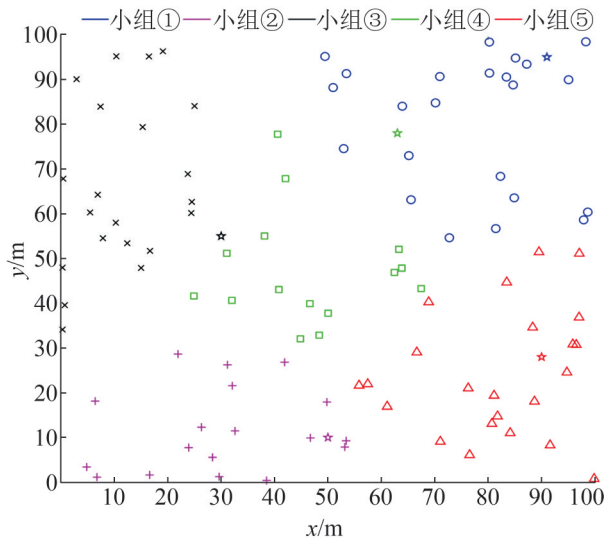


图2 算法1分簇结果
Fig. 2 Grouping result of algorithm 1

即领导者始终属于某一固定社区, 当该对应关系发生变化时, 聚类过程停止, 并保留上一次的聚类结果, 如图3所示。

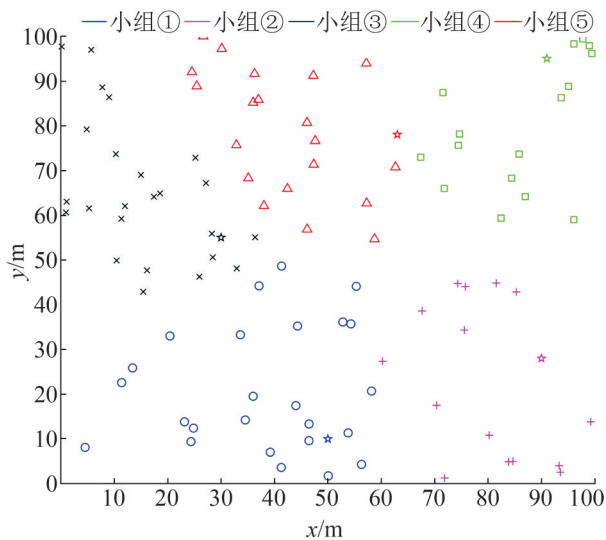


图3 算法2分簇结果
Fig. 3 Grouping result of algorithm 2

从图3可以看出, 相较于算法1, 算法2所实现的分簇使得社区与领导者之间联系的更为紧密, 避免智能体在运动过程中发生碰撞的危险。但社区内部的智能体之间间隔较大, 成员分布较为稀疏。

算法3: 将互为邻居的智能体间用直线连接, 可将多智能体网络分解为多幅子图。文献[13]提出

选择子图中某一关键节点作为牵制控制节点, 网络中只有牵制控制节点接收到领导者信息, 剩余节点则通过局部通信实现逐渐一致。智能体的控制输入 u_i 重写为

$$u_i = f_i^g + f_i^d + h_i \cdot f_i^y \quad (9)$$

式中: h_i 为牵制控制节点选择项

$$h_i = \begin{cases} 1, & i \text{ is a key - point} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

对牵制控制节点集合使用K-means聚类算法。对于如何选择关键节点问题, 文献[21]提出的影响度矩阵既考虑到节点本身的重要性, 也考虑到邻居节点的重要程度对它的影响。节点基于影响度矩阵的介数中心性越高, 其影响力越大。本实验基于影响度矩阵选择牵制控制节点, 并对牵制节点进行聚类分析, 结果如图4所示。

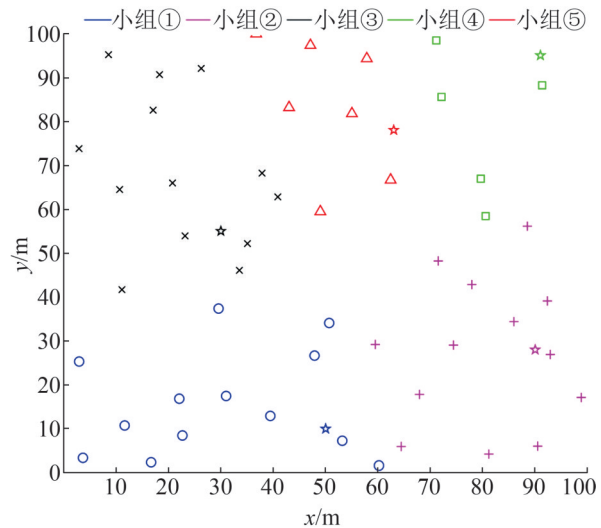


图4 算法3分簇结果
Fig. 4 Grouping result of algorithm 3

从图4可以看出, 需要聚类的智能体数减少至初始数量的一半。且智能体初始分布空间越小或智能体数量越多, 其网络中子图数量也就越少, 算法优势将越大。

3 仿真实验

3.1 实验1

设多智能体系统个体数量 $N=100$, 所有个体

的初始位置在 $100\text{ m} \times 100\text{ m}$ 的空间内随机分布。智能体感知半径 $r=6$ 。共有5位领导者，其初始位置和速度信息如表1所示，假设领导者匀速运动。

表1 领导者信息
Table 1 Leader information

领导者	初始位置	速度/(m/s)
1	(50, 10)	(0.8, -2)
2	(90, 28)	(0.2, -1.1)
3	(30, 55)	(1.2, 2.5)
4	(91, 95)	(2.2, -0.3)
5	(63, 78)	(0.8, 0)

3种算法中智能体的运动方式相同，差异在于如何确定智能体的领导-跟随关系。以算法3为例，在确定领导-跟随关系后，智能体集群的最终状态如图5所示。从图5可以看出，跟随者成功跟随领导者，未出现碰撞或丢失的情况。

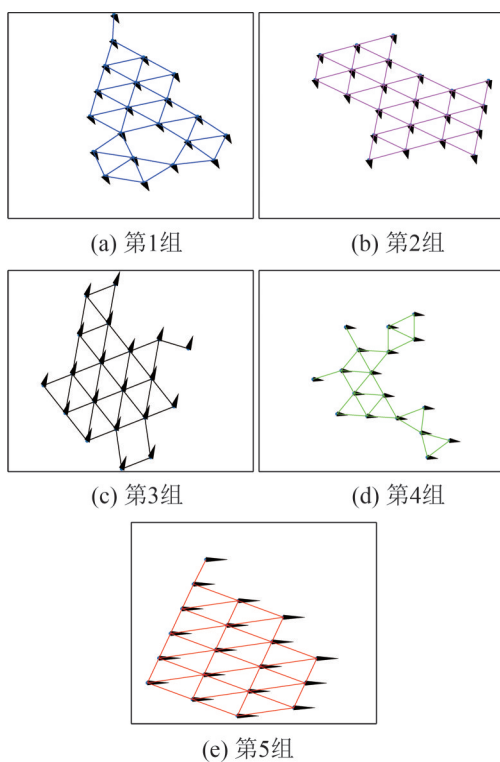


图5 智能体最终分布

Fig. 5 Final state distribution of multi-agent system

智能体的速度变化如图6所示，从图6可以看出，智能体跟随对应的领导者，速度逐渐一致，使得集群整体实现一致性。

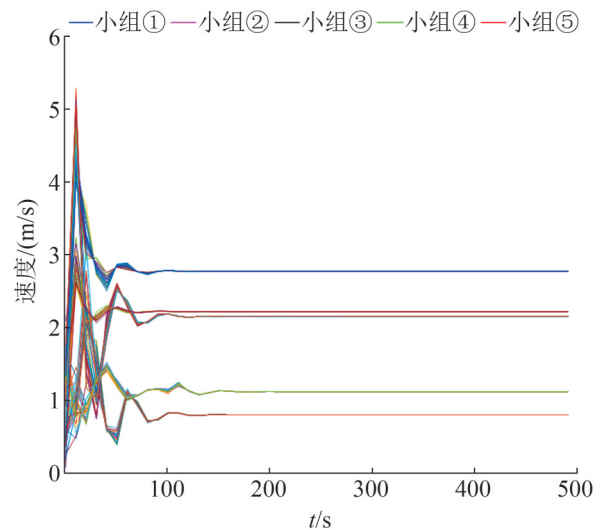


图6 智能体速度变化

Fig. 6 Multiple agent velocity variations

3.2 实验2

本实验主要分析3种算法对智能体集群分簇的优劣。其中算法3只对牵制节点聚类，而剩余智能体则靠局部通信跟随牵制智能体，所以将子图中剩余智能体划入牵制节点对应的社区中。假设领导者和智能体集群初始位置随机， $k=5$ ， $N=100$ 。共选择5组数据，计算势能函数 $\ln(F(\mathbf{P}_y, \mathbf{x}))$ ，如表2所示。

表2 势能函数值

Table 2 Values of potential function					
算法	样本1	样本2	样本3	样本4	样本5
1	8.530 1	7.648 3	8.165 8	8.413 5	8.401 1
2	7.837 6	7.493 7	7.629 1	7.966 7	7.793 1
3	7.856 9	7.454 4	7.655 6	7.962 3	7.731 7

从表2可以看出，算法3和算法2所划分的社区大致相同且要优于算法1，这说明整个网络中存在冗余的信息，而算法3仅对网络中关键节点进行聚类，既保证了足够的精度也降低了计算复杂度。多智能体集群中个体数量越多，算法3优势越大。算法1虽然保证社区内部成员紧密和社区间保持足够距离，但因为领导者与最终聚类中心的位置差异导致领导者作为社区中心时要劣于另外2种算法。

3.3 实验3

本实验主要分析3种算法对系统整体达到一致性所需时间的影响, 定义速度差异函数

$$g_i = \|q_{\gamma_i} - q_i\| \quad (11)$$

记智能体 i 与所属领导者 γ_i 保持同步条件

$$\varpi_i = 1, \text{ if } |g_i| < 0.01 \quad (12)$$

最终系统保持同步条件 $\sum_{i=1}^n \varpi_i = N$ 。在实验2的5组样本基础上, 系统最终收敛时间如表3所示。

表3 收敛时间
Table 3 Convergence time

算法	样本1	样本2	样本3	样本4	样本5	平均值
1	29.0	48.2	52.5	48.5	78.6	51.36
2	34.4	56.6	84.2	89.1	90.3	70.92
3	59.2	88.4	90.4	102.1	96.2	87.26

从表3可以看出, 算法1由于社区内部智能体间联系的更为紧密, 响应速度更快, 可以快速实现整个集群的一致性。而算法3相较于其他2种算法, 因为只有某些关键智能体可以接收到领导者信息, 剩余智能体则通过局部通信与周围智能体保持一致。所以在收敛过程中, 有些智能体会出现中途更换领导者的情况, 系统总收敛时间较长。

从以上实验可以得出以下结论:

算法1中K-means算法运行完整, 因而产生的 k 个社区彼此间距离大且社区内成员联系更为紧密。但聚类中心不可预测, 这导致社区有时距领导者较远。所以算法1更适合智能体分布空间广且需要系统快速实现一致性的场景。

算法2以领导者位置作为初始聚类中心, 相较于算法1, 社区内智能体与领导者联系的更为紧密, 且设置迭代上限, 弥补了算法1中最终聚类中心未能均匀分布在领导者周围的不足。算法2更适用于空间小, 智能体分布稀疏的场景。

算法3相较于算法1、2, 直接参与聚类的智能体数量大大减少, 降低了计算复杂度。牵制控制使得仅需少量智能体安装全局控制器, 剩余智

能体则凭借局部控制器使系统达到一致, 更加节约成本, 但系统达到一致性所需时间较长。所以在不考虑系统收敛时间, 为降低控制成本的前提下, 算法3优势更大。

4 结论

本文基于K-means算法解决多智能体集群跟随多个领导者时如何合理分配跟随者的问题。在中途任务变更或任务完成后领取新任务时, 本文所提出的算法可跟随新任务重新分配给合适的智能体执行。文中的3种算法各有优劣, 其中算法3在吸收前两者优势的基础上, 提出牵制控制以引导集群实现一致, 极大降低工程难度。集群规模越大, 智能体越密集其收敛速度越快。此外本文所做工作主要是提高多领导者的集群规模并将集群初始状态改为随机。目前关于此方向研究较少, 所以在横向比较中略显不足。但这也是接下来的研究内容, 如何将现有优秀多领导者跟随算法应用于大规模集群并提高收敛速度是未来的主要研究方向。

参考文献:

- [1] 吴虎胜, 张凤鸣, 吴庐山. 一种新的群体智能算法——狼群算法[J]. 系统工程与电子技术, 2013, 35(11): 2430-2438. Wu Husheng, Zhang Fengming, Wu Lushan. New Swarm Intelligence Algorithm-Wolf Pack Algorithm[J]. Journal of Systems Engineering and Electronics, 2013, 35(11): 2430-2438.
- [2] Cavagna A, Cimarelli A, Giardina I, et al. Scale-Free Correlations in Starling Flocks[J]. Proceedings of the National Academy of Sciences (S0369-8211), 2010, 107(26): 11865-11870.
- [3] Li Xiaolei, Lu Fei, Tian Guohui, et al. Applications of Artificial Fish School Algorithm in Combinatorial Optimization Problems[J]. Journal of Shandong University (Engineering Science) (S1672-3961), 2004, 34(5): 64-67.
- [4] Tarasewich P, McMullen P R. Swarm Intelligence[J]. Communications of the Acm (S0001-0782), 2002, 45(8): 62-67.
- [5] Ren W, Beard R W. Consensus Seeking in Multiagent Systems under Dynamically Changing Interaction

- Topologies[J]. IEEE Transactions on Automatic Control (S0018-9286), 2005, 50(5): 655-661.
- [6] Li Zhongkui, Wen Guanghui, Duan Zhisheng, et al. Designing Fully Distributed Consensus Protocols for Linear Multi-agent Systems with Directed Graphs[J]. IEEE Transactions on Automatic Control (S0018-9286), 2015, 60(4): 1152-1157.
- [7] Liu Bo, Su Housheng, Li Rong, et al. Switching Controllability of Discrete-Time Multi-agent Systems with Multiple Leaders and Time-Delays[J]. Applied Mathematics & Computation (S0096-3003), 2014, 228: 571-588.
- [8] Hou Wenyong, Wu Zongze, Fu Mingyue, et al. Constrained Consensus of Discrete-Time Multi-agent Systems with Time Delay[J]. International Journal of Systems Science (S0020-7721), 2018, 49(5): 1-7.
- [9] Wang Dong, Wang Wei. Necessary and Sufficient Conditions for Containment Control of Multi-agent Systems with Time Delay[J]. Automatica (S0005-1098), 2019, 103: 418-423.
- [10] Liang J, Liu K, Ji Z, et al. Event-Triggered Consensus Control for Linear Multi-agent Systems[J]. IEEE Access (S2169-3536), 2019, 7: 144572-144579.
- [11] Garcia E, Cao Y, Casbeer D W. Decentralized Event-Triggered Consensus with General Linear Dynamics[J]. Automatica (S0005-1098), 2014, 50(10): 2633-2640.
- [12] Olfati-Saber R. Flocking for Multi-agent Dynamic Systems: Algorithms and Theory[J]. IEEE Transactions on Automatic Control (S0018-9286), 2006, 51(3): 401-420.
- [13] Hu Y, Su H, Lam J. Adaptive Flocking with A Virtual Leader of Multiple Agents Governed by Locally Lipschitz Nonlinearity[J]. International Journal of Robust & Nonlinear Control (S1049-8923), 2013, 23(9): 978-990.
- [14] Fan S, Guo Y, Ji Y. Fixed-Time Consensus for Leader-Following Multi-groups of Multi-agent Systems with Unknown Non-Linear Inherent Dynamics[C]// The 38th Chinese Control Conference (CCC2019). Guangzhou: TCCT, 2019: 5903-5908.
- [15] Couzin I, Krause J, Franks N, et al. Effective Leadership and Decision-Making in Animal Groups on the Move[J]. Nature (S0028-0836), 2005, 433: 513-516.
- [16] Gao Leitao, Zhao Guangshe, Li Guoqi, et al. Containment Control of Directed Networks with Time-Varying Nonlinear Multi-agents Using Minimum Number of Leaders[J]. Physica, A. Statistical Mechanics and Its Applications (S0378-4371), 2019, 526: 120859.
- [17] Santo Fortunato. Community Detection in Graphs[J]. Physics Reports (S0034-4877), 2010, 486: 75-174.
- [18] Zhu X, Ghahramani Z. Learning from Labeled and Unlabeled Data with Label Propagation[J]. Tech Report (S1348-0677), 2002, 54: 2865.
- [19] Ian X Y Leung, Pan Hui, Pietro Lio', et al. Towards Real-Time Community Detection in Large Networks[J]. Physical Review E (S1539-3755), 2009, 79(6): 066107.
- [20] Wong J, Wong M. Algorithm AS 136: A K-means Clustering Algorithm[J]. Journal of the Royal Statistical Society (S0952-8385), 1979, 28(1): 100-108.
- [21] 何明, 马子玉, 刘锦涛, 等. 基于影响度介数中心性的多智能体牵制控制算法[J]. 控制与决策, 2021, 36(6): 1442-1448.
- He Ming, Ma Ziyu, Liu Jingtao, et al. Multi-agent Pinning Control Algorithm Based on Betweenness Centrality with Influence Degree[J]. Control and Decision, 2023, 36(6): 1442-1448.