

基于时空和作战编组的兵棋推演系统轨迹聚类算法

万宜春, 陈志龙[†], 何昌其, 胡 水
(陆军指挥学院, 江苏 南京 210045)

摘 要: 针对兵棋推演系统信息繁杂, 不利于用户理解作战态势的问题, 提出了基于时空和作战编组的轨迹聚类 CTUW 算法。该算法分为轨迹压缩、相似性度量、轨迹线段聚类和可视化 4 个部分, 可从繁杂的轨迹信息中提取主要内容并进行概括表达, 达到既简洁概述兵棋推演的整体态势变化, 又不丢失棋子机动过程中的重要细节信息的目的。实验结果表明, CTUW 算法的轨迹聚类效果相比 TRACCLUS 算法和 CTECW 算法更精细, 计算复杂度也较低, 即使在处理具有异常机动速度、特殊轨迹形态、路线交叉重叠等特殊轨迹数据时仍能保持较好的聚类效果。

关键词: 兵棋; 轨迹聚类; 时空; 编组

中图分类号: E919

文献标志码: A

DOI: 10.3969/j.issn.1673-3819.2023.01.018

Trajectory clustering algorithm of wargaming system based on space-time and combat grouping

WAN Yi-chun, CHEN Zhi-long, HE Chang-qi, HU Shui

(Army Command College, Nanjing 210045, China)

Abstract: In view of the complicated information of wargaming system, which is not conducive for users to understand the combat situation, a trajectory clustering algorithm CTUW (clustering trajectories of units in wargame) based on space-time and combat grouping is proposed. The algorithm is divided into four parts: trajectory compression, similarity measurement, trajectory segments' clustering and visualization. The main content can be extracted from the complicated track information and summarized, so as to achieve the purpose of concise overview of the overall situation changes of the military chess deduction without losing the important details in the process of chess maneuver. The experiment shows that the trajectory clustering effect of the CTUW algorithm is more refined than that of the TRACCLUS algorithm and the CTECW algorithm, and the computational complexity is lower. It can maintain a good clustering effect even when dealing with special trajectory data with abnormal maneuvering speed, special trajectory shape and trajectories overlap.

Key words: wargame; trajectory clustering; spatial-temporal; combat grouping

兵棋推演系统是在虚拟环境中检验作战计划的重要手段, 主要组成为棋盘、棋子和裁决规则^[1]。推演人员在兵棋推演系统上对棋子下达的各种指令, 都会按照裁决规则计算结果转化为机动、直瞄、间瞄、飞行、侦察等相应行动及棋盘态势。随着推演进程的发展, 棋子的位置也在不断发生变化。为提供推演后的复盘回放和数据分析功能, 兵棋系统需记录下推演过程中所有棋子的位置信息, 形成大量轨迹数据。这些轨迹数据除了位置信息, 还记录了途经时间以及关联棋子的基本信息。在战术及以上级别的兵棋推演中, 轨迹数据存在机动形态杂、数据存储大和轨迹不连续等问题。另外, 轨迹数据在时间维度、空间维度和棋子维度上涉及范围广, 加上停留点和细微机动调整点的存在, 使得兵棋轨迹数据更加复杂。传统兵棋系统往往缺乏对轨

迹数据的处理手段, 直接将其呈现在兵棋系统上, 不利于指挥员快速理解作战态势, 给复盘分析、组织协同和方案评估等工作带来较大困难^[2]。为此, 有必要研究轨迹聚类方法, 从繁杂的轨迹信息中提取主要内容, 使其既能简洁概述兵棋推演的整体态势变化, 又不丢失棋子机动过程中的重要细节信息。

本文回顾了轨迹聚类的相关工作, 设计了 CTUW 算法, 通过实验测试了 CTUW 算法的运行效果, 验证了其用于兵棋推演系统轨迹聚类的可行性。

1 轨迹聚类的相关研究

轨迹聚类可被视为一种数据处理框架, 涉及轨迹压缩、相似性度量、聚类 and 可视化等 4 个步骤, 前 3 项内容是轨迹聚类的中心工作^[3]。

轨迹压缩主要完成去除冗余点和提取关键点的工作, 在不丢失轨迹关键特征信息的前提下, 尽可能压缩轨迹存储信息。李名^[4]等引入了经典的 Douglas-Peucker 算法^[5]用于查找停留点等冗余信息, 以达到压缩轨迹信息的目的。Li Q^[6]等提出基于几何特性的停

收稿日期: 2022-07-01

修回日期: 2022-07-21

作者简介: 万宜春(1990—), 男, 讲师, 研究方向为联合作战与兵棋推演。

[†] 通信作者: 陈志龙(1988—), 男, 讲师。

留点检测方法,算法简单高效,但需要根据不同样本调整参数。徐凯^[7]等将时间维纳入考虑依据,将轨迹压缩算法转换到向量空间执行,进一步改善了压缩效率。董婉婷^[8]等基于 Sliding Window 算法^[9],在删除停留点的同时,尽可能保留轨迹特征点,丰富了压缩轨迹的局部形态特征。

相似性度量用于对轨迹之间相似性进行评估量化,是聚类算法的前提和数据基础。常用的相似性度量方法包括欧氏距离、隐马尔科夫模型、最长公共序列和动态时间调整等。Lee^[10]等从垂直距离、平行距离和角度聚类3个层面量化空间维度上的相似度,该方法不受轨迹线段长短的限制,较好地度量了轨迹线段之间的空间相似度。郭名静^[11]和龙真真^[12]等利用多维属性对聚类划分的共同影响,从非地理空间的数据维度入手,设置了不同数据维度对空间距离的影响权值,以修正相似性度量函数。周超^[13]等结合军事运用的需求,构建了基于实体、行动和交互等要素的兵力聚合模型,使用层次分析法确定权重影响矩阵,并依此计算相似度。

聚类算法是轨迹聚类的核心,目的是将相似轨迹聚合为类簇^[14]。典型的聚类算法包括基于划分的 K-means 算法、基于层次的 CHAMELEON 算法、基于网络的 CLIQUE 算法、基于图论的 Spectral Clustering 算法、基于密度的 DBSCAN 算法和 DPC 算法等,它们被广泛应用于模式识别、机器学习和数据挖掘等领域。兵棋推演系统的轨迹信息属于分布形态随机性大、类簇重叠频率高的数据样本,大多数传统聚类算法处理这类数据样本的效果均不佳^[15]。然而,DBSCAN^[16]和 DPC^[17]等基于密度的聚类算法能够适应非球形分布的数据样本。DBSCAN 算法提出的密度可达概念,可以在嘈杂的样本中找到形状各异的类簇,但是该算法的输入参数设定缺乏理论支撑,参数值受主观影响较大,且在处理密度重叠样本方面存在短板。DPC 算法,又称为密度峰值聚类算法,除能够解决非球形聚类问题外,还更容易确定唯一的输入参数,在处理类簇重叠的数据样本时,也能保持较好的聚类效果,因此近年来受到广泛关注^[15]。

本文的轨迹聚类基本思路是先分割后聚合,与该思路比较接近的有 TRACCLUS 算法^[10]和 CTECW 算法^[3],但是这些算法存在以下不足。

1) 在轨迹压缩方面,TRACCLUS 算法采用 MDL(最小描述长度)方法进行轨迹压缩,压缩效果不佳,且计算较复杂^[18]。CTECW 算法引入了简化轨迹概念,先将原始轨迹转化为简化轨迹,再进行分段,进一步优化了轨迹分段效率,但还是无法查找停留点,轨迹压缩率有

待进一步提高。

2) 在相似性度量方面,TRACCLUS 算法和 CTECW 算法仅从空间维度上定义了相似度,只能根据空间距离相邻关系对轨迹进行聚类。若直接将这两个算法应用于兵棋推演系统,其无法利用轨迹的时间和棋子编组关系等信息进行精细化轨迹聚类,导致静态图形式的聚类结果无法反映不同作战时节、不同作战力量、不同类型行动的动态机动情况,影响了轨迹聚类的军事应用意义。

3) 在轨迹线段聚类方面,TRACCLUS 算法和 CTECW 算法均是基于 DBSCAN 聚类算法的基本框架。受到 DBSCAN 算法短板的影响,这两个算法无法适应类簇交叉重叠的数据样本,且 DBSCAN 算法的输入参数较多,参数值受主观影响波动较大。虽然 CTECW 算法使用密度估计熵为参数值的选择提供了重要理论依据,但实质上变相增加了数据预处理的环节,降低了算法应用的灵活性。

4) 在可视化方面,TRACCLUS 算法为每个轨迹线段类簇提取代表轨迹后,没有进一步处理,导致代表轨迹线段较为凌乱,可读性不高。CTECW 算法通过分析代表轨迹之间的后继关系,试图进一步提升轨迹聚类效果,但是其使用的后继关系分析方法容易出现“后继循环”,反而可能造成轨迹聚类结果的错误表达。

针对以上不足,本文提出 CTUW 算法:首先,对原始轨迹进行压缩,去除停留点等冗余信息,提取轨迹特征点并进行轨迹分段;再基于提出的时间、空间和编组关系相似度计算方法,定量描述轨迹线段之间的相似性;然后,使用 DPC 聚类算法对轨迹线段进行聚类;最后,根据兵棋推演系统轨迹聚类的军事需求,可视化展示聚类结果。

2 CTUW 算法

2.1 轨迹压缩

定义1 棋子轨迹:每个棋子在推演过程中都会产生一条轨迹记录。从棋子轨迹起点开始,兵棋系统每隔一段时间采集棋子位置信息,这些信息按照采集时间由先到后的顺序排列得到的序列即为棋子轨迹。用 Tra 表示棋子轨迹:

$$Tra = p_1 p_2 \cdots p_i p_{k+1} \cdots p_n \quad (1)$$

其中 $p_k = (x_k, y_k, t_k)$, x_k 为棋子位置的横坐标, y_k 为棋子位置的纵坐标, t_k 为棋子途径位置时的作战时间。

定义2 特征点:在轨迹中,棋子的运动趋势发生较大改变的点称为特征点。特征点集合用 F 表示。

定义3 停留点:在轨迹中,若某时刻记录的棋子

位置点相比前一时刻没有发生明显的位置变化,则称为停留点,用 $s_i (1 < i \leq n)$ 表示。如图 1 所示,停留点分为两类:一类是棋子停留在一个位置持续较长时间,如 p_3, p_4, p_5 点的位置均相同,但这 3 个点的时间 t 不同, p_3 点相比 p_2 点产生了较大位置变化,所以 p_3 点不是停留点,而 p_4 和 p_5 点为停留点。另一类是棋子在某个中心点附近区域徘徊,如 p_7, p_8, p_9, p_{10} 点的时间 t 相近,且这 4 个点均在以 A 点为中心的一个较小范围内, p_7 点相比 p_6 点产生了较大位置变化,所以 p_7 点不是停留点,而 p_8, p_9, p_{10} 点为停留点。

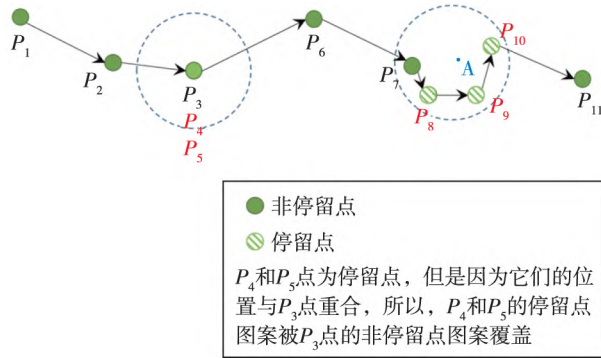


图 1 停留点示意图

Fig. 1 Schematic diagram of dwell point

定义 4 轨迹线段:选取轨迹中的 p_i 点和 p_j 点 ($i < j$) 将 p_i 点和 p_j 点连接组成线段,用 $L = p_i p_j$ 表示。

定义 5 轨迹段:选取轨迹中的多个点 $\{p_{k_1}, p_{k_2}, \dots, p_{k_m}\} (k_1 < k_2 < \dots < k_m)$ 将 p_{k_i} 点和 $p_{k_{i+1}}$ 点组成轨迹线段 ($i \in \{1, 2, \dots, m-1\}$) 这些轨迹线段的集合称为轨迹段,表示为

$$T = \{p_{k_1} p_{k_2}, p_{k_2} p_{k_3}, \dots, p_{k_m} p_{k_{m+1}}, \dots, p_{k_{n-1}} p_{k_n}\} \quad (2)$$

按照定义 5,棋子的原始轨迹可以写成轨迹段形式,表示为

$$T_{\text{original}} = \{p_1 p_2, p_2 p_3, \dots, p_i p_{i+1}, \dots, p_{n-1} p_n\} \quad (3)$$

兵棋系统为了记录推演过程,将每隔一段时间记录一次棋子的位置和采集时间,由此构成棋子的轨迹信息。轨迹中有一些冗余信息,不仅对轨迹聚类的最终效果没有贡献,反而会增加轨迹聚类算法的时间和空间复杂度。冗余信息主要分为两类:一类是非特征点带来的冗余信息,即棋子运动趋势基本保持不变的一段轨迹上的若干个连续轨迹信息,如图 2 所示,非特征点 p_2 参与的轨迹段 $\{p_1 p_2, p_2 p_3\}$ 描述的是略微曲折的一小段轨迹,而这种信息对反映棋子整体机动意图的作用不大。如果将其简化为轨迹段 $\{p_1 p_3\}$,不仅不会影响轨迹信息的概述,还将降低轨迹聚类算法的空间和时间复杂度,甚至还能避免轨迹聚类算法出现过度拟合。另一类冗余信息是停留点带来的冗余信息,即处于停留状态的一段轨迹上的若干个连续轨迹信息,

如图 2 所示,轨迹段 $\{p_3 p_4, p_4 p_5\}$ 和轨迹段 $\{p_7 p_8, p_8 p_9, p_9 p_{10}\}$ 都处于停留状态,它们无须被轨迹聚类,因此可以将其删除。

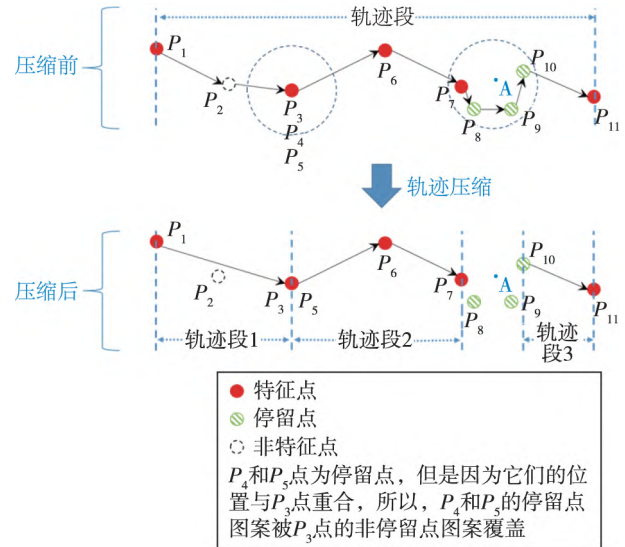


图 2 轨迹压缩示意图

Fig. 2 Schematic diagram of track compression

定义 6 轨迹压缩:从原始轨迹删除停留点和非特征点带来的冗余信息后,形成多个新的轨迹段,这些轨迹段组成的集合称为压缩后的轨迹,用 T_{compress} 表示。

如图 2 所示,棋子原始轨迹 Tra 对应的轨迹段为 $\{p_i p_{i+1} | 1 \leq i \leq 10\}$,经过轨迹压缩后,得到轨迹段集合:

$$T_{\text{compress}} = \{\{p_1 p_3\}, \{p_5 p_6, p_6 p_7\}, \{p_{10} p_{11}\}\}$$

压缩前原始轨迹共有 10 个轨迹线段,压缩后仅为 4 个轨迹线段,说明轨迹信息确实被压缩。另外,原始轨迹为 1 个轨迹段,压缩后被切分化为 3 个轨迹段。这种轨迹压缩方法虽然把连续轨迹转化为不连续的多个轨迹段,但本文的 CTUW 算法基本思路是先分割后聚合,轨迹压缩环节要为后续的聚类算法提供的本来就是分割后的轨迹线段,且轨迹间断的位置对应的是对轨迹聚类没有帮助的停留点信息,所以,压缩后轨迹可能不连续的结果不会对轨迹聚类造成不良影响。

如图 3 所示,轨迹压缩分为 3 个阶段:查找原始轨迹停留点,将原始轨迹切割为多个轨迹段,查找每个轨迹段特征点。

设原始轨迹点数量为 n ,算法具体步骤如下:

- 1) $i = 2$;
- 2) 若 i 大于 n ,则跳转到第 7) 步;

3) 从 p_{i-1} 点开始,向轨迹起点方向遍历,寻找首次满足以下条件的 p_j 点: p_j 点与 p_i 点的时间距离大于等于时间阈值 $timeThresh$,且 p_{j+1} 点与 p_i 点的时间距离小于时间阈值 $timeThresh$ 。若无法找到这样的 p_j 点,则 $i = i + 1$ 。

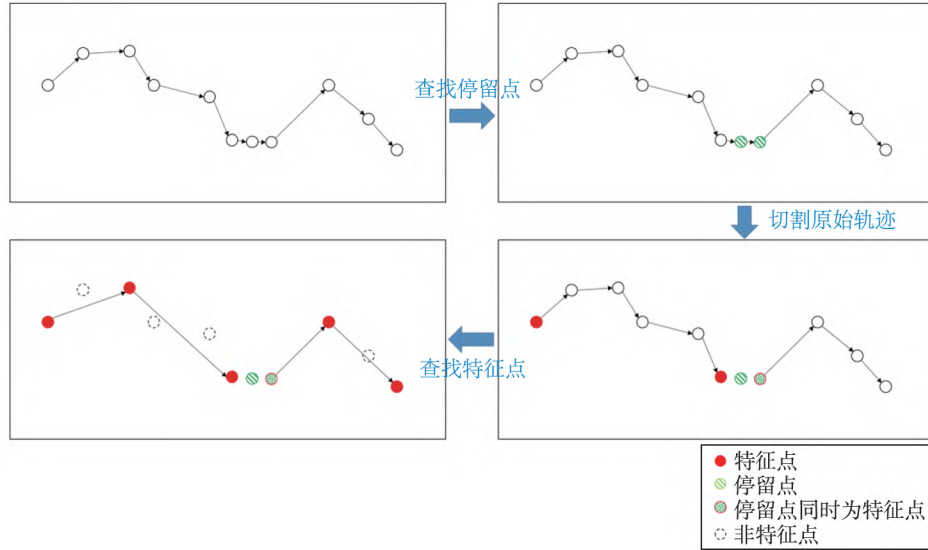


图3 轨迹压缩流程示意图

Fig. 3 Schematic diagram of track compression process

1 且跳转到第2)步;

4) 若 p_j 点与 p_i 点的空间距离小于距离阈值 $distThresh$ 则从 p_j 点开始,沿原始轨迹 Tra 向轨迹起点方向遍历,寻找首次满足以下条件的 p_k 点: p_{k-1} 点与 p_j 点的空间距离大于等于距离阈值 $distThresh$ 且 p_k 点与 p_j 点的空间距离小于距离阈值 $distThresh$ 。若遍历到 p_1 点时,都没有找到满足该组合条件的 p_k 点,则令 $k = 1$;

5) 点集合 $\{p_h | k + 1 \leq h \leq i\}$ 中所有点均为停留点;

6) $i = i + 1$ 且跳转至第2步;

7) 按照以下规则,初步设置轨迹上的特征点:①若 p_h 点为停留点且 p_{h+1} 点为非停留点,则 p_h 点也是特征点;②若 p_h 点为非停留点且 p_{h+1} 点为停留点,则 p_h 点是特征点;③ p_1 点为特征点;④若 p_n 非停留点,则为特征点;

8) 从原始轨迹 Tra 中删除那些不是特征点的停留点,则原始轨迹被切割为若干个轨迹段 T_1, T_2, \dots, T_m 。根据第7)步的原则,这些轨迹段的起点和终点肯定为特征点;

9) 针对 T_1, T_2, \dots, T_m 中的每个轨迹段,都进行以下步骤寻找轨迹段内的特征点,将找到的特征点连线成为压缩后的轨迹段;

10) 假设某个轨迹段起点的序号为 s ,终点的序号为 e ,则该轨迹段对应的轨迹为 $p_s p_{s+1} \dots p_e$;

11) 设置该轨迹上的滑动窗口 $\{p_i p_{i+1} \dots p_j\}$,其中 $s \leq i \leq j \leq e$,初始化该窗口为 $i = s, j = i + 1$;

12) 针对滑动窗口内 p_{i+1} 点、 p_{i+2} 点、 \dots 、 p_j 点,逐个计算每个点到线段 $p_i p_j$ 的欧氏距离;

13) 若这 $j - i$ 个欧氏距离均小于距离阈值

$feaThresh$,则 $j = j + 1$ 且跳转到第12)步。否则 p_{j-1} 点为特征点 $j = j - 1$ 且跳转到第12)步。

以上算法中,步骤1)~6)为查找原始轨迹停留点的阶段,步骤7)~8)为将原始轨迹切割为多个轨迹段的阶段,步骤9)~13)为每个轨迹段查找特征点的阶段。其中 $timeThresh$ 和 $distThresh$ 的含义是棋子途径某个点之后 $timeThresh$ 时间,棋子仍在该点附近 $distThresh$ 距离范围内,则认为棋子处于停留状态。

轨迹压缩算法的步骤第9)~13),采用 Sliding Window 算法查找特征点,该算法擅长局部轨迹的压缩优化。兵棋棋子轨迹压缩,需要体现关键路口、特殊地貌和限制地物等通行环境对机动路线规划的影响,还需要体现穿插、迂回和渗透等行动的特点,所以,兵棋轨迹压缩需要在删除冗余信息的同时,保留更多局部特征点。Sliding Window 算法较好满足了该需要。该算法的基本思想是使用滑动窗口向后遍历轨迹,每次以刚找到的最新特征点为滑动窗口的起点向后探索,找到下一个特征点作为滑动窗口的终点,下一特征点还应满足以下两个条件:滑动窗口起点和终点组成连线,窗口内所有点到该连线的欧氏距离小于设定的距离阈值;在固定滑动窗口起点的前提下,尽可能使窗口范围更大。如图4所示,设原始轨迹起点 p_1 为特征点,初始化滑动窗口为 $\{p_1 p_2\}$ 。滑动窗口 $\{p_1 p_2\}$ 内的两个点到线段 $p_1 p_2$ 的距离小于设定的距离阈值 $feaThresh$ 。进一步扩大滑动窗口范围,将滑动窗口的终点后移一位,扩展为窗口 $\{p_1 p_2 p_3\}$,该窗口内所有点到线段 $p_1 p_3$ 的欧氏距离均小于设定的距离阈值 $feaThresh$ 。尝试进一步扩大滑动窗口为 $\{p_1 p_2 p_3 p_4\}$,在该窗口内 p_3 点到线段 $p_1 p_4$ 的欧氏距离大于设定的距离阈值 $feaThresh$,说明

$\{p_1, p_2, p_3\}$ 是固定滑动窗口起点 p_1 的前提下能够找到的符合搜索条件的宽度最大的滑动窗口。滑动窗口 $\{p_1, p_2, p_3\}$ 的终点 p_3 为轨迹上特征点 p_1 之后的下一个特征点。接着以 p_3 点作为新的滑动窗口的起点,将滑动窗口设定为 $\{p_3, p_4\}$,重复以上步骤。在滑动窗口 $\{p_3, p_4\}$ 逐步扩展到 $\{p_3, p_4, p_5, p_6\}$ 的过程中,发现滑动窗口内所有点到滑动窗口起点与终点连线的欧氏距离始终小于设定的距离阈值 $feaThresh$ 。因此,滑动窗口 $\{p_3, p_4, p_5, p_6\}$ 的终点 p_6 也为特征点。最终,图4中轨迹特征点为 p_1 、 p_3 和 p_6 点。

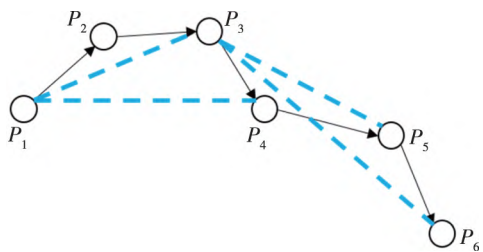


图4 轨迹压缩流程示意图

Fig. 4 Schematic diagram of track compression process

2.2 相似性度量

轨迹聚类是对多个棋子在一段时间内产生的众多轨迹线段进行聚类,聚类的前提是能够准确描述轨迹线段两两之间的相似性。相似性度量的具体方法直接体现了轨迹聚类的意图,影响轨迹聚类的效果。石崇林等针对兵棋系统提出轨迹聚类算法 CTECW,巧妙设计了轨迹线段之间的空间距离计算方法,能够满足对称性、非负性和三角不等式关系。但是,CTECW 算法忽视了兵棋轨迹是复合了时间、空间和关联棋子信息的组合体,不能单纯将地理位置相近的多个轨迹聚合在一起,而应将时间、空间和关联棋子的编组关系都相近的多个轨迹聚合在一起。只有这样,轨迹聚类的最终结果才不是一个静态的线段组合图,而是一个能够反映时空动态变化的多群队机动态势图。

如图5a)所示,10个棋子分为左翼攻击群、纵深攻击群、右翼攻击群和穿插突击队等4个群队,产生了多条轨迹,图例说明了每条轨迹的发生时间以及隶属群队名称。

如图5b)所示,传统轨迹聚类算法 TRACCLUS 的聚类效果仅能反映全局棋子的运动趋势,因为其只将轨迹线段之间的空间距离作为相似性度量依据,所以聚类效果不精细。

如图5c)所示,CTECW 算法相比 TRACCLUS 算法,优化了相似性度量方法和轨迹线段聚类算法,聚类结果反映了更多的细节信息。但是 CTECW 算法还是仅从空间距离这一层面优化了相似性度量方法,导致

聚类类簇内部各轨迹线段在群队关系和时间方面不具备相似性,不易理解聚类结果。图5c)左侧的轨迹线段类簇中,包含了两个群队的轨迹,而且原本应分为3个时间段($T+0'$ 至 $T+50'$ 、 $T+40'$ 至 $T+70'$ 、 $T+90'$ 至 $T+150'$)的轨迹被合并到了该类簇中。图5c)右侧的轨迹线段类簇,包含了3个群队的轨迹,原本应分为4个时间段($T+0'$ 至 $T+50'$ 、 $T+40'$ 至 $T+70'$ 、 $T+90'$ 至 $T+150'$ 、 $T+0'$ 至 $T+120'$)的轨迹被合并到了该类簇中。使用 CTECW 得到的类簇内部,轨迹线段的编组关系和时间分布较为凌乱且缺乏规律,难以清晰体现轨迹聚类后的军事含义,影响了该算法的军事应用价值。

如图5d)所示,本文提出的 CTUW 算法从时间、空间和编组关系等3个维度进行轨迹线段的相似性度量。因此,不仅能够将相邻的多个群队轨迹分离开,还能对纵深攻击群各作战时节机动轨迹进一步分割,最终将4个群队10个棋子的13条轨迹线段聚类成5个类簇,每个类簇内部的轨迹在时间、空间和编组关系上均具相似性。按照该相似性度量的思路,能使聚类结果既细致体现多群队机动轨迹的时空变化规律,又便于精炼概括各个聚类类簇的军事含义。

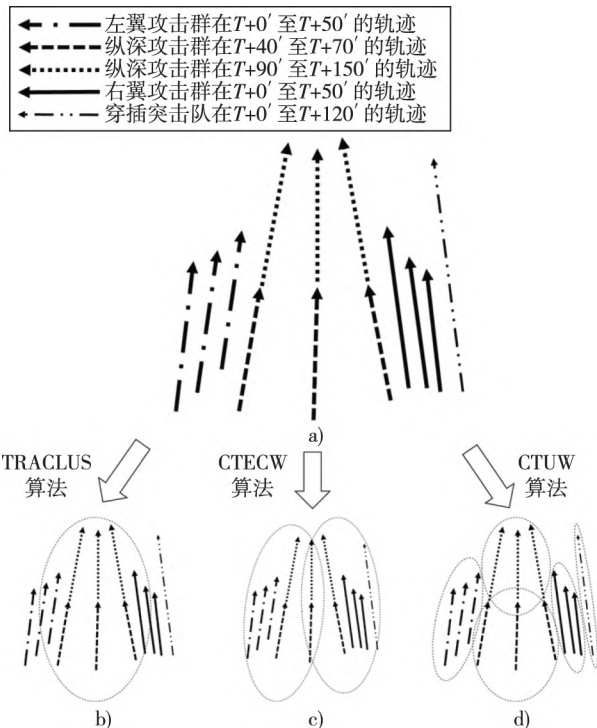


图5 轨迹聚类效果对比图

Fig. 5 Comparison of trajectory clustering effects

2.2.1 空间相似度

CTECW 算法能够较为全面地计算两个轨迹线段之间的空间距离。本文在此基础上,改进了部分计算公式,并就相关权值的选取给出了分析。

定义7 空间平移距离: 轨迹线段 L_i 与 L_j 之间的

空间平移距离,指线段 L_i 中点与线段 L_j 中点连线的线段长度,用 $geoShiftDist_{ij}$ 表示。

定义8 空间角度距离: 轨迹线段 L_i 与 L_j 之间的空间角度距离,指平移线段 L_i 使其中点与线段 L_j 中点重合后,线段 L_i 终点与线段 L_j 终点连线的线段长度,即

$$geoAngleDist_{ij} = \sqrt{\|L_i\|^2 + \|L_j\|^2 - 2 \cdot \|L_i\| \cdot \|L_j\| \cdot \cos(\theta_{ij})} \quad (4)$$

其中 θ_{ij} 为线段 L_i 与线段 L_j 之间的夹角,且满足 $0 \leq \theta_{ij} \leq \pi$ 。

定义9 空间距离: 轨迹线段 L_i 与 L_j 之间的空间距离由空间平移距离和空间角度距离组成,即

$$geoDist(L_i, L_j) = \omega_{geoShift} \cdot geoShiftDist_{ij} + \omega_{geoAngle} \cdot geoAngleDist_{ij} \quad (5)$$

其中权重 $\omega_{geoShift}$ 和 $\omega_{geoAngle}$ 的取值依据不同聚类需求。 $geoShiftDist_{ij}$ 反映两个轨迹线段质心间的距离, $geoAngleDist_{ij}$ 反映两个轨迹线段的向量夹角大小和线段长度差异。相关权重的默认取值为: $\omega_{geoShift} = 1$, $\omega_{geoAngle} = 1$ 。根据不同应用场景,还可以调整权重值: 若轨迹线段聚类侧重运动方向相似、线段长度相似等因素,那么,可以适当增加 $\omega_{geoAngle}$ 的取值; 若仅侧重考虑轨迹线段间隔距离,则可适当增加 $\omega_{geoShift}$ 的取值。另外需要说明的是,在 CTUW 算法执行聚类前,会对时间、空间和编组关系这3个相似度值进行归一化处理,所以调整权重 $\omega_{geoShift}$ 和 $\omega_{geoAngle}$ 取值的主要目的是调整权重的相对比值,而不是权重的绝对大小。

2.2.2 时间相似度

在兵棋推演过程中,常常出现两个棋子先后通过同一路段的情况。例如,棋子 A 在 t_1 时刻途经1号点,沿着某公路机动,在 t_2 时刻到达2号点; 棋子 B 在 $t_1 + \Delta t$ 时刻途经1号点,沿着同一公路机动,在 $t_2 + \Delta t$ 时刻到达2号点。两个棋子的轨迹线段起始时间差为 Δt , 终点时间差也为 Δt 。若按照“起始时间差 + 终点时间差”的方式计算时间距离的话,将得到 $2 \cdot \Delta t$, 这会过度放大两个轨迹线段之间的时间相似度,影响轨迹聚类效果。相比之下,两个轨迹线段的时间距离取值 Δt 可能更合适。为此,本文取起始时间差和终点时间差的最大值作为时间相似度。

假设轨迹线段 $L_i = p_s p_{e_i}$, p_s 点是 L_i 的起点, p_{e_i} 点是 L_i 的终点,途径 p_s 点的时刻为 t_{s_i} , 途径 p_{e_i} 点的时刻为 t_{e_i} , 则轨迹线段 L_i 的发生时间段为 t_{s_i} 至 t_{e_i} 。同理,轨迹线段 L_j 的发生时间段为 t_{s_j} 至 t_{e_j} 。两个线段的起始时间差为 $|t_{s_i} - t_{s_j}|$, 终点时间差为 $|t_{e_i} - t_{e_j}|$ 。

定义10 时间距离: 轨迹线段 L_i 与 L_j 之间的时间

距离为

$$timeDist(L_i, L_j) = \max(|t_{s_i} - t_{s_j}|, |t_{e_i} - t_{e_j}|) \quad (6)$$

其中,函数 $\max(x, y)$ 表示取 x 与 y 中的最大值。

2.2.3 编组关系相似度

轨迹线段 L_i 与 L_j 编组关系相似度主要用于描述 L_i 关联棋子与 L_j 关联棋子之间的编组关系亲疏程度,作为相似性度量的组成部分,引导聚类算法尽可能将同一群队的轨迹聚为一类。

在实际作战和兵棋推演中,参战单位是以作战编组的结构进行组织的,而且作战编组结构是作战和推演前的已知信息,基于该信息即可知晓每个作战实体或棋子隶属于哪个群队。如图6所示,作战编组可用树状结构表示。一般情况下,兵棋系统的各个棋子为树状结构中的叶子节点。为提升算法普适性,本文假设该结构的非叶子节点也能对应兵棋棋子。

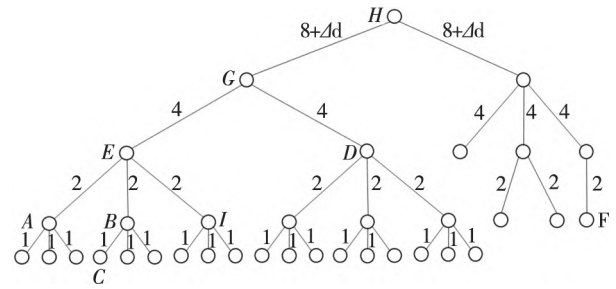


图6 编组关系树状结构图

Fig. 6 Tree structure of grouping relationship

兵棋系统中,两个棋子的编组关系相似性就是它们在作战编组结构中的关系远近程度。根据作战编组含义,为满足轨迹聚类需求,编组关系相似性度量算法应满足以下几个性质: 1) 低层级的兄弟棋子之间编组关系比高层级的兄弟棋子之间编组关系紧密; 2) 父子棋子的编组关系比兄弟棋子紧密; 3) 兄弟棋子之间编组关系比叔侄棋子紧密; 4) 隔代少的近亲棋子之间编组关系比隔代多的紧密; 5) 共同祖先层级低的棋子之间编组关系比共同祖先层级高的紧密; 6) 同一棋子的两个轨迹线段编组关系距离为0。

定义11 编组关系距离: 轨迹线段 L_i 与 L_j 之间的编组关系距离,也是它们关联棋子 C_i 和 C_j 的编组关系距离,即棋子在编组关系树状结构图中的最短路径的长度,表示为 $orgaDist(L_i, L_j)$ 或者 $orgaDist(C_i, C_j)$ 。其中,树状结构中每条边的长度值按照如下规则设置: 最低层级的边长为1,上一层级的边长是下一层级的2倍,最顶部边的边长在此计算模式基础上额外增加特定正数 Δd , Δd 的取值根据具体应用场景而定。

结合图6示例,检验定义11是否满足编组关系相似度应具备的几个性质: 低层级的兄弟棋子

$orgaDist(A, B)$ 为 4, 高级别的兄弟棋子 $orgaDist(D, E)$ 为 8, 满足性质 1); 父子棋子 $orgaDist(E, A)$ 为 2, 兄弟棋子 $orgaDist(A, B)$ 为 4, 满足性质 2); 兄弟棋子 $orgaDist(A, B)$ 为 4, 叔侄棋子 $orgaDist(A, C)$ 为 5, 叔侄棋子 $orgaDist(D, A)$ 为 10, 满足性质 3); 隔 1 代的近亲棋子 $orgaDist(D, A)$ 为 10, 隔 2 代的近亲棋子 $orgaDist(D, C)$ 为 11, 满足性质 4); H 比 G 的层级高, 共同祖先为 G 的棋子 $orgaDist(D, A)$ 为 10, 共同祖先为 H 的棋子 $orgaDist(A, F)$ 为 $28 + 2 \cdot \Delta d$, 满足性质 5); 结合定义 11, 易证其满足性质 6)。

2.2.4 轨迹线段相似度距离

定义 12 轨迹线段相似度距离: 轨迹线段 L_i 与 L_j 之间的相似度距离表示为

$$Dist(L_i, L_j) = \omega_{geo} \cdot geoDist'(L_i, L_j) + \omega_{time} \cdot timeDist'(L_i, L_j) + \omega_{orga} \cdot orgaDist'(L_i, L_j) \quad (7)$$

其中 $geoDist'$ 、 $timeDist'$ 和 $orgaDist'$ 为归一化后的数值。默认情况下, 权重 ω_{orga} 取值为 1。因为同一编组内棋子的轨迹相对集中, 编组关系距离与空间距离具有强相关性, 为避免重复放大差异, 权重 ω_{geo} 取值应小于 1。当仅需要考虑全时段机动轨迹整体趋势时, 权重 ω_{time} 可以小于 1, 甚至取值为 0。当需要详细展示不同作战时节机动轨迹聚类情况时, 权重 ω_{time} 应该大于等于 1。

$Dist(L_i, L_j)$ 的取值越小, 轨迹线段 L_i 与 L_j 之间相似度越高。

2.3 轨迹线段聚类

兵棋推演系统轨迹线段的聚类具有以下几个特点: 1) 每场兵棋推演可能都不一样, 轨迹聚类前无法向聚类算法提供先验知识; 2) 与决心图的进攻方向标绘方式类似, 轨迹线段非球形分布, 通常呈曲线分布形态; 3) 群队之间可能在时间和空间上存在轨迹的重叠或交叉; 4) 高速机动棋子(如飞机)或者特殊作战节奏的棋子(如战斗预备队)轨迹在聚类过程中, 常以噪声的形式存在, 不适合与其他轨迹有效聚合, 需要查找到这类特殊轨迹并在聚类结果中单独呈现, 以突出诸军兵种机动方式的多样性。

DPC 聚类算法能较好适应该应用场景, 因为该算法无需先验知识就可以处理非球形分布的数据, 在处理类簇重叠的数据方面有较好的聚类效果, 并且具有较强的噪声处理能力。本文采用 DPC 算法的框架来进行轨迹线段的聚类。

设 D 是压缩轨迹 $T_{compress}$ 内所有轨迹线段的集合, 表示为 $D = \{L_1, L_2, \dots, L_m\}$ 。

定义 13 局部密度: 轨迹线段 L_i 的局部密度为

$$\rho_i = \sum_{L_j \in D \text{ 且 } L_j \neq L_i} \exp\left(-\frac{Dist^2(L_i, L_j)}{d_c^2}\right) \quad (8)$$

其中 d_c 为截止距离, 是 DPC 算法需要人工设定的唯一参数。默认情况下 d_c 的取值达到如下情况时为佳: 将所有轨迹线段之间的相似度距离按照由小到大的顺序排序, d_c 是该序列前 1% 至 2% 位置处的值。根据轨迹聚类的不同需求, 还可进一步调整 d_c 取值: 当需要进一步详细切分不同作战时节、不同作战地域、不同作战编组的轨迹聚类效果时, 可适当减小 d_c 取值; 当需要从全局、全程、全时的更高视角通盘掌握棋子轨迹态势时, 可适当增加 d_c 取值。

定义 14 相对距离: 轨迹线段 L_i 的相对距离指局部密度大于 ρ_i 的轨迹线段中离 L_i 最近的相似度距离。但当轨迹线段 L_i 的局部密度最大时, 该线段的相对距离取值为所有轨迹线段相对距离的最大值。定义为

$$\delta_i = \begin{cases} \max_j (Dist(L_i, L_j)) & \text{如果 } \rho_i \text{ 为最大局部密度} \\ \min_{j: \rho_j > \rho_i} (Dist(L_i, L_j)) & \text{其他情况} \end{cases} \quad (9)$$

按照以上定义, 可以得到每个轨迹线段的局部密度和相对距离, 将其绘制在决策图上, 如图 7 所示。对图 7a) 中各个轨迹线段进行编号, 然后, 基于轨迹线段之间的相似度距离使用 DPC 算法聚类, 得到的决策图 7b)。

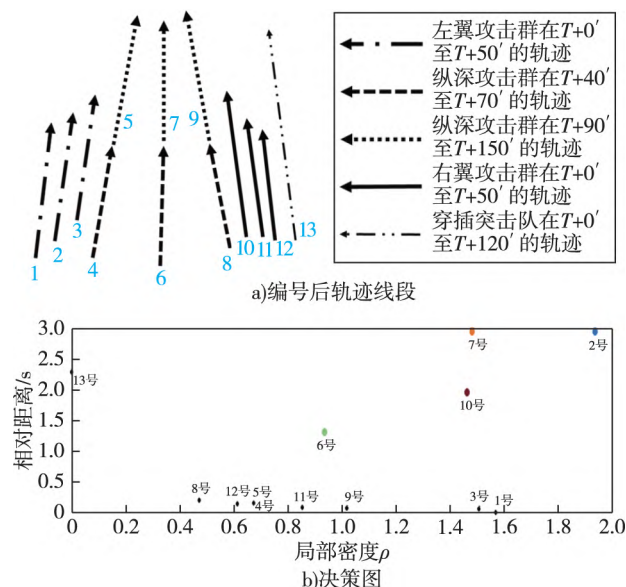


图 7 编号后的轨迹线段与决策图对应关系

Fig. 7 Corresponding relationship between the numbered track segment and the decision diagram

选取决策图中局部密度 ρ_i 和相对距离 δ_i 都较大的轨迹线段作为类簇中心。如图 7b) 所示, 选取了 2、6、7、10 号轨迹线段作为类簇中心。

再将其余非类簇中心的各个轨迹线段划分到距离其最近的更高局部密度轨迹线段所属类簇中, 即可完

成轨迹线段的初步聚类。如图8所示,依据已经确定为类簇中心的4个轨迹线段,其他非类簇中心的轨迹线段也将归类到距离其较近类簇中,例如,在比3号轨迹线段局部密度大的所有轨迹线段中,2号是离3号最近的轨迹线段,因此,3号轨迹线段应与2号归为一类,依此方法,即可将13条轨迹线段初步划分为4个类簇。

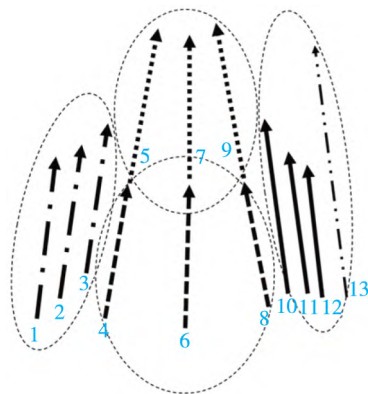


图8 轨迹线段初步聚类效果图

Fig. 8 Preliminary clustering effect of track segments

为了更好地说明聚类效果,我们将本文中提出的时间、空间和编组关系这3个维度的距离压缩为2个维度,如图9所示,横坐标为时间距离和空间距离的复合表达,纵坐标为编组关系距离,图9中的编号与如图7b)对应。结合图7b)的颜色示例,图9还说明了初步聚类的结果,例如图9左下角的3个蓝色样本点及其附近文字说明了1、2和3号轨迹线段同属于以2号轨迹线段为中心的类簇。

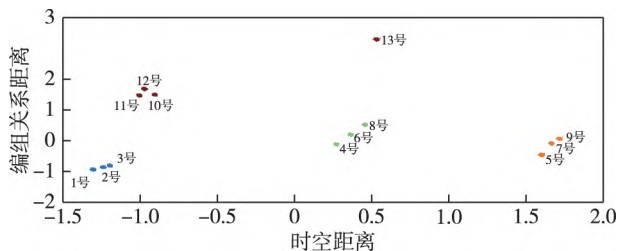


图9 轨迹线段相似度距离的二维视图

Fig. 9 Two dimensional view of track segment similarity distance

图9中,10、11、12和13号都是棕色,图7b)中10号也是棕色,说明图7a)中的10、11、12和13号轨迹线段被初步聚类归为一个类簇。但是从图7a)上可以看出,13号轨迹线段与其他轨迹线段在时间、空间和编组关系方面都存在较大差异。在图9中,13号距离其所归属类簇的中心10号的时空距离和编组关系距离均较远,13号实际上应为离群轨迹线段。那么,13号被归入以10号为中心的类簇的原因是,众多类簇其实都

远离13号,以10号为中心的类簇只是这些类簇中离13号相对近一点的类簇而已。因此,13号这种离群轨迹线段,不应被划分到任何类簇,而应在轨迹聚类最终结果中单独显示。为了分离出该类离群轨迹线段,我们进行如下定义。

定义15 外围区域:若轨迹线段 L_i 属于某个类簇,但是与 L_i 相似度距离小于等于 d_c 范围内存在其他类簇的轨迹线段,则轨迹线段 L_i 属于外围区域。

定义16 离群阈值:外围区域的所有轨迹线段里,局部密度的最大值定义为离群阈值,记为 ρ_b 。

将各个类簇中局部密度小于 ρ_b 的轨迹线段从初步聚类结果中剥离出来,作为离群轨迹线段在轨迹聚类效果中单独显示。

2.4 聚类效果可视化

轨迹聚类的最后一个阶段就是可视化显示,该环节决定了轨迹聚类的最终效果。TRACUS算法没有有效利用类簇之间的关系,仅仅为每个类簇提取了一条代表轨迹,可视化效果难以全面精细反映机动轨迹情况。CTECW算法针对该问题进行了改进,分析了各个类簇之间的后继关系,并利用这些后继关系绘制出首尾相连的多个箭头,提升了聚类精度,突出了轨迹聚类的军事意义。但是CTECW算法计算后继关系的方法,容易造成多个类簇的“后继循环”现象,即“A类簇是B类簇的后继,B类簇是C类簇的后继,C类簇又是A类簇的后继”,这将导致聚类结果的混乱表达。本文提出的CTUW算法构建的类簇具有时间属性和编组属性,无须二次处理即可明确类簇之间的后继关系。

CTUW算法还能对非球形分布的数据样本进行聚类,因此,其聚类得到的轨迹线段类簇可能呈曲线形态。如图10所示,某群队的3个棋子呈纵队方式沿着某道路进行机动,该类簇中的轨迹线段发生时间和地理坐标相近,且均属于同一群队,因此,这批轨迹线段被聚为一类。

为了将轨迹类簇信息进行精细可视化,本文在完成聚类后,首先对各个类簇内轨迹线段进行二次DPC聚类,聚类算法依据的距离不再是相似度距离,而是空间距离,以完成对类簇内轨迹线段的二次空间切割,形成多个子类簇;再提取每个子类簇的代表轨迹,包含起点、方向、长度、时间范围和单位总称等信息,最后,绘制战斗经过图中的轨迹内容。

在描述可视化算法前,先明确几个概念。

定义17 子类簇:针对类簇内的所有轨迹线段,基于线段之间的空间距离,使用DPC算法对其进行聚类,得到的各个集合称为子类簇。

定义18 编组权重:轨迹线段关联棋子在编组结

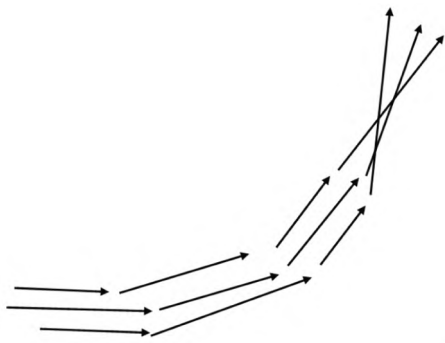


图 10 轨迹线段聚类后的某个类簇

Fig. 10 A cluster after track segment clustering

构中的权重,记为 γ 。顶层棋子编组权重为 1,下层棋子编组权重是上层棋子的一半,依此类推。如图 6 所示, H 棋子的编组权重是 1, G 棋子为 0.5, E 棋子为 0.25, A 棋子为 0.125。

定义 19 子类簇代表轨迹起点、方向和长度:假设 $D_s = \{L_i, L_{i+1}, \dots, L_j\}$ 为子类簇,其代表轨迹的起点 (x, y) 表示为 $x = \sum_{k=i}^j \omega_k x_{s_k}^k, y = \sum_{k=i}^j \omega_k y_{s_k}^k$,代表轨迹的方向和长度组成的向量表示为 $\sum_{k=i}^j \omega_k \vec{L}_k$ 。其中 \vec{L}_k 为轨迹线段 L_k 的向量, $\|L_k\|$ 为轨迹线段 L_k 的长度, (x_{s_k}, y_{s_k}) 为轨迹线段 L_k 的起点坐标, γ_k 为轨迹段 L_k 关联棋子的编组权重 ω_k 的计算公式为

$$\omega_k = \frac{(\|L_k\| \times g_k)}{\sum_{i=1}^j (\|L_i\| \times g_i)} \quad (10)$$

定义 20 子类簇代表轨迹的时间范围和单位总称:为了简洁描述聚类信息,子类簇代表轨迹的时间范围为子类簇中所有轨迹线段发生时间的并集,代表轨迹的单位总称分为简写和全写两种,简写总称为所代表棋子的最近共同祖先,全写为合并单位后的所有棋子名称。如图 6 所示,棋子 A, B, D 的单位总称简写为 G 单位,全写为 G 单位欠 I 单位。

基于以上定义,算法具体步骤如下:

1) 基于轨迹线段聚类得到的类簇,为每个类簇内部的轨迹线段进行二次 DPC 聚类形成若干个子类簇,聚类时依据的轨迹线段之间距离不再为相似度距离,而是空间距离;

2) 按照定义 18 和 19 计算每个子类簇的代表轨迹,包含起点、方向、长度、时间范围和单位总称等信息;

3) 将第 2) 步成果绘制成代表轨迹,将同一个类簇下的多个子类簇形成的代表轨迹进行美化,首尾相连的多个小箭头可以整合为一个弯曲大箭头;

4) 为第一次轨迹线段聚类过程中查找到的离群

轨迹线段绘制轨迹。

3 实验分析

为验证 CTUW 算法的实际效果,我们采集某款计算机兵棋推演系统的一次推演数据进行红方轨迹聚类,该组数据中包含 637 个红方棋子,下达了 4 328 条涉及机动行动的命令。如图 11 所示,分别展示了 CTUW 算法运行前的轨迹样本情况和执行算法后的聚类效果。因为使用的兵棋推演系统为六角格地图,原始轨迹锯齿状特性较明显,集中大量绘制锯齿状轨迹将干扰展示效果,所以图 11a) 展示的是基于原始轨迹进行轨迹压缩后形成的简化轨迹。

从 CTUW 算法的轨迹聚类效果来看,图 11c) 能够实现轨迹线段的聚类,并且在可视化表现上能够为代表轨迹赋予关联棋子总称和时间信息,能够支持机动态势的动态展示。进一步分析如下:

1) 在处理异常机动速度方面,位于图 11b) 的 D 区是炮兵群棋子由西向东间歇性机动轨迹,该群队原地开火一段时间后就转移一次阵地,因此,该群队的轨迹停留点比例较高,CTUW 算法不仅能剔除大量冗余的停留点信息,还能准确识别并聚类这种短停频率较高的轨迹。

2) 在处理特殊轨迹形态方面,图 11b) 的 E 区为右翼攻击群的机动轨迹, H 区为侦察队的机动轨迹,右翼攻击群在推演后半阶段分为两路的情况在图 11c) 中得到体现,零散、细碎、轨迹较多的侦察队在大多数聚类算法中容易被聚类到轨迹线段较长的 E 或者 G 区域,但是 CTUW 算法较好地将其聚为独立的一类。

3) 在轨迹交叉方面,图 11b) 中 A 区为左翼攻击群, B 区为纵深攻击群, C 区为推演前半阶段的左翼攻击群和推演后半阶段的纵深攻击群,从图 11c) 可以看到,CTUW 算法能够将 C 区域内空间重叠但时间不同的轨迹进行区分聚类;图 11b) 中 F 区为空军航空兵的航线轨迹,该轨迹与 E 区的右翼攻击群轨迹非常靠近,但是因为 CTUW 算法考虑了编组关系距离,所以没有错误地将其与右翼攻击群轨迹聚为一类;图 11b) 中 G 区为陆航队航线,该轨迹与 F 区域的后半段重合,但是 CTUW 算法依旧能将这两个群队的轨迹区分进行聚类。

因此,CTUW 算法不仅能够简洁地概括轨迹动态变化情况,还能同时满足轨迹精细聚类的要求。

从运行效率来看,CTUW 算法需要时刻维护和更新距离矩阵,算法运行测试数据样本的主要耗时集中在数据初始化阶段。因为该算法主要作为兵棋推演系统的构件发挥作用,所以实际应用该算法时能够直接

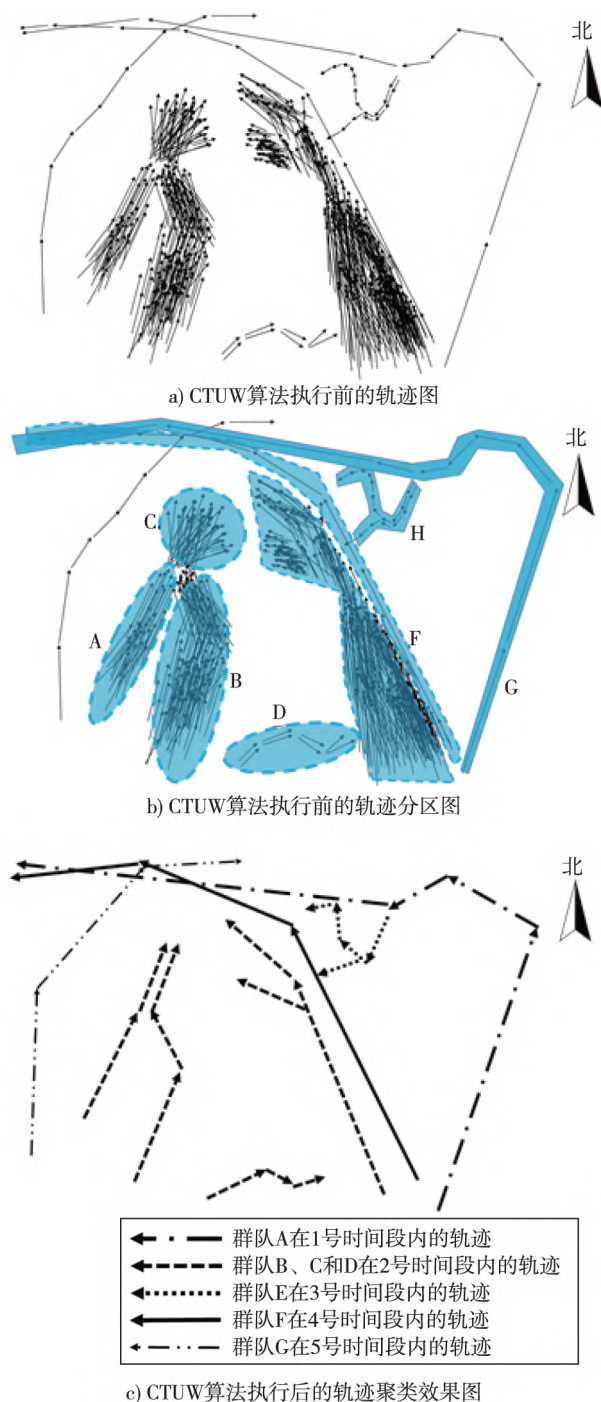


图11 不同算法执行后的轨迹聚类效果图

Fig. 11 Effect chart of trajectory clustering after implementation of different algorithms

读取已经存放于计算机内存中的推演数据,耗时最大的数据初始化环节无需考虑时间代价,轨迹压缩、轨迹线段聚类 and 可视化这3个阶段的算法复杂度分别为 n 、 $O(n \cdot \log(n))$ 和 $O(n \cdot \log(n))$,CTUW 算法的总体时间复杂度为 $O(n \cdot \log(n))$ 。

从算法泛化能力来看,虽然 CTUW 算法涉及时间、空间和编组关系等多维度的距离,但是算法能够适应六角格地图、矢量路网地图和普通地图上形成的多种

轨迹描述类型,还能兼容各类作战编组形式。CTUW 算法执行聚类前对3个维度的距离数据的归一化处理,也解决了各组兵棋推演数据的规模、时长和地幅面积差异性较大的样本问题^[19],而且 CTUW 算法采用的 DPC 聚类框架,无需先验知识,仅需人工设置1个阈值,降低了应用该算法的难度。另外,CTUW 算法关于时间、空间和编组关系等维度距离的权值,可由用户通过交互界面进行个性化设置,以满足不同轨迹聚类需求。

4 结束语

轨迹聚类技术不仅直接决定了兵棋推演系统等指挥训练系统的推演效率,还对指挥员基于信息系统进行作战指挥具有较大影响。本文基于时空和编组关系距离,引入 DPC 聚类算法框架,提出了满足军事应用需求的可视化显示方法,通过具体实例分析,验证了该轨迹聚类算法可行性,为指挥员和指挥机构高效组织兵棋推演发挥了重要作用。

参考文献:

- [1] 何昌其. 桌面战争: 美国兵棋发展应用及案例研究 [M]. 北京: 航空工业出版社, 2017.
HE C Q. Desktop war: development, application and case study of American war chess [M]. Beijing: Aviation Industry Press, 2017.
- [2] 胡晓峰, 罗批. 兵棋推演: 复杂系统管理的创新与实践 [J]. 指挥控制与仿真, 2016, 38(6): 1-5, 21.
HU X F, LUO P. Wargaming: innovation and practice for complex system management [J]. Command Control & Simulation, 2016, 38(6): 1-5, 21.
- [3] 石崇林, 淦文燕, 吴琳, 等. 计算机兵棋作战实体轨迹聚类算法 [J]. 软件学报, 2013, 24(3): 465-475.
SHI C L, GAN W Y, WU L, et al. Clustering trajectories of entities in computer wargames [J]. Journal of Software, 2013, 24(3): 465-475.
- [4] 李名, 胡勤友, 孟良. 基于 AIS 的船舶运动轨迹压缩技术研究 [J]. 航海技术, 2010(1): 11-13.
LI M, HU Q Y, MENG L. Research on ship motion trajectory compression technology based on AIS [J]. Marine Technology, 2010(1): 11-13.
- [5] DOUGLAS D H, PEUCKER T K. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature [J]. Cartographica: the International Journal for Geographic Information and Geovisualization, 1973, 10(2): 112-122.
- [6] LI Q N, ZHENG Y, XIE X, et al. Mining user similarity

- based on location history [C] // Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Irvine2008: 1-10.
- [7] 徐凯, 邱家瑜, 李燕. 一种加入时间维的船舶轨迹高效离线压缩算法研究 [J]. 计算机科学, 2017, 44 (S2): 498-502.
- XU K, QIU J Y, LI Y. Offline efficient compression algorithm for AIS data retains time elapsing dimension [J]. Computer Science, 2017, 44 (S2): 498-502.
- [8] 董婉婷, 于红, 周弈志, 等. 基于改进滑动窗口的渔船 AIS 轨迹数据压缩算法 [J]. 大连海洋大学学报, 2020, 35 (3): 462-468.
- DONG W T, YU H, ZHOU Y Z, et al. Automatic identification system (AIS) trajectory data compression algorithm of a fishing vessel based on improved Sliding Window [J]. Journal of Dalian Ocean University, 2020, 35 (3): 462-468.
- [9] KEOGH E J, CHU S, HART D, et al. An online algorithm for segmenting time series [C] // Proceedings of the 2001 IEEE International Conference on Data Mining, 2001: 289-296.
- [10] LEE J G, HAN J W, WHANG K Y. Trajectory clustering: a partition-and-group framework [C] // Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, Beijing2007: 593-604.
- [11] 郭名静, 边少锋, 单潮龙, 等. 面向多维时空位置数据的动态加权聚类模型 [J]. 测绘科学, 2019, 44 (11): 35-42.
- GUO M J, BIAN S F, SHAN C L, et al. Study on dynamic weighted clustering algorithm orienting spatio-temporal data mining [J]. Science of Surveying and Mapping, 2019, 44 (11): 35-42.
- [12] 龙真真, 张策, 王维平, 等. 一种基于数据流聚类的动态目标分群框架 [J]. 上海交通大学学报, 2010, 44 (7): 921-925.
- LONG Z Z, ZHANG C, WANG W P, et al. A dynamic framework for target-grouping based on clustering data streams [J]. Journal of Shanghai Jiao Tong University, 2010, 44 (7): 921-925.
- [13] 周超, 胡晓峰, 郑书奎, 等. 战略战役兵棋演习系统兵力聚合问题研究 [J]. 指挥与控制学报, 2017, 3 (1): 19-26.
- ZHOU C, HU X F, ZHENG S K, et al. Force integration in strategic and operational war-game maneuver system [J]. Journal of Command and Control, 2017, 3 (1): 19-26.
- [14] 高强, 张凤荔, 王瑞锦, 等. 轨迹大数据: 数据处理关键技术研究综述 [J]. 软件学报, 2017, 28 (4): 959-992.
- GAO Q, ZHANG F L, WANG R J, et al. Trajectory big data: a review of key technologies in data processing [J]. Journal of Software, 2017, 28 (4): 959-992.
- [15] 徐晓, 丁世飞, 丁玲. 密度峰值聚类算法研究进展 [J]. 软件学报, 2022, 33 (5): 1800-1816.
- XU X, DING S F, DING L. Survey on density peaks clustering algorithm [J]. Journal of Software, 2022, 33 (5): 1800-1816.
- [16] ESTER M, KRIEGEL H P, SANDER J, et al. A density-based algorithm for discovering clusters in large spatial databases with noise [C] // Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. Portland1996: 226-231.
- [17] RODRIGUEZ A, LAIO A. Clustering by fast search and find of density peaks [J]. Science (American Association for the Advancement of Science), 2014, 344 (6191): 1492-1496.
- [18] 吉根林, 王敏. 时空轨迹聚集模式挖掘研究进展 [J]. 南京师大学报(自然科学版), 2015, 38 (4): 1-7.
- JI G L, WANG M. Research progress of mining of gathering pattern in spatio-temporal trajectory [J]. Journal of Nanjing Normal University (Natural Science Edition), 2015, 38 (4): 1-7.
- [19] 刘水泉, 张长泉, 李增路, 等. 基于实体图层次划分法的兵力聚合方法 [J]. 火力与指挥控制, 2011, 36 (6): 128-130, 134.
- LIU S Q, ZHANG C Q, LI Z L, et al. A new force aggregation approach based on the entity map-layer analysis [J]. Fire Control & Command Control, 2011, 36 (6): 128-130, 134.

(责任编辑: 张培培)