

均分点蚁群算法在群集机器人任务规划中的应用与研究^①

余伶俐^② 蔡自兴 刘晓莹 高平安

(中南大学信息科学与工程学院 长沙 410083)

摘 要 提出了一种求解群集机器人协作任务规划问题的均分点蚁群算法(EDPACA)。通过多组蚂蚁群相互协作搜索,构架了一种新蚁群算法的解结构,并设计了更合理的评估函数,使其在评价时充分考虑均衡任务点探测,最后利用 2-opt 技术解决了各子周游路径的交叉问题,获得了总代价最优的解。该算法将蚁群技术首次应用于集群机器人的任务调度规划中,成功解决了中大规模任务规划问题。仿真实验结果表明,均分点蚁群算法能提高群集机器人执行任务的效率,同时也是解决多旅行商问题的另一种新思路。

关键词 蚁群算法,群集机器人,任务规划

0 引 言

任务规划问题是近年来群集机器人研究领域的难点问题,根据不同的需求,实时有效地为各本体分配任务,规划其运动路径及运动速度,是降低能耗节约资源有利途径。国内外已有学者对群集机器人任务规划进行了部分研究,Parker^[1]和 Lagoudakis 等^[2]分别证明了多机器人任务分配是 NP-Hard 问题。Gerkey 等^[3]根据任务(最小分配单元)需要的机器人数目将任务划分为单机器人任务与多机器人任务两类,并在最优分配问题(optimal assignment problem, OAP)框架下利用匈牙利法或拍卖的分布式方法^[4]求解最优分配问题。机器人系统运动规划目的^[5-7]在于根据环境限制与任务需要为各移动机器人规划一条从当前位置达到目标位置的安全最短路径,并确定机器人的移动速度。孙树栋^[8]采用链接图法建立机器人工作空间模型,应用遗传算法规划多移动机器人路径;蔡自兴和彭志红^[9]提出基于协作协进化自适应遗传算法的规划系统能较好地实现静态环境下的多移动机器人路径规划;樊晓平等^[10]和朱庆保^[11]分别采用蚁群算法研究了二维动态环境下移动机器人路径分配问题。这些文献主要集中在机器人路径规划方面,而群机器人任务规划的研究目前尚处于萌芽期^[12]。蚂蚁算法是模拟真实蚁群协作的过程,目前已解决了许多组合优化问题,如旅行商

问题(travel salesmen problem, TSP)问题、分配调度问题、流水作业问题等,所得到的解无论在质量上还是在收敛速度上都要优于遗传算法、模拟退火算法,且具有一定的鲁棒性、自适应性、并行性和正反馈的优点^[13,14]。本文利用均分任务点蚁群算法(equal division point ant colony algorithm, EDPACA)来解决群机器人任务规划问题,通过多组蚂蚁群相互协作搜索,构架了一种新蚁群算法的解结构,同时,为充分考虑任务点探测均衡性,设计了合理的评估函数。

1 群集机器人任务调度规划模型

定义 1: 机器人执行任务的代价是指机器人从当前位置运动到任务位置所需付出的代价。在任务规划时,总代价与机器人当前位置和所需执行的任务之间的距离成正比。

定义 2: 同质机器人为各机器人能力相当,并能独立完成任务,并且各机器人具有相同的行驶速度。

假设 1: 本文只考虑行进过程中的总代价。

假设 2: 任务点均分即各个机器人必须完成均衡的任务点,使得所完成的最大点任务数和最小点数之差最小。

经上述定义和假设,群集机器人任务调度问题可以理解为 k 个机器人从中心任务点出发(本文的初始任务点均设为第一个任务点),分别去执行剩下的 $n-1$ 个任务,每个任务点有且仅有一个机器人

^① 国家自然科学基金(90820302)资助项目。

^② 女,1983 年生,博士生;研究方向:机器人任务规划,仿生硬件,联系人, E-mail: llyu@mail.csu.edu.cn (收稿日期:2008-04-21)

到达执行即可,最后返回中心任务点。如何调度才能使得 k 个机器人在任务点均分的前提下总执行任务的消耗最小,是本文研究的重点。

下面给出群集机器人任务调度规划的数学模型表达式。设 $V = (v_1, v_2, \dots, v_n)$ 是平面上的 n 个任务点的集合, $G = (V, E)$ 是 V 上的完全图。称 H 为图 $G = (V, E)$ 的 k -周游路,如果它是 k 条子周游路的集合 $H = (H_1, H_2, \dots, H_k)$, 这里, H_i 为至少包含 3 条边的简单图, $i = 1, 2, \dots, k$; H_i 经过中心点 v_1 , $i = 1, 2, \dots, k$; 任给 $v \in V \setminus \{v_1\}$, 存在唯一的子周游路 H_i 经过 v 。 k -周游路 H 的长度记为 $C(H)$, 即

$$C(H) = \sum_{i=1}^k C(H_i) = \sum_{i=1}^k \sum_{e \in H_i} C(e) \quad (1)$$

其实群集机器人任务规划问题与多旅行商问题 (multi-person traveling salesman problem, mTSP) 类似, 也即寻求最短 k -周游路 H 问题, 其模型如下式表示:

Minimize:

$$C(H) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (2)$$

$$\text{s.t.} \quad \sum_{j=2}^n x_{1j} = k \quad (3)$$

$$\sum_{j=2}^n x_{j1} = k \quad (4)$$

$$\sum_{i=1}^n x_{ij} = 1, j = 2, \dots, n \quad (5)$$

$$\sum_{j=1}^n x_{1j} = 1, i = 2, \dots, n \quad (6)$$

$$x_{ij} \in \{0, 1\}, \forall (i, j) \in E \quad (7)$$

其中当机器人经过任务点 (i, j) 时, $x_{ij} = 1$, 否则为 0, 如式(7)所示; c_{ij} 为代价。式(3)和(4)保证 k 个机器人从初始任务点出发后回到中心任务点, 式(5)和(6)为一般性约束。

显然, $k = 1$ 时, 该问题类似于 TSP 问题, m TSP 问题可转换成一般形式的 TSP 问题。但目前只有为数不多的文献^[15,16]对 m TSP 问题进行了讨论, 因为对 m TSP 问题作为一个纯粹的数学问题进行研究, 难度较大, 复杂度高, 各种算法都有自身的不足。本文主要针对群集机器人任务规划这个具有特定背景的实际问题进行研究和探讨。

2 基本蚁群系统原理

下面给出对蚁群算法的数学模型的描述。设

$b_i(t) (i = 1, 2, \dots, n)$ 为 t 时刻任务点 i 的蚂蚁数, 则 $m = \sum b_i(t)$ 为全部蚂蚁数。每只人工蚂蚁遵循以下 3 个特性^[17]和 4 个公式^[18]:

① 每只蚂蚁根据路径上残留的信息量和启发信息选择下一任务点, 采用轮转法实现下一任务点的选择, 如式

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in J_k} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta} & j \in J_k \\ 0 & \text{其他} \end{cases} \quad (8)$$

所示。其中, $J_k(i) = \{1, 2, \dots, n\} - \text{tabu}_k$ 表示蚂蚁 k 下一步允许选择的任务点集合。列表 tabu_k 记录了蚂蚁 k 当前走过的任务点。 η_{ij} 是一个启发式因子, 表示蚂蚁从任务点 i 转移到任务点 j 的期望程度。

② 每只人工蚂蚁只能走合法路线, 除非一次周游结束, 否则不允许转到已访问过的任务点, 该过程由蚂蚁的禁忌表来控制。

③ 完成一次周游后, 蚂蚁在其访问过的每一条边上留下相应的信息素。当所有蚂蚁完成一次周游后, 各路径上的信息素根据式

$$\tau_{ij}(t+n) = (1-\rho) \times \tau_{ij}(t) + \Delta\tau_{ij} \quad (9)$$

$$\Delta\tau(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (10)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{第 } k \text{ 只蚂蚁本次循环经过 } ij \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (11)$$

$$\text{或} \quad \Delta\tau_{ij}^k = \begin{cases} \frac{Q}{d_{ij}} & \text{第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 之间} \\ \text{经过 } ij \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (12)$$

$$\text{或} \quad \Delta\tau_{ij}^k = \begin{cases} Q & \text{第 } k \text{ 只蚂蚁在 } t \text{ 和 } t+1 \text{ 之间} \\ \text{经过 } ij \text{ 边} \\ 0 & \text{其他} \end{cases} \quad (13)$$

进行更新。 $\Delta\tau_{ij}$ 表示本次迭代边 ij 上信息素的增量。 $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次迭代中留在边 ij 上的信息素量。其中, Q 为正常数, L_k 表示第 k 只蚂蚁在本次周游中所走过路径的总长度。式(12)、(13)中利用的是局部信息, 即蚂蚁完成一步后更新路径上的信息素; 式(11)利用的是整体信息, 即蚂蚁完成整个循环后, 更新路径上的所有信息素, 通常采用式(11)作为蚁群算法的基本模型^[13]。

3 任务均分点蚁群算法(EDPACA)求解群集机器人任务规划

3.1 EDPACA 的基本思想

(1)EDPACA 中解的结构

求解 TSP 问题的蚁群算法中,每只蚂蚁可构造 TSP 问题的一个解,根据此原理,在解决 m TSP 问题时,采用 k 只蚂蚁共同构造问题的一个解结构, k 表示机器人个数。在 EDPACA 算法中将有 m 组($m \times k$ 个)蚂蚁共同协作来寻求问题的最优解。 m_j^i 表示第 i ($i = 1, 2, \dots, m$)组的第 j ($j = 1, 2, \dots, k$)只蚂蚁从中心任务点出发后回到中心任务点所应该访问的任务点数,同时 m_j^i 还必须满足式(14)的约束:

$$\begin{cases} m_j^i \geq 2 \\ \sum_{j=1}^k m_j^i = n - 1 \end{cases} \quad \text{其中 } i = 1, 2, \dots, m \quad (14)$$

同时,第 i ($i = 1, 2, \dots, m$)组的第 j ($j = 1, 2, \dots, k$)只蚂蚁分配一个周游列表 H_j^i ($i = 1, 2, \dots, m, j = 1, 2, \dots, k$),记录了蚂蚁 j 当前已访问过的任务点,当 H_j^i 所访问过的任务点的数目等于 m_j^i 时,停止蚂蚁 j 的搜索,回到中心任务点。当所有的子周游路径 $\{H_1^i, H_2^i, \dots, H_k^i\}$ ($i = 1, 2, \dots, m$) 便构成了 m TSP 问题的一个解,每次迭代中 m 组蚂蚁可以生成 m 组解。

(2)蚂蚁路径的选择

每只蚂蚁根据边上信息素可用轮转法^[14]独立地选择下一个任务点,即在时刻 t ,对于第 m 组的蚂蚁 k 从任务点 i 转移到任务点 j 的转移概率 $p_{ij}^{mk}(t)$ 为

$$p_{ij}^{mk} = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in J_m} [\tau_{is}(t)]^\alpha [\eta_{is}]^\beta} & j \in J_m \\ 0 & \text{其他} \end{cases} \quad (15)$$

其中, $J_m(i) = \{1, 2, \dots, n\} - \bigcup_{k=1}^k H_k^i$ 表示蚂蚁 k 下一步允许选择的任务点集合。 H_k^i 记录了第 i 组第 k 个蚂蚁所经过的任务点数,当 H_k^i 中的任务点数等于每组各蚂蚁限定任务点数时,蚂蚁回到中心任务点。 η_{ij} 是一个启发式因子, η_{ij} 通常取任务点 i 与任务点 j 之间距离的倒数。 α 和 β 分别表示信息素和启发式因子的相对重要程度。

(3)信息素更新

完成一次周游后,蚂蚁在其访问过的每一条边上留下相应的信息素。当所有组的蚂蚁完成一次周

游后,最优组蚂蚁所经过的路径上的信息素按照式(9) - (11)进行调整更新^[19]。

(4)评估函数的设计

评估函数选用 $fitness = \min(d_1^2 + d_2^2 + \dots + d_k^2)$, d_i 表示每组第 i 只蚂蚁子周游的距离,函数选用其距离的平方和最小,也是均衡访问任务点的体现。

3.2 EDPACA 算法步骤描述

步骤 1:NC(迭代次数) = 0,初始化各边上的信息素 τ_{ij} ($i, j = 1, 2, \dots, n$),将 $m \times k$ 只蚂蚁分成 m 组(每组 k 只)置于任务点中心任务点中。

步骤 2:按照假设 2 绝对均衡任务点数,使得所访问的最大点数和最小点数之差最小,计算第 i 组的蚂蚁 j 所能经过的任务点数 m_j^i 。

步骤 3:对第 i 组的蚂蚁 j ,如果其子周游列表 H_j^i 中的任务点数小于 m_j^i ,则按公式(15)生成选择概率,并根据该值利用轮转法选择下个任务点,否则蚂蚁 j 回到任务点中心任务点。如果所有蚂蚁都回到任务点,转步骤 4,否则转步骤 3。

步骤 4:计算各组蚂蚁的评估函数 $fitness = \min(d_1^2 + d_2^2 + \dots + d_k^2)$,记录当前的最好解。

步骤 5:按公式(9) - (11)更新边上的信息素。

步骤 6:NC = NC + 1,若 NC < NC _ MAX 预定的迭代次数且无退化行为,即找到的都是相同解,则转步骤 2,否则输出最优解。

步骤 7:对最终的最优解,进行各个子周游路径本身 2-opt 去交叉,算法结束。

4 仿真实验及结果分析

为使结果具有参照性和可比性,实验选用 TSPLIB 库中三组标准数据进行测试。图 1 为中南大学智能机器人实验室的 MORCS-2 群集机器人实物图,表 1 为仿真实验所选用参数值,其中 m 为蚂蚁组数,MAX _ NC 为最大迭代次数, q_0 为赌轮法阈值, α 、 β 为式(8)中的参数, ρ 为式(9)中蒸发系数, Q 为式(11)中系数。表 2 为 k 取不同值,任务数分别为 51、76、101 时所得到的最优结果、最差结果和平均值,这是各种情况下算法独立运行 10 次所得结果。在假设 1 的前提下,单机器人执行所有任务总耗时为总体最优距离除以其运行速度,而群集机器人执行全部任务所需时间只取某机器人所耗最大时间值 $\max(TR_1, TR_2, \dots, TR_k)$ 。由同质机器人(定义 2)可得 $\max(TR_1, TR_2, \dots, TR_k) = \max(Len R_1, Len R_2,$

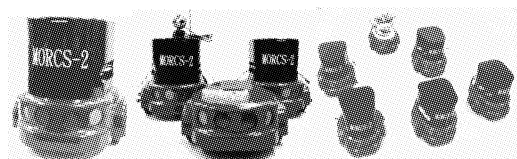


图1 群集机器人实物图

表1 EDPACA 算法参数设置

m	MAX_NC	q_0	α	β	ρ	Q
10	1000	0.25	1.0	5.0	0.5	100

$\dots, Len R_k)/v$, $Len R_k$ 表示第 k 个机器人所行路程, 所以群机器人完成任务总时间远小于单机器人耗时。但从表 2 中看出与单 TSP ($k=1$) 的情况相比 m TSP 总路程消耗略大些, 这说明了费距离而省时间理论。但从执行任务的效率方面理解, 多个机器人进行分头行动, 大大节约了总时间花销, 提高执行总任务的效率, 这就是群集机器人协同合作的优越之处。因此本文实现了单个机器人最大路程最短和

近似总路程最短的总代价最优。

表 3 为两种算法的最优值对比结果, 其中一种方法是在任务点均分时文献[16]算法所得, EDPACA 为本文推荐方法。文献[16]引入 $M-1$ 个虚拟点, 记为 $N+1, N+2, \dots, N+M-1$, 并加入一定的约束条件, 将 m TSP 问题转化为单 TSP 进行解决, 转换后总边数变成 $N+M$ 条, 其约束条件增加, 加上均分点附加约束, 其算法复杂度增加, 算法效果下降。并且各子路径交叉问题没有得到控制, 而 EDPACA 利用了新的解结构, 降低了算法的实现难度, 并在每得到最优解后, 就利用 2-opt 进行去子周游路径交叉和优化, 为此 EDPACA 最优解优于文献[16], 此效果在大规模数据中得到了更好的体现。文献[4-7]提到的市场法和匈牙利法都具有很强的局限性, 任务规模不具可比性, 其算法复杂度也很高。所以 EDPACA 算法为解决规划类组合优化问题提供了新思路。

表2 k 取不同值时, 每种情况分别运行 10 次后的结果

机器人个数 k 取值	最优结果	最差结果	平均值 AVE	相对误差率 (%)
任务点数为 51 个	$k=1$	目前所知整数最优解: 426		
	$k=2$	457.31	485.66	4.59
	$k=3$	464.11	507.60	5.69
	$k=5$	548.81	603.93	5.47
任务点数为 76 个	$k=1$	目前所知整数最优解: 538		
	$k=2$	568.26	595.86	3.60
	$k=3$	605.37	671.83	4.15
	$k=5$	715.45	750.49	3.76
任务点数为 101 个	$k=1$	目前所知整数最优解: 629		
	$k=2$	707.48	739.79	2.69
	$k=3$	714.77	767.63	5.05
	$k=5$	786.50	873.20	5.96

表3 两种算法的最优值比较

	文献[16] 均分点方法最优结果	EDPACA 最优结果
51 个任务点, 2 个机器人	463.74	457.31
51 个任务点, 3 个机器人	485.50	464.11
51 个任务点, 5 个机器人	589.63	548.81
76 个任务点, 2 个机器人	576.83	568.26
76 个任务点, 3 个机器人	625.21	605.37
76 个任务点, 5 个机器人	737.61	715.45
101 个任务点, 2 个机器人	717.80	707.48
101 个任务点, 3 个机器人	727.61	714.77
101 个任务点, 5 个机器人	838.25	786.50

图 2 至图 4 为任务点总数为 51, 76 和 101 时, 机器人数量 k 为 2, 3, 5 时任务的规划路径, 图中各小圆

点代表任务点, 连线代表不同机器人所求解的各分派路径。图 5 至图 7 分别为任务点总数为 51、76 和

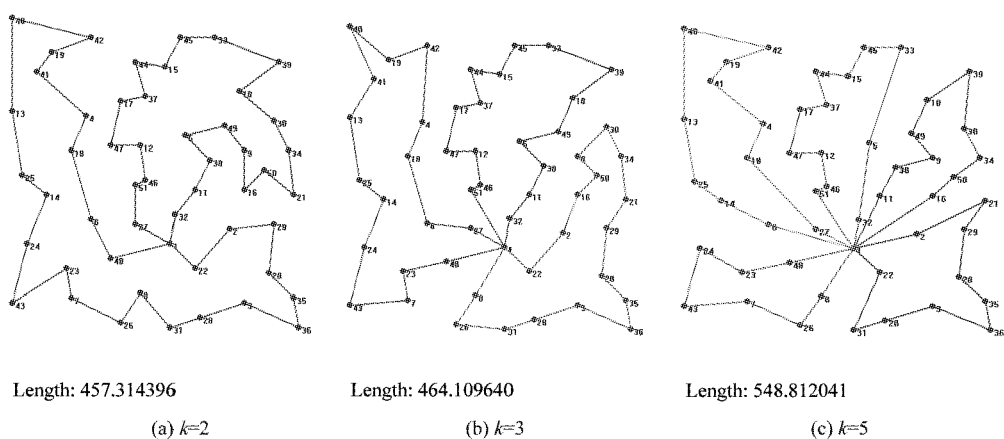


图 2 51 个任务点时最佳任务规划结果

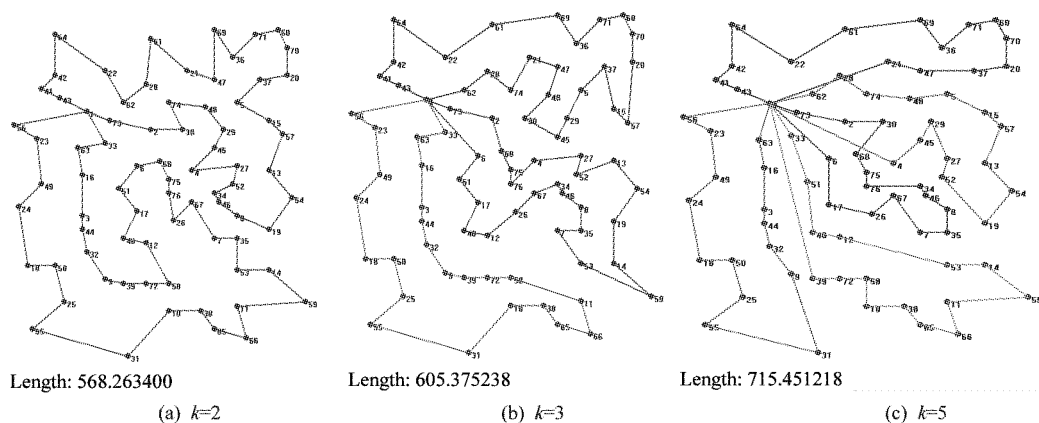


图 3 76 个任务点时最佳任务规划结果

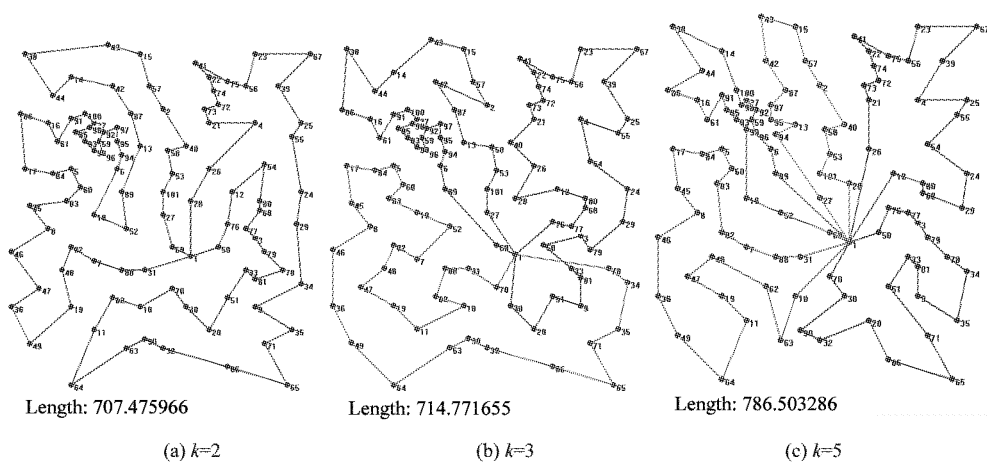


图 4 101 个任务点时最佳任务规划结果

101 时迭代收敛情况。需要进一步说明的是,纵坐标均为 $fitness$ 函数为各子周游距离的平方和 $d_1^2 + d_2^2 + \dots + d_k^2$, 横坐标为迭代次数。每次循环得到的 $best_value$, 当其收敛后,立即将其值设置为 0, 第一个为零的点对应的横坐标 NC 就是收敛次数。例

如:在图 6 中,2 个机器人执行任务时 529 次迭代就可进入收敛,3 个机器人时迭代 541 次进入收敛,5 个机器人时迭代 686 次进入收敛。

综上所述,蚁群算法解的结构需根据求解模型来构造,这样有利于降低算法的实现难度和复杂程

度,同时评价函数也要根据模型求解目标定义,本文需要达到均分任务点的目的,所以 *fitness* 函数和此点联系紧密,这样相互呼应,达到最优的效果。

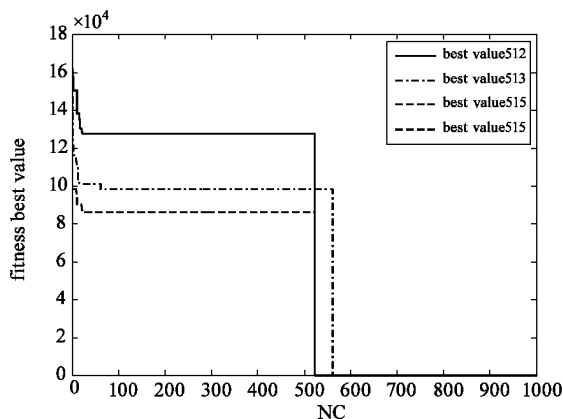


图5 51个任务点迭代收敛情况

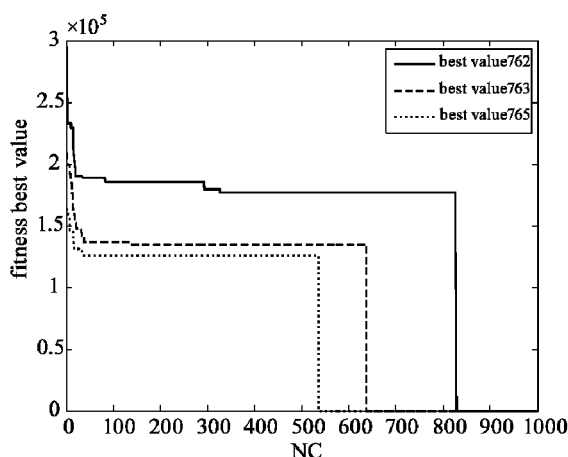


图6 76个任务点迭代收敛情况

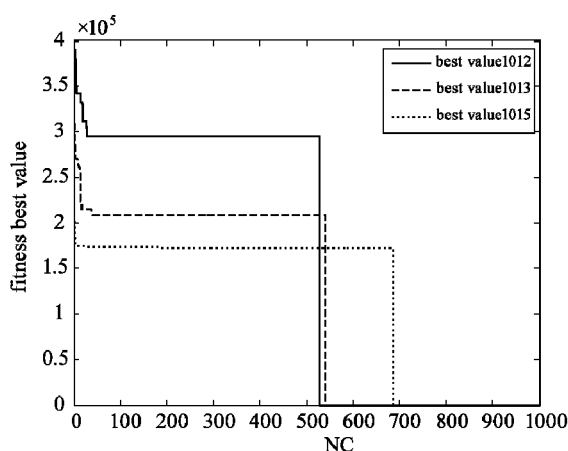


图7 101个任务点迭代收敛情况

5 结论

本文根据蚁群算法的特点,在其解的构造上和评估函数设计上进行了改进,并利用蚁群算法自身特点来解决群集机器人任务规划问题,同时也为解决类似 *m*TSP 难题提供了新的思路。如果没有均分前提,群机器人任务规划应是多目标规划问题,既要使得执行任务总路程最短,又要使得各机器人执行任务点数尽量均衡分配。例如在不满足均分点条件时,若有 3 个机器人执行 51 个任务点,就有 207 种分配方案,除去不满足约束条件式(14)的 24 种,还有 183 种可行解。如果对这 183 种可行解都进行一次 EDPACA 算法的计算的话,运算时间将很长;若有 3 个机器人分派执行 76 个任务点,就有 481 种可能分派的解,除去 36 种不可行解,仍有 405 种可行解。如果是大规模的任务的规划,这种穷举法是不可行的,所以必须结合 EDPACA 算法设计一种选择淘汰机制,得到少量可行解的基础上进行 EDPACA 算法,这将成为今后研究的突破口之一。本文利用 2-opt 技术对子周游路径本身进行去交叉,但子周游路径与子周游路径间的去交叉问题还是没有得到很好的解决,到目前为止在国际知名刊物上也未见能很好地解决此问题的方案。

动态环境下的群集机器人任务规划是研究难点之一,其任务点随机地增加和减少,或是机器人在恶劣的环境下受到损坏不能正常工作等问题,而这些情况在实际的现场都很有可能发生。另外,环境中出现动态的障碍物以及机器人间的避碰问题,这些难题已在多机器人系统研究中得到充分的体现。如何解决复杂环境下的动态任务规划已成为当今机器人研究领域的难点问题。综上所述,群集机器人任务规划研究还处于初始阶段,需对其进行更加深刻彻底的研究。

参考文献

- [1] Parker L E. Lifelong adaptation in heterogeneous multi-robot teams: response to continual variation in individual robot performance. *Autonomous Robots*, 2000, 8(3): 239-267
- [2] Lagoudakis M, Berhault M, Koenig S, et al. Simple auction with performance guarantees for multi-robot task allocation. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sendai, Piscataway, USA, 2004. 698-705
- [3] Gerkey B, Mataric M. Sold: auction methods for multi-robot

- coordination. *IEEE Transactions on Robotics and Automation*, 2002, 18(5):758-768
- [4] 董胜龙,席裕庚,陈卫东.多机器人不确定协作任务的动态优化方法.机器人, 2002, 24(1): 31-34
- [5] Zlot R, Stentz A. Complex task allocation for multiple robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005. 1515-1522
- [6] Fua C H, Ge S S. COBOS: cooperative back off adaptive scheme for multi-robot task allocation. *IEEE Transactions on Robotics*, 2005, 21(6): 1168-1178
- [7] Vig L, Adams J A. Multi-robot coalition formation. *IEEE Trans on Robotics*, 2006, 22(4): 637-649
- [8] 孙树栋,林茂.基于遗传算法的多移动机器人协调路径规划.自动化学报, 2000, 26(5): 672-676
- [9] Cai Z X, Peng Z H. Cooperative coevolutionary adaptive genetic algorithm in path planning of cooperative multi-robot systems. *Journal of Intelligent and Robotic Systems*, 2002, 33(1):61-71
- [10] 樊晓平,罗熊,易晟等.复杂环境下基于蚁群优化算法的机器人路径规划.控制与决策, 2004, 19(2): 166-170
- [11] 朱庆保.全局未知环境下多机器人运动蚂蚁导航算法.软件学报, 2006, 17(9): 1890-1898
- [12] Carusi A, D'Abramo G, Valsecchi G B. Orbital and mission planning constraints for the deflection of NEOs impacting on Earth. *Icarus*, 2008, 194(2): 450-462
- [13] Dorigo M, Maniezzo V, Colomi A. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on systems man and cybernetics-Part B: Cybernetics*, 1996, 26(1): 29-41
- [14] Dorigo M, Gambardella L M. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans on evolutionary computation*, 1997, 1(1): 53-65
- [15] Sofge D, Schultz A, Kenneth D J. Evolutionary computational approaches to solving the multiple traveling salesman problem using a neighborhood attractor schema. In: Lecture Notes in Computer Science. Heidelberg: Springer Berlin, 2002. 2279. 153-162
- [16] Pan J J, Wang D W. An ant colony optimization algorithm for multiple traveling salesman problem. In: Proceedings of the International Conference on Innovative Computing Information and Control, Beijing, China, 2006. 210-213
- [17] 胡小兵. 蚁群优化原理、理论及其应用研究: [博士学位论文]. 重庆: 重庆大学控制理论与控制工程专业, 2004
- [18] Kulatunga A K, Liu D K, Dissanayake G, et al. Ant colony optimization based simultaneous task allocation and path planning of autonomous vehicles cybernetics. In: Proceedings of the IEEE Conference on Intelligent Systems, Beijing, China, 2006. 1-6
- [19] Wang L, Zhang Q, Wan N F. Simulation study on searching for food by cooperation of multi-robots with swarm intelligence. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007. 2296-2300

Application and research on swarm robots mission planning using equal division point ant colony algorithm

Yu Lingli, Cai Zixing, Liu Xiaoying, Gao Pingan

(School of Information Science & Engineering, Central South University, Changsha 410083)

Abstract

An equal division point ant colony algorithm (EDPACA) was proposed to solve the swarm-robot mission planning problem. Firstly, the algorithm was designed by a novel solution construction through multi-group ants' cooperative search, and a more reasonable evaluation function which sufficiently considered the equal division mission point factor. Secondly, the crossover problems of sub-circular paths were solved by the 2-opt method. It is the first time to apply an ant colony algorithm to swarm-robot mission planning, the proposed algorithm can solve large scale mission planning problems effectively. The experiment results show that EDPACA improved the efficiency of executing a large number of tasks. Additionally, it also provides a new idea to solve the multi-person traveling salesman problem.

Key words: ant colony algorithm, swarm-robot, mission planning