



基于分组拍卖的集群机器人启发式图形构造算法

曲韵¹, 辛斌^{1,2,3*}, 王晴^{1,2}, 张钧溪¹, 郭苗¹

1. 北京理工大学自动化学院, 北京 100081;

2. 复杂系统智能控制与决策国家重点实验室, 北京 100081;

3. 北京智能机器人与系统高精尖创新中心, 北京 100081

* E-mail: brucebin@bit.edu.cn

收稿日期: 2021-08-04; 接受日期: 2021-11-26

国家自然科学基金青年基金项目(批准号: 62003044)资助

摘要 集群机器人的图形构造问题是指通过控制集群机器人的运动趋使其形成一个特定的图形. 集群机器人中的图形构造问题通常可以分解为两个子问题: 机器人与目标点之间的任务分配以及机器人与目标点之间的路径规划. 根据集群机器人图形构造问题规模大、易拥堵、碰撞的特点, 提出了一种集中优化、分组拍卖以及分布式交互相结合的OGADI(optimized grouping auction and distributed interaction)方法, 以缩短图形构造的完成时间. 将OGADI算法与最短路径集诱导顶点排序算法对比, 结果表明, 在集群机器人规模分别为500, 1000, 1300下, OGADI算法图形构造任务平均完成时间分别缩短了16.1%, 13.6%, 14.4%, 仿真验证了OGADI算法的可行性和有效性.

关键词 集群机器人, 图形构造, 拍卖, 分布式交互, 优化

1 引言

自然界的生物群体从其动态的自组织特性出发衍生出鱼群聚集、鸟群迁徙^[1]、蚂蚁搬家等行为, 并展现出强大的聚散现象. 群体相较于单体表现出更强大的环境适应性、更稳健的系统鲁棒性和更丰富的任务能力, 因此, 受自然界行为的启发学者们将集群行为开始应用于智能对抗^[2]、机器人自组装^[3~6]、信息中继^[7]等各个领域. 自然界提供了充足的证据证明稳健和自适应的群体构建是可能的, 研究学者们旨在赋予工程系统类似的特性.

图形构造问题^[8~11]作为集群行为中的一个经典问题被广泛研究, 主要通过相应的集群控制技术将集群

机器人编成期望的目标构型, 从而满足环境或者任务能力要求. 图形构造问题是一个组合优化问题, 旨在以较快的任务完成时间来实现目标构型. 通常来讲, 一个图形构造问题可以划分为两个子问题: 在目标构型中机器人的位置分配问题与集群机器人间的无碰撞路径规划问题. 在早期研究中, 学者们利用运筹学优化方法, 将图形构造中的优化问题转化为标准的非线性优化形式, 利用内点算法^[12,13]进行求解, 算法整体的时间复杂度较为明晰, 但其公式推导相对复杂, 不同约束下的算法可移植性较差. Haghighi和Cheah^[14]将图形进行分组, 并将机体划分为三种属性: 组间全局领航者、组内局部领航者以及跟随者, 同时利用领航跟随法为各机体设计相应的控制率, 精准的控制率在集群

引用格式: 曲韵, 辛斌, 王晴, 等. 基于分组拍卖的集群机器人启发式图形构造算法. 中国科学: 技术科学, 2022, 52
Qu Y, Xin B, Wang Q, et al. Heuristic shape formation algorithm in swarm robots based on group auction (in Chinese). Sci Sin Tech, 2022, 52, doi:
[10.1360/SST-2021-0349](https://doi.org/10.1360/SST-2021-0349)

控制精度上有较大的优势, 但整体任务的完成度对领航者的依赖性较大. Yu等人^[15,16]建立一个底层网络图, 通过集中式控制策略, 为各机器人分配相应的期望位置点, 并在智能体发生拥堵的过程中设计调度算法, 从而计算群体机器人的最优路径. Zhang等人^[17]提出了一种基于最优分配方案的启发式算法, 其将多个机器人分配给不同的目标点, 同时机器人根据其不同的通信范围, 调整分配策略. Jung和Kim^[18]设计了一种基于势函数的启发式群体图形构造方法, 通过设计参考点来吸引机器人向目标点运动, 同时利用虚拟点使得机器人逃离局部最小值点.

在上述的解决方案中, 集中式集群算法能够传递一个较优的解决方案, 但是对于数目较大的集群而言, 其物理可实现性受通信网络的限制且计算复杂度较高, 同时单个机体的损坏制约整个集群的任务效果. Izzo和Pettazzi^[19]提出了一种基于行为的路径规划算法, 利用聚集、避碰和停靠三个行为规则指导集群机器人完成期望图形构造. Bi等人^[20]设计了一种启发式的自组织图形构造算法, 主要利用人工势场法使得单个机器人自主决定其运动方向与运动速度, 同时通过动态均匀分布与期望最大化的方法使得机器人相对均匀地分布在给定位置区域附近, 但其松弛的拥堵消除机制使得大规模集群下的任务完成时间较长.

通过综合分析集中式与分布式算法的特点, 本文旨在设计一种集中式分配与分布式运动规划相结合的系统架构, 全局操控端对信息进行汇集并利用OGADI (optimized grouping auction and distributed interaction) 算法生成大规模图形构造问题的解, 单体机器人通过分布式避碰机制与交互规则自主决定其运动方向与运动速度, 机器人间通过分布式交互减少通信开销, 提升系统的容错率, 缓解智能体交互过程中的拥堵状况, 从而缩短图形构造任务完成的时间.

文章第二部分介绍了图形构造问题. 第三部分介绍了图形构造算法的原理. 为了进一步说明算法的有效性, 第四部分进行了仿真实验, 将OGADI算法与已有算法进行对比分析. 文章最后进行了总结与展望.

2 图形构造问题

如图1所示, 集群机器人的图形构造问题是指通过控制集群机器人的运动趋使其形成一个特定的图形.

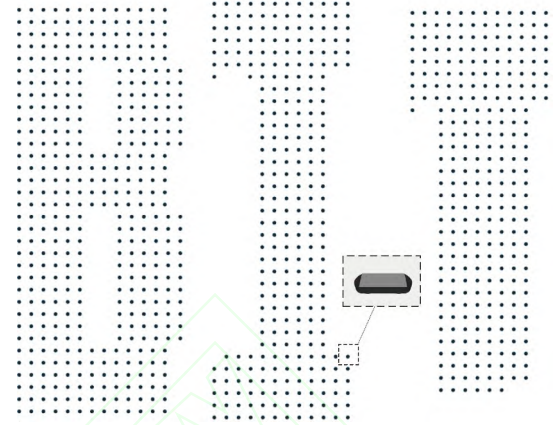


图1 图形构造示例

Figure 1 A shape formation example.

在图形构造的过程中, 需要规划集群机器人的位置和路径, 同时要考虑其运动学限制、安全距离及运动代价. 大规模图形构造问题的难点在于如何快速完成图形构造任务并避免机器人之间的碰撞.

在本文中, 一组先验的目标点表示期望图形, 利用分组拍卖来构造图形任务的解(即一个机器人匹配一个目标点), 并通过优化分组拍卖中的参数进一步优化机器人与目标点的匹配关系, 最后, 在集群机器人运动过程中利用分布式交互机制进行避障.

假定可行区域 M 中有 N 个机器人, N 个机器人的状态向量为

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N], \quad (1)$$

其中, \mathbf{a}_i 为第 i 个机器人的状态向量, 记为

$$\mathbf{a}_i = [p_i, v_i], \quad (2)$$

式中, p_i 为第 i 个机器人的绝对位置坐标, v_i 为第 i 个机器人的运动速度.

首先, 第 i 个机器人根据其速度上限(v_{\max})与避障约束对下一时刻机器人的期望运动速度(v_{id})加以修正, 其次, 根据修正后的运动速度与下一时刻机器人的运动方向角(θ_i)解算出下一时刻机器人的绝对位置坐标, 因此, 第 i 个机器人的时不变离散状态方程表示为

$$\begin{cases} \mathbf{a}_i(k+1) = f(\mathbf{a}_i(k), v_{id}, \theta_i), \\ \mathbf{a}_i(k) \in M, \\ v_i \leq v_{\max}, \\ i = 1, 2, \dots, N, \\ k = 1, 2, \dots, t_i, \end{cases} \quad (3)$$

式中, 非线性映射 $f(\mathbf{a}_i(k), v_{id}, o_i)$, 根据 k 时刻第 i 个机器人的状态向量与 $k+1$ 时刻第 i 个机器人的期望运动速度与方向角, 映射到 $k+1$ 时刻的状态向量, t_i 为第 i 个机器人的任务完成时间. 图形构造任务的完成时间为最后一个机器人到达任务点所用的时间, 记为

$$t = \max\{t_i\}, i = 1, 2, \dots, N. \quad (4)$$

本文通过设计OGADI算法, 将机器人的运动过程离散化, 根据当前时刻机器人的状态信息与约束条件, 给出机器人下一时刻的状态信息. 机器人运动满足以下条件及约束:

(1) 机器人具备全向运动的能力;

(2) 将机器人运动过程离散化, 给出机器人下一时刻的位置点, 考虑到其物理可实现性, 在运动学层面对机器人的运动速度加以约束, 使得机器人可以从当前状态运动到下一时刻的位置点.

3 图形构造算法

针对大规模、低时延、快速响应的图形构造任务的需求, 本文设计了一种OGADI算法用以解决图形构造问题. 图形构造算法的框架如图2所示, 首先初始化参数, 包括机器人的绝对坐标, 集群机器人分组数量, 目标图形(一组先验的目标点)等参数. 其次, 利用分组拍卖构造与目标图形构型相对应的解(即每个智能体与目标点的匹配关系, 一个机器人只能匹配一个目标点), 并判断是否满足收敛条件, 若满足收敛条件, 机器人向各自的目标点运动, 并利用分布式交互机制进行合理避障; 若不满足收敛条件, 离线评估OGADI算法生成的解, 通过优化分组拍卖算法中的参数进一步优化智能体与目标点的匹配关系.

3.1 OGADI算法

该算法利用分块处理、分组拍卖、局部交互的方式来解决图形构造问题. 为了更好地说明解的生成规则, 利用一个示例来阐述OGADI算法, 其中方阵图形为当前 n 个机器人所在的位置, 圆环图形为目标图形构型, OGADI算法的流程图如图2所示, 具体包括以下几个步骤.

步骤1, 一次分类, 在集中决策时, 机器人处于全连通的状态, 如图3所示, 利用 k -means算法^[21]对 N 个机器

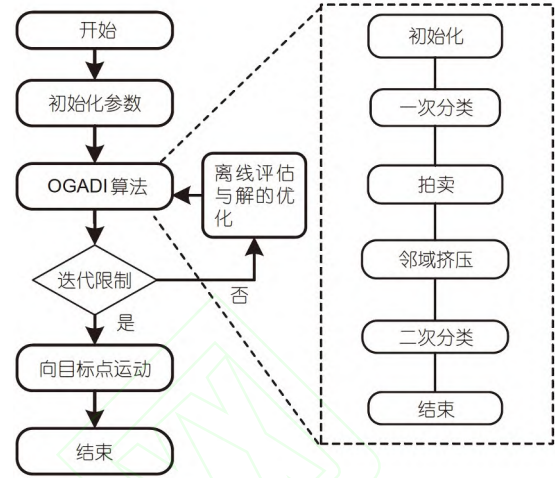


图2 图形构造算法流程图

Figure 2 Flowchart of shape formation algorithm.

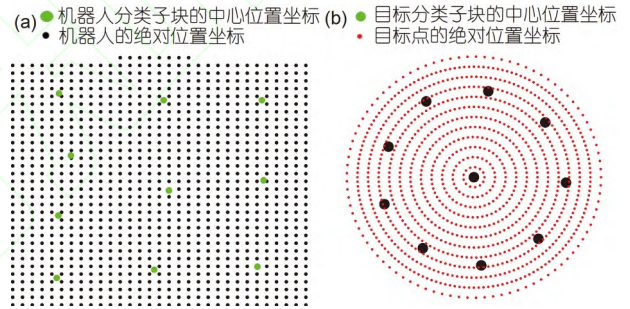


图3 一次分类. (a) 机器人的一次分类结果; (b) 目标点的一次分类结果

Figure 3 A classification. (a) A classified result of the robot; (b) a classified result of the target.

人(方阵图形)的绝对位置坐标和目标图形(圆环图形)的绝对位置坐标进行分类, 将 n 个机器人分为 m 类, 并根据分类结果分别得到 m 个子块的中心位置坐标, 记为 b_{cu} (简称子块, 图3中的绿色点)与各目标图形子块的中心位置坐标, 记为 b_t (简称目标子块, 图3中的黑色点), 计算 n 个机器人到 m 个子块的距离 d_{ui} ,

$$d_{ui} = \|b_{cu} - p_i\|, i = 1, 2, \dots, N, u = 1, 2, \dots, m, \quad (5)$$

式中, b_{cu} 为第 u 个子块的中心位置坐标. 在图4中, 不同颜色表示不同中心位置所包含机器人的情况, 将距离各子块(b_{cu})较近的机器人划分到集合 s_u 下:

$$s_u \leftarrow \min\{d_{ui}\}, u = 1, 2, \dots, m, \quad (6)$$

式中, s_u 为第 u 个子块(b_{cu})所包含的机器人数目、机器

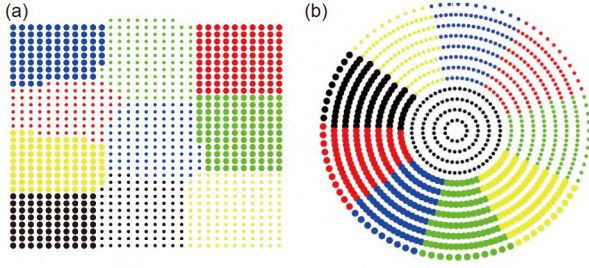


图4 不同中心点所包含的机器人划分。(a) 各中心点所包含机器人情况; (b) 各中心点所包含目标点情况

Figure 4 Classified result of the different centers. (a) Robot contained in each central point; (b) target contained in each central point.

人绝对位置坐标与机器人标识(如图4中左侧方阵颜色划分情况), 同理得到 g_u , g_u 为第 u 个目标子块(b_{iu})分别所包含的目标点数目、目标点绝对位置坐标与目标点标识(如图4中右侧圆环颜色划分情况)。

步骤2, 拍卖, 将各子块(b_c)与各目标子块(b_i)之间利用拍卖算法建立配对关系, H_{ul} 为第 u 个子块(b_{cu})与第 l 个目标子块(b_{il})之间的代价, 记为

$$H_{ul} = E_{ul} + c_{ul}, \quad (7)$$

式中, E_{ul} 为第 u 个子块(b_{cu})与第 l 个目标子块(b_{il})之间的欧式距离, c_{ul} 为第 u 个子块(b_{cu})与第 l 个目标子块(b_{il})之间的累积值。第 u 个子块(b_{cu})的出价更新规则如下:

$$p_{ru} = p_{ru} + \min\{H_{ul}\} - \minp\{H_{ul}\}, \quad l = 1, 2, \dots, m, \quad (8)$$

式中, $\minp\{H_{ul}\}$ 为代价(H_{ul})的次小值, l_{\min} 为代价最小值($\min\{H_{ul}\}$)所对应目标子块的下标。出价最高的子块($\max\{p_{ru}, u = 1, 2, \dots, m\}$)获胜, 获胜子块与第 l_{\min} 个目标子块($b_{il_{\min}}$)建立匹配关系, 在当前轮次中, 只更新第 l_{\min} 个目标子块($b_{il_{\min}}$)与 m 个子块(b_c)所对应的累积值($c_{ul_{\min}}$), 记为

$$c_{ul_{\min}} = H_{ul_{\min}} + \max\{p_{ru}\}, \quad u = 1, 2, \dots, m. \quad (9)$$

每一轮竞价中, 只有未建立匹配关系的子块出价。如图5所示, 有3个子块(1, 2, 3)与3个目标子块(A, B, C), 在第一轮竞价中, 子块1与目标子块B配对; 在第二轮竞价中只有子块2与子块3参与竞价, 且子块2与目标子块A配对; 在第三轮竞价中, 子块3与目标子块B配对, 则子块1置空; 在第四轮竞价中所有子块都成功配对(即每个子块只与一个目标子块配对), 视为拍卖结束。

步骤3, 邻域挤压, 根据拍卖结果, 依次对各子块(b_c)中包含的机器人数目进行调整, 即目标子块(b_i)中机器人数目保持不变, 调整子块中机器人的数目, 将各子块(b_c)中需要减少机器人数目的子块集合记为 L_b 、需要增加机器人数目的子块集合记为 M_b 、不需要调整机器人数目的子块集合记为 N_b , L_b 按照调整路线向 M_b 调整各子块所包含的机器人个数, 迭代终止条件为配对的各子块与各目标子块之间所包含的数目相等。如图6所示, 图中蓝色子块通过邻域挤压的方式调整蓝色子块所包含的智能体数目, 图6中的右侧给出了蓝色子块与绿色子块之间调整完成后的局部放大图。

具体的调整方式如下。

(1) 调整目标, L_b 中第 s 个子块(L_{bs})分别计算与 M_b 中所有子块的欧式距离:

$$E_{sg} = \|L_{bs} - M_{bg}\|, \quad g = 1, 2, \dots, G, \quad (10)$$

式中, G 为 M_b 中所包含子块的个数。 L_{bs} 向目标子块 M_{ba} 调整机器人的数目, M_{ba} 可表示为

$$M_{ba} \Leftarrow \min\{E_{sg}\}, \quad g = 1, 2, \dots, G. \quad (11)$$

(2) 调整数目, L_{bs} 所需减少机器人的数目记为 A_{num} , A_{num} 为两者相比(L_{bs} 与 M_{ba})所包含机器人数目较少的一方。

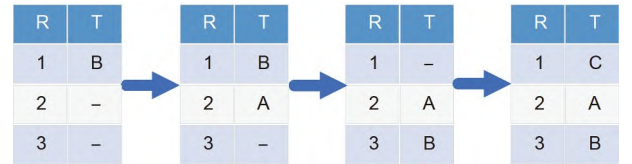


图5 拍卖流程
Figure 5 Auction algorithm.

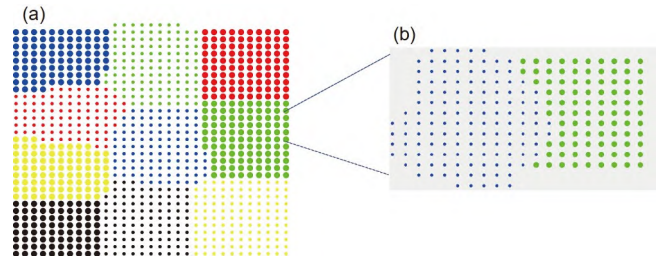


图6 邻域挤压。(a) 邻域挤压前; (b) 邻域挤压后局部放大图
Figure 6 Process of the neighborhood squeeze. (a) Robot classification before the neighborhood squeeze; (b) the local enlargement of the classified result after the neighborhood squeeze.

(3) 调整路线, L_{bs} 与调整目标子块(M_{ba})的欧式距离记为 d_{LM} :

$$d_{LM} = \|L_{bs} - M_{ba}\|. \quad (12)$$

计算 L_{bs} 与 b_c 中除 L_{bs} 外其余子块之间的欧式距离, 并将距离值从小到大排序, 存储在排序列表中, 排序列表中最小值对应的子块为 b_{\min} :

$$b_{\min} \leftarrow \min\{L_{bs} - b_{cu}\}, u = 1, 2, \dots, m. \quad (13)$$

b_{\min} 与调整目标子块(M_{ba})之间的欧式距离为

$$d_{\min} = \sqrt{(b_{\min} - M_{ba})^2}. \quad (14)$$

若 $d_{LM} \geq d_{\min}$, 将 b_{\min} 的中心坐标放入调整路线中, 并将 L_{bs} 所包含机器人的绝对位置坐标与子块 b_{\min} 的中心坐标之间欧式距离最小的前 A_{num} 个机器人调整给子块 b_{\min} , 同时 L_b 中当前需要减少机器人数的子块位置更新为 b_{\min} 的中心位置; 若 $d_{LM} < d_{\min}$, 则将排序列表中当前最小的欧式距离值删除, d_{\min} 记为排序列表中新的最小欧式距离值, 重复判断与 d_{LM} 的大小, 直到 b_{\min} 为调整目标子块(M_{ba}), 视为 L_b 中第 s 个子块(L_{bs})调整完成. 重复邻域挤压规则, 直到配对的各子块与各目标子块之间所包含的数目相等.

步骤4, 二次分类, 在各子块内将机器人分为 m_1 类, 各子块所包含的机器人数量记为 B_n , 对各子块重复进行步骤1、步骤2与步骤3, 若 $m_1 \geq B_n$, 则当前子块中 m_1 修正为 B_n , 将当前子块中所包含的各机器人拍卖给当前子块所对应的目标子块中的各目标点, 即完成当前子块的任务分配; 若 $m_1 < B_n$, 则将当前子块中所包含的机器人分为 m_1 类, 对 m_1 类中的机器人与目标点利用贪婪算法^[22]进行任务分配.

3.2 粒子群优化

通过优化启发式算法中的分类数目(m, m_1)进一步优化机器人与目标点的匹配关系. 本文设计了适应值函数 $f(m, m_1)$, 记为

$$f(m, m_1) = far(m, m_1) / d_{\max}, \quad (15)$$

式中, $far(m, m_1)$ 表示机器人与目标点之间最远的欧式距离, d_{\max} 表示机器人在当前环境下可能分配的最远欧式距离. 适应值函数 $f(m, m_1)$ 用来评价解的质量, 并利用粒子群优化算法^[23]对解进行优化. 变量 m 与 m_1 的

速度和位置更新公式如下:

$$v_{m_{n,d}}^{e+1} = w * v_{m_{n,d}}^e + C_1 * rand_1 * (pbest_{n,d}^e - x_{m_{n,d}}^e) + C_2 * rand_2 * (nbest_{n,d}^e - x_{m_{n,d}}^e), \quad (16)$$

$$x_{m_{n,d}}^{e+1} = x_{m_{n,d}}^e + v_{m_{n,d}}^e, \quad (17)$$

$\mathbf{x}_{m_n}^e = [x_{m_{n,1}}^e, x_{m_{n,2}}^e, \dots, x_{m_{n,D}}^e]^T$ 和 $\mathbf{v}_{m_n}^e = [v_{m_{n,1}}^e, v_{m_{n,2}}^e, \dots, v_{m_{n,D}}^e]^T$ 分别表示 d 维搜索空间中第 n 个粒子在其第 e 次迭代时的位置和速度向量, $\mathbf{pbest}_n^e = [pbest_{n,1}^e, pbest_{n,2}^e, \dots, pbest_{n,D}^e]^T$ 表示第 n 个粒子在前 e 次迭代过程中个体发现的最佳位置, $\mathbf{nbest}_n^e = [nbest_{n,1}^e, nbest_{n,2}^e, \dots, nbest_{n,D}^e]^T$ 表示第 n 个粒子所在邻域中所有粒子在前 e 次迭代过程中找到的最佳位置, 加速度系数 C_1 与 C_2 设置为2, 惯性因子 w 设置为0.5, $rand_1$ 与 $rand_2$ 表示区间 $[0, 1]$ 上的随机数, 变量个数 $n=2$, 维度 $D=10$, 种群进化代数40代, 通过100次仿真实验得出 m 与 m_1 分类数目均小于40, 且分组数量的增加带来的时间开销较大, 因此, m 与 m_1 的搜索空间为 $[2, \min(\sqrt{N}, 40)]$, N 为机器人的个数.

3.3 分布式交互

在上述OGADI算法生成的解满足迭代收敛条件后, 各机器人利用人工势场法^[24]向目标点运动, 根据总势场的大小和梯度方向决定机器人下一时刻的运动方向. 由于人工势场法存在死锁问题, 因此在分布式交互过程中, 设置信息交互规则, 使得各机器人在避障的同时避免出现死锁问题.

对于当前机器人而言, 本文将感知到的其余机器人视为作业环境中的动态“障碍物”, 具体的信息交互规则如下.

(1) T_{a_i} 为当前机器人 a_i 的目标位置, 其在感知半径 r 的范围内感知到的机器人数量为 $j = 0, 1, 2, \dots, q$, 机器人 a_j 的目标位置记为 T_{a_j} , d_1 表示无目标交换下的运动代价, 记为

$$d_1 = \sqrt{(p_i - T_{a_i})^2} + \sqrt{(p_j - T_{a_j})^2}. \quad (18)$$

d_2 表示目标交换下的运动代价, 记为

$$d_2 = \sqrt{(p_i - T_{a_j})^2} + \sqrt{(p_j - T_{a_i})^2}. \quad (19)$$

如图7所示, 两条红色虚线之和为 d_1 , d_2 为两条黑色虚线之和, 由于三角形两边之和大于第三边, 因此若两条红色虚线之间有交点(同理适用于两虚线共线), 则 $d_1 > d_2$.

当前机器人 a_i 按照感知距离对 j 个机器人进行排序, 按照距离从近到远开始对 j 个机器人进行遍历, 若 $d_1 > d_2$, 交换第 i 个机器人与第 j 个机器人的目标位置, 机器人局部信息交互后, 各机器人的目标排布如图8所示.

(2) 考虑到机器人运动速度上限与机器人的安全

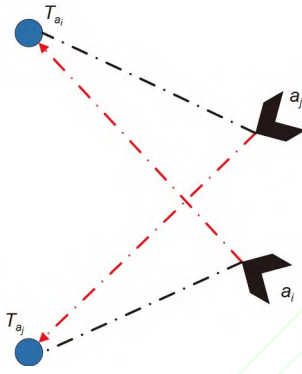


图7 智能体目标交换

Figure 7 Targets exchange between the different agents.

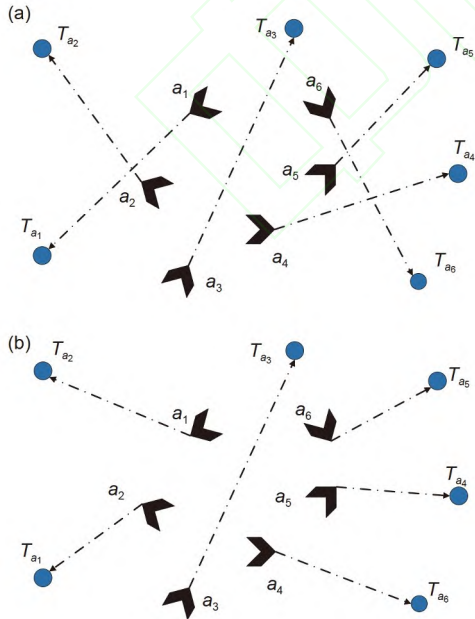


图8 局部信息交互. (a) 信息交互前; (b) 信息交互后

Figure 8 Local information interaction between agents. (a) Before the local information interaction; (b) after the local information interaction.

距离, 在运动过程中机器人的运动速度 v_i 记为

$$v_i = \min(0.4 * ps, v_{imax}), \quad (20)$$

其中, ps 为当前机器人感知的最短欧式距离.

4 算法验证

为了充分说明算法的可行性和有效性, 测试算例包含不同的机器人规模、图形尺寸、图形类别. 同时将OGADI算法分别与最短路径集诱导顶点排序算法(SPSIVO)^[15]、势函数算法(APF)^[18]、自组织图形构造算法(DUDEM)^[20]进行对比, 并用IntelliJ IDEA仿真软件进行对比算法复现, 利用AnyLogic仿真软件对本文所提的算法进行了仿真验证, 所有实验在Inter(R)Core(TM)i5-10210U 1.60GHz CPU环境中进行, 同时利用显著水平为0.05的秩和检验两种算法之间的差异性. 在下述仿真数据中集群规模的取值为{500, 1000, 1300, 1500}, 目标图形有组合字母和均匀点阵. 如图9所示, 1000个机器人从均匀点阵切换到字母B所用时间为133 s.

在下述图形图表中, 为了方便记录, 用机器人当前的数量与目标队形间距来表示当前测试算例, 比如1000_5, 表示当前测试算例中有1000个机器人, 目标图形中目标点之间的间距为5 m.

为了测试启发式算法的有效性, 分别在同一机器人数量与目标队形间距下设置了20组随机生成的测试算例, 如图10所示, 蓝色和橘色分别表示OGADI算法和SPSIVO算法图形构造任务的完成时间. 在SPSIVO算法中单位格子的划分与机器人的任务完成时间无关, 其任务完成时间与机器人的导航速度有关, 在随机初始化过程中每两个机器人的间距大于等于1 m, 因此在对比算法中一个格子长度为1 m, 机器人的运行速度设置为1 m/s, 在OGADI算法中机器人的运行速度上限为1 m/s. 从图10中可知, 在500_6, 1000_5, 1300_4集群机器人规模下, OGADI算法平均任务完成时间分别为109, 139, 124 s, SPSIVO算法平均任务完成时间分别为130, 161, 145 s, OGADI算法任务平均完成时间分别缩短了16.1%, 13.6%, 14.4%. 在图10中, SPSIVO算法在三种不同规模下任务完成时间的峰值分别为178, 214, 210 s, OGADI算法对应的峰值分别为141, 154, 147 s. 在同一工作空间中, 目标图形的尺寸影响着集群机器

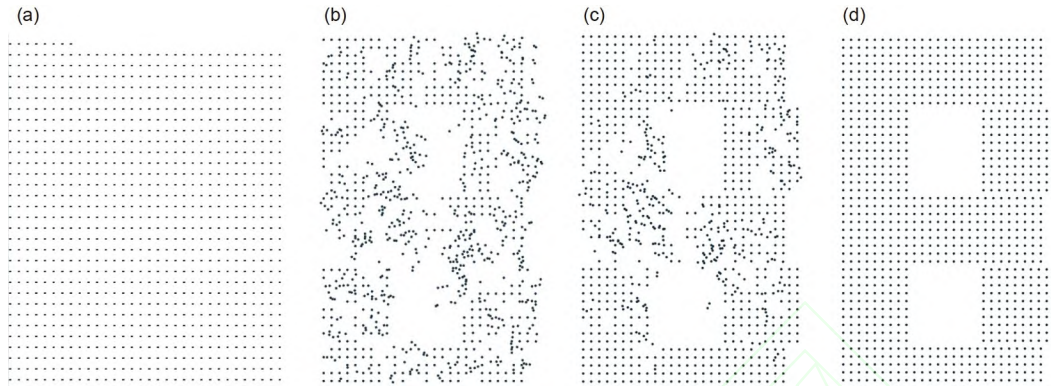


图9 1000个智能体从点阵变换到字母B. (a) $T=0$ s; (b) $T=20$ s; (c) $T=48$ s; (d) $T=133$ s

Figure 9 Shape formation of the letter B in 1000 agents. (a) $T=0$ s; (b) $T=20$ s; (c) $T=48$ s; (d) $T=133$ s.

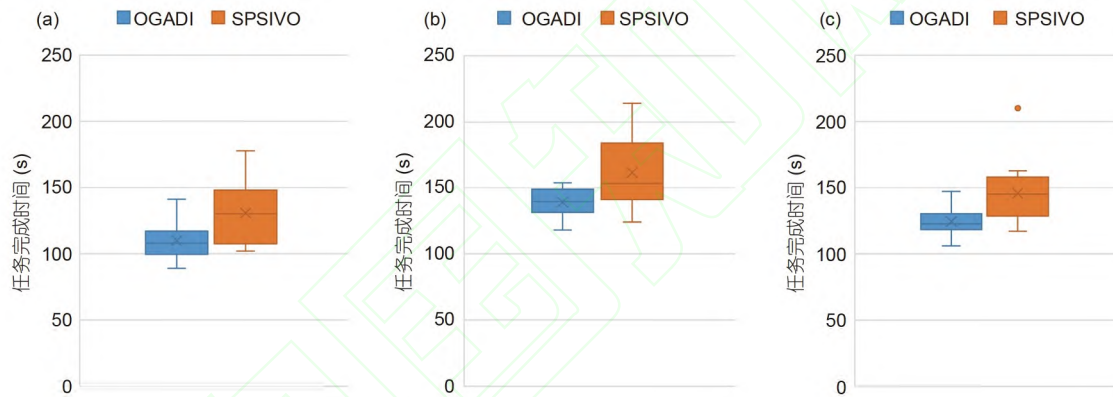


图10 不同智能体规模下图形构型完成时间. (a) 500_6; (b) 1000_5; (c) 1300_4

Figure 10 Completion time of the shape formation under different agent scales. (a) 500_6; (b) 1000_5; (c) 1300_4.

人的任务完成时间,随着机器人数的增多,目标图形的尺寸逐渐减小,任务完成时间增加较为缓慢.在图10中OGADI算法在三种不同规模下任务完成时间的标准差分别为12.8, 11.3, 10.4, SPSIVO算法对应的标准差分别为23.2, 27.1, 22.4.从图10的数值波动范围可以看出,OGADI算法的任务完成时间较短,且对初始化参数的敏感度较低.由于OGADI算法对多机器人进行分类处理与分组拍卖,组内中机器人保持整体的运动趋势,同时在小组中其通过目标引导与局部交互的方式趋于目标点,进而能降低随机初始化位置对算法性能的影响.从集群机器人数量的角度分析,随着机器人数量的增加,OGADI算法较SPSIVO算法而言任务整体完成时间较为稳定,同时SPSIVO算法随着机器人数的增多,任务分配计算时间较长.

在OGADI算法中,分组拍卖为大规模集群机器人划分了不同的运动区域与任务点,为了进一步提高解的质量,利用优化算法对分组拍卖中的分组数量进行优化.因此为了进一步分析OGADI算法中分组拍卖和优化算法的作用效能,通过将OGADI算法分别与APF算法、DUDEM算法进行对比,同时在表1中设计了只采用本文所提算法中的分组拍卖而不采用优化算法的仿真实验.通过20组测试算例来统计不同算法下图形构造问题的平均完成时间,算例中分别包含了均匀分布的字母O与字母A.

从表1中可以看出APF算法平均任务完成时间较长,如图11所示,随着任务点数目的增多,该方法从图形非均匀化到图形均匀化的过程较长,由于其逃离局部最小值的时间较长,且算法的完成时间依赖于参考

表 1 图形构造问题在不同算法中的任务完成时间(单位: s)

Table 1 Task completion time of the shape formation problem in different algorithm (unit: s)

机器人数量	APF ^[18]	DUDEM ^[20]	分组拍卖	OGADI
50	840	121	112	90
100	1272	260	132	101
200	1584	576	138	122
300	1902	860	287	254
400	2206	1100	454	424
500	2280	1162	645	610

点的选取. DUDEM算法任务完成时间相较于APF算法较快, 如图12所示, 虚线框为图形构造任务的给定区域, 由于其利用的动态均匀分布与期望最大化的方法使机器人相对均匀地分布在给定位置区域附近, 进一步缓解了APF算法中机器人陷入局部最小值的情况. DUDEM方法较OGADI算法而言其任务完成时间较长, 由于DUDEM算法中利用期望最大化方法使得机器人分布相对均匀, 但随着机器人数的增多, 机器人从非均匀区域向均匀区域挤压的过程较长. OGADI算法与DUDEM算法相比, 由于分组拍卖算法起到了降低维度的作用, 使得集群机器人在组内保持整体运动的趋势, 同时通过目标引导与局部交互的方式趋于目标点, 分组拍卖算法缩短了56.6%的任务完成时间, 优化算法在分组拍卖的基础上, 进一步优化分组数目, 进而能降低随机初始化位置对算法性能的影响, 进一步缩短了3.54%的任务完成时间.

为了测试阵型间距对启发式算法的影响, 如图13所示, 横轴表示10个不同的随机种子, 纵轴表示1500个机器人在三种点阵间距下的图形构造完成时间, 机器人的运行速度上限为3 m/s. 从图中三种曲线走势可以看出随着目标图形中目标点之间间距的增加, 机器人整体任务完成时间呈上升趋势. 从单个曲线走势中可以看出在机器人数量相同的情况下, 目标图形间距较大的图形构造任务完成时间波动范围较大, 可能是由于在启动避障算法时对机器人的步长进行了限制, 随着目标间距变大, 机器人在避障过程中运动步长受限, 进而延长了整体的任务完成时间.

为了进一步说明优化算法对启发式解的作用, 图14给出了4种不同机器人规模下的启发式解的优化情况. x 轴表示迭代次数, y 轴表示当前迭代次数下的适应

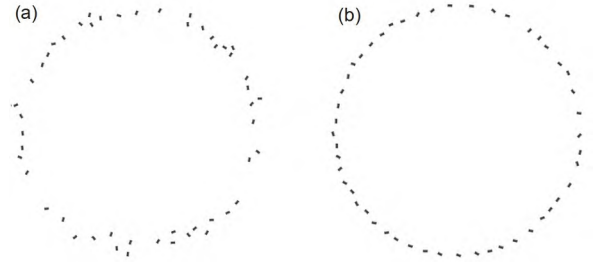


图 11 APF算法实现图形构造任务. (a) 非均匀化; (b) 均匀化

Figure 11 APF algorithm is applied to complete the shape formation. (a) Inhomogeneous; (b) homogeneous.

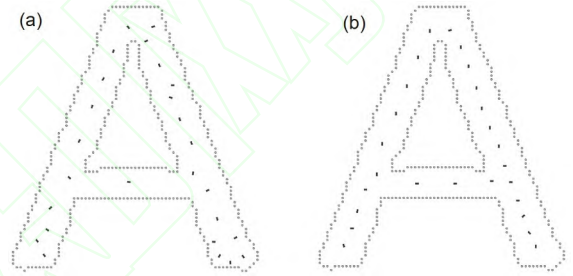


图 12 DUDEM算法完成图形构造任务. (a) 非均匀化; (b) 均匀化

Figure 12 DUDEM algorithm is applied to complete the shape formation. (a) Inhomogeneous; (b) homogeneous.

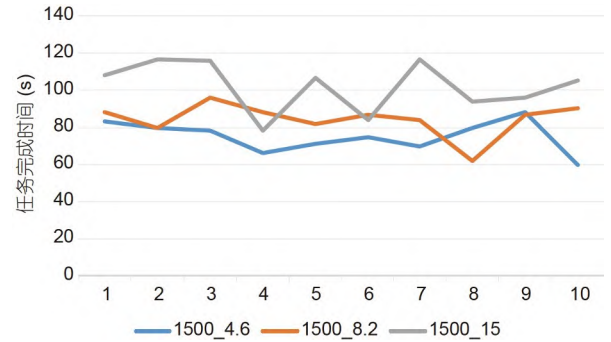


图 13 1500个机器人三种点阵间距的图形构造任务完成时间

Figure 13 Shape formation task completion time of 1500 robots at three lattice spacing.

值函数. 种群中个体数为10, 迭代次数为40代, 从图14中可以看出PSO算法在迭代初期收敛到最优解附近, 随着迭代代数的增加, PSO算法能够继续寻优, 并最终收敛.

前面已经分析了算法在1与3 m/s速度上限情形下的任务完成时间, 为了更好地说明所提算法在不同运

行速度上限情形下任务的可行性, 图15给出了10 m/s速度上限环境中不同机器人规模下各机器人与目标点之间的偏差曲线。如图15所示, 图形构造算法在500,

1000, 1300个不同机器人规模下其任务完成时间分别为99, 130, 107 s。从图中可以看出, 随着时间步长的增加, 图形构造任务整体的偏差呈下降趋势, 集群机器人

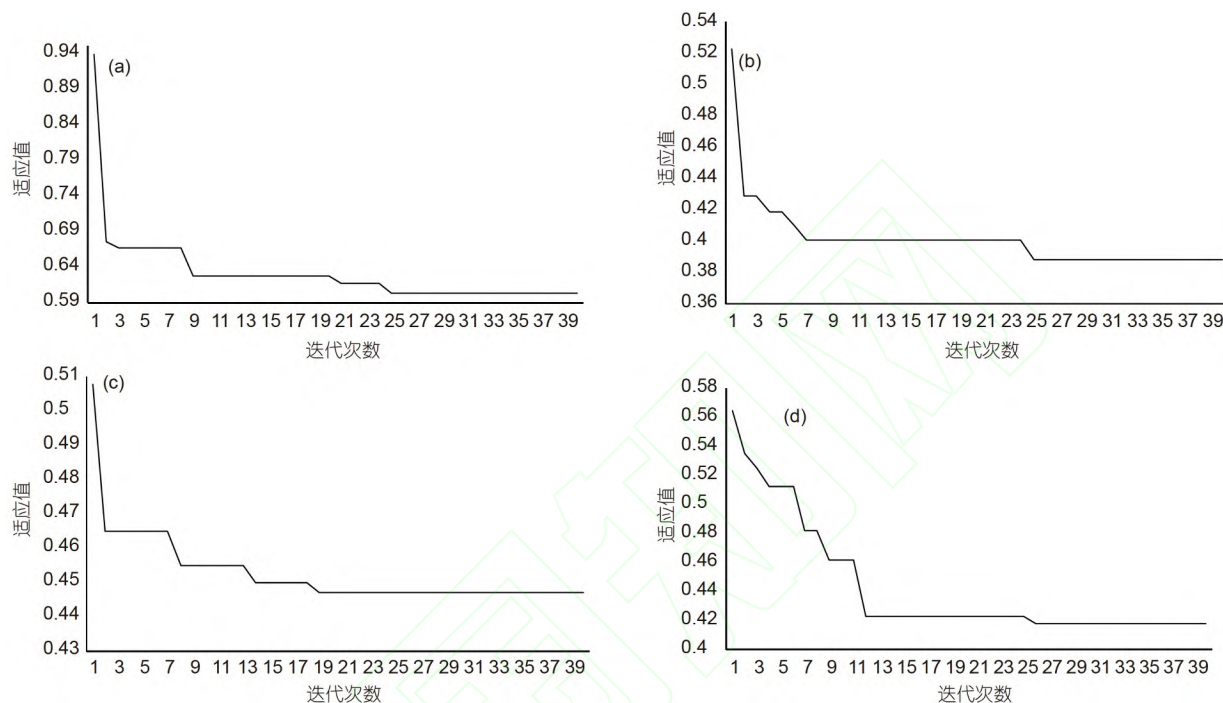


图 14 不同机器人规模下的适应值函数变化. (a) 500_4.5; (b) 1000_4.5; (c) 1300_4.5; (d) 1500_4

Figure 14 Curve of fitness function changes under different robot scales. (a) 500_4.5; (b) 1000_4.5; (c) 1300_4.5; (d) 1500_4.

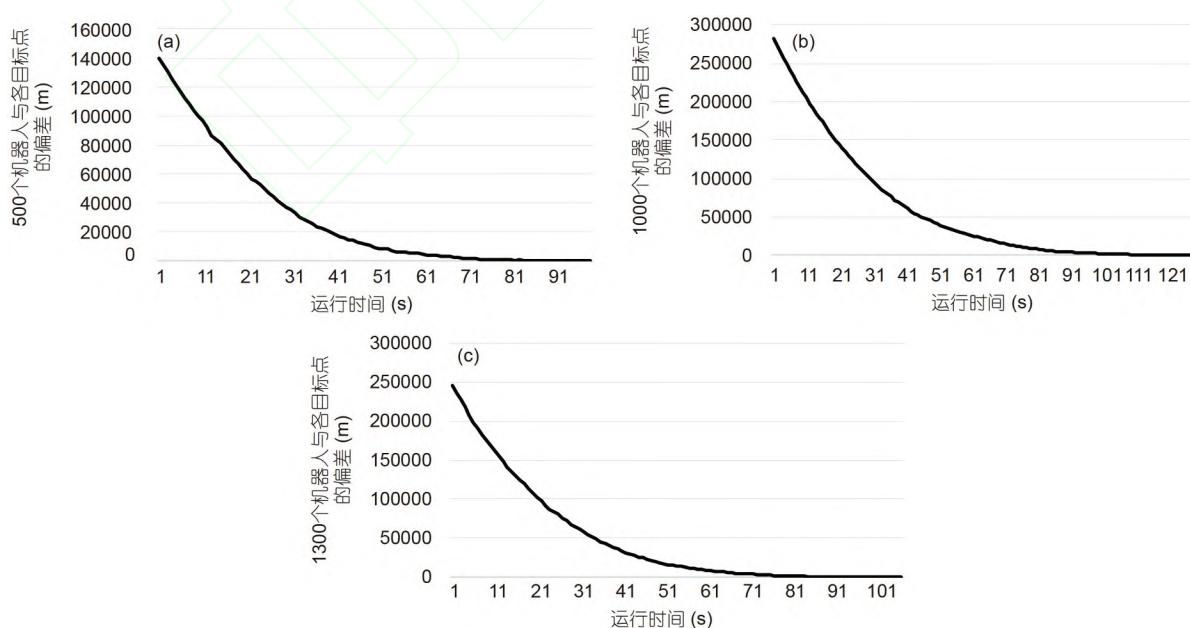


图 15 不同机器人规模下的各机器人与目标点之间的偏差和. (a) 500个机器人; (b) 1000个机器人; (c) 1300个机器人

Figure 15 Deviation between each robot and the corresponding target at different robot scales. (a) 500 robots; (b) 1000 robots; (c) 1300 robots.

朝着各自的目标点运动, 整体偏差最终趋于0.

两种算法在显著水平0.05的秩和检验下的计算结果如表2所示, (+)/(≈)/(-)分别代表启发式算法显著优于, 统计学无显著差异, 显著差于对比算法.

从表2中可以看出, 在下述测试算例中启发式算法显著优于对比算法, 这些测试算例的变量包含了不同的机器人数量, 不同的图形间距. 相比较而言, 针对大规模算例, 启发式算法通过预处理的方式在一定程度上降低了机器人之间的拥堵状态, 从而缩短了整体任务的完成时间, 因此启发式算法的性能表现较好.

5 结论

本文利用启发式方法将大规模图形构造问题进行分解, 通过分块处理、分组拍卖的方式形成初步的预分配方案, 使得子块内的机器人形成一个整体的运动趋势, 通过优化算法对预分配方案进行评估, 并进一步利用分布式避碰机制与交互规则对预分配方案进行

表 2 两种算法在测试算例下的差异性

Table 2 Differences of the two algorithms in the test examples

算例	p 值
500_6	$3.8 \times 10^{-3}(+)$
500_10	$3.64 \times 10^{-3}(+)$
1000_4	$3.64 \times 10^{-2}(+)$
1300_4	$1.99 \times 10^{-3}(+)$

修正. 实验结果表明, 在大规模图形构造问题中, 启发式算法任务平均完成时间与最短路径集诱导顶点排序算法任务平均完成时间相比, 在500_6, 1000_4, 1300_4集群机器人规模下分别缩短了16.1%, 13.6%, 14.4%.

需要指出的是, 本文的问题模型主要考虑了集群机器人在运动步长约束的情况下如何有效地完成图形构造并避免碰撞. 在实际的运动过程中, 可以在避碰机制中加入耗散项, 进一步将离散的运动步长连续化处理. 在未来的研究中, 为了进一步强化无人机的灵活性与容错性, 考虑利用纯分布式算法实现图形构造.

参考文献

- 1 Qiu H X, Duan H B. From collective flight in bird flocks to unmanned aerial vehicle autonomous swarm formation (in Chinese). *J Univ Sci Technol B*, 2017, 39: 317–322 [邱华鑫, 段海滨. 从鸟群集群飞行到无人机自主集群编队. *工程科学学报*, 2017, 39: 317–322]
- 2 Wang B H, Wu T Y, Li W H, et al. Large-scale UAVs confrontation based on multi-agent reinforcement learning (in Chinese). *J Syst Simul*, 2021, 33: 1739–1753 [王泊涵, 吴婷钰, 李文浩, 等. 基于多智能体强化学习的大规模无人机集群对抗. *系统仿真学报*, 2021, 33: 1739–1753]
- 3 Arbuckle D J, Requicha A A G. Self-assembly and self-repair of arbitrary shapes by a swarm of reactive robots: Algorithms and simulations. *Auton Robot*, 2010, 28: 197–211
- 4 Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. *Science*, 2014, 345: 795–799
- 5 Li S, Batra R, Brown D, et al. Particle robotics based on statistical mechanics of loosely coupled components. *Nature*, 2019, 567: 361–365
- 6 Hegmann T, Qi H, Marx V M. Nanoparticles in liquid crystals: Synthesis, self-assembly, defect formation and potential applications. *J Inorg Organomet Polym*, 2007, 17: 483–508
- 7 Pincirolì C, Birattari M, Tuci E, et al. Self-organizing and scalable shape formation for a swarm of Pico satellites. In: 2008 NASA/ESA Conference on Adaptive Hardware and Systems. Noordwijk, 2008. 57–61
- 8 Fikri M R, Djamari D W. Full-state feedback control design for shape formation using linear quadratic regulator. *IJoCED*, 2020, 2: 83–90
- 9 Wang H, Rubenstein M. Shape formation in homogeneous swarms using local task swapping. *IEEE Trans Robot*, 2020, 36: 597–612
- 10 Bandyopadhyay S, Chung S J, Hadaegh F Y. A probabilistic eulerian approach for motion planning of a large-scale swarm of robots. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Daejeon, 2016. 3822–3829
- 11 Li G, St-Onge D, Pincirolì C, et al. Decentralized progressive shape formation with robot swarms. *Auton Robot*, 2019, 43: 1505–1521
- 12 Derenick J C, Spletzer J R. Convex optimization strategies for coordinating large-scale robot formations. *IEEE Trans Robot*, 2007, 23: 1252–1259
- 13 Wang Y, Wang D B, Wang J H. A convex optimization based method for multiple UAV autonomous formation reconfiguration (in Chinese). *Sci Sin Tech*, 2017, 47: 249–258 [王寅, 王道波, 王建宏. 基于凸优化理论的无人机编队自主重构算法研究. *中国科学: 技术科学*, 2017, 47: 249–258]

- 14 Haghighi R, Cheah C C. Multi-group coordination control for robot swarms. *Automatica*, 2012, 48: 2526–2534
- 15 Yu J, Lavalle S M. Shortest path set induced vertex ordering and its application to distributed distance optimal formation path planning and control on graphs. In: *Proceedings of the IEEE Conference on Decision and Control*. Florence, 2013
- 16 Yu J, Lavalle S M. Distance optimal formation control on graphs with a tight convergence time guarantee. In: *Proceedings of the IEEE Conference on Decision and Control*. Maui, 2012. 4023–4028
- 17 Zhang J, Li W, Kang S, et al. Assigning multiple AUVs to form arrays under communication range limitations based on the element zero method. *IEEE Syst J*, 2021, 15: 1664–1673
- 18 Jung H, Kim D H. Potential-function-based shape formation in swarm simulation. *Int J Control Autom Syst*, 2014, 12: 442–449
- 19 Izzo D, Pettazzi L. Autonomous and distributed motion planning for satellite swarm. *J Guid Control Dyn*, 2015, 30: 449–459
- 20 Bi Q, Huang Y. A self-organized shape formation method for swarm controlling. In: *Proceedings of the 37th China Control Conference*. Wuhan, 2018. 7205–7209
- 21 Xia S X, Li W C, Zhou Y, et al. Improved k -means clustering algorithm. *J Southeast Univ*, 2007, 23: 435–438
- 22 Katsavounis S. A greedy algorithm for scheduling tasks on production lines. *Oper Res Int J*, 2001, 1: 285–298
- 23 Pan F, Chen J, Xin B, et al. Several characteristics analysis of particle swarm optimizer (in Chinese). *Acta Automat Sin*, 2009, 35: 1010–1016 [潘峰, 陈杰, 辛斌, 等. 粒子群优化方法若干特性分析. *自动化学报*, 2009, 35: 1010–1016]
- 24 Hou P, Pan H, Guo C. Simulation research for mobile robot path planning based on improved artificial potential field method recommended by the AsiaSim. *Int J Model Simul Sci Comput*, 2017, 08: 1750046

Heuristic shape formation algorithm in swarm robots based on group auction

QU Yun¹, XIN Bin^{1,2,3}, WANG Qing^{1,2}, ZHANG JunXi¹ & GUO Miao¹

¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China;

² State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China;

³ Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China

The shape formation refers to forming a special shape by controlling the movement of the swarm robots. The problem of shape formation in swarm robots can often be reduced to two sub-problems: the task allocation and the path planning between the robots and the targets. According to the characteristics of large-scale, easy congestion and collision in the shape formation problem, the optimized grouping auction and distributed interaction (OGADI) algorithm is proposed to shorten the completion time of the task. The OGADI algorithm is compared with the shortest path set induced vertex ordering algorithm. The results show that the average completion time of the OGADI algorithm on a scale of 500, 1000, 1300 swarm robots are shortened by 16.1%, 13.6%, and 14.4%, respectively. Some simulation results verify the effectiveness of the proposed method in this paper.

swarm robots, shape formation, auction, distributed interaction, optimization

doi: [10.1360/SST-2021-0349](https://doi.org/10.1360/SST-2021-0349)